

**Софийски университет “Св. Климент Охридски”
Факултет по Математика и Информатика
Катедра "Информационни технологии"**

ДИПЛОМНА РАБОТА

**Информационна система за труд и
работна заплата в строителна фирма**

Научен ръководител
Н.с. Сергей Върбанов

Дипломант
Момчил Стефанов Рангелов
Информатика, фак. № 43059

София
Ноември 2005

СЪДЪРЖАНИЕ

УВОД	3
ГЛАВА 1. Необходимост от системата и приложение.	5
1.1 Развитие на технологиите и информационното общество	5
1.2 Информационните технологии в сферата на строителството.	8
1.3 Описание на системата и необходимост от нейното използване.	14
1.4 Сравнителни характеристики на подобни ситеми и тяхната ефективност.	18
ГЛАВА 2. Средства за реализация на системата.	29
2.1 Избор на среда за разработка.	29
2.2 Разработка на приложения с Delphi	32
2.1.1 Характерни черти на Delphi	32
2.1.2 Програмиране на Delphi	33
2.1.3 Средствата на Delphi	34
2.1.3.1 Термини	34
2.1.3.2 Среда за разработка	35
2.3 Релационни база данни и Delphi.	37
2.3.1 Основи на бази данни в Delphi	37
2.3.2 Компоненти в Delphi за работа с бази данни	38
2.3.3 Компоненти на други разработчици.	44
ГЛАВА 3. Проектиране и архитектура на ситемата	47
3.1 Проектиране на базата данни и схема на таблиците	47
3.2 Проектиране на потребителския интерфейс	55
3.3 Обектен модел на системата.	59
ГЛАВА 4. Описание на функциониращата система.	65
4.1 Инсталиране и начин на работа с продукта.	65
4.2 Преглед на основните възможности на продукта.	67
4.3 Допълнителни характеристики и функционалност.	83
ЗАКЛЮЧЕНИЕ	88
БИБЛИОГРАФИЯ	89
ПРИЛОЖЕНИЕ	90

УВОД

През последните две, три години строителството в България е в голям подем като непрекъснато се увеличават новоизграждащите се обекти и фирмите, които ги изпълняват. Това от своя страна води до увеличаване на персонала, зает в този сектор. За да бъдат конкурентноспособни и гъвкави на пазара, строителните фирми трябва да използват освен съвременни технологии и материали за строителството на обектите, също така и съвременни технологии за проектиране и управление.

Поради тези причини на някои фирми от строителния бранш се налага да използват специализирани решения за управление на своята дейност, отразяващи техни специфични изисквания за начина им на работа. В предлаганата дипломна работа като цел се поставя анализирането на процеса на отчитане на труда и формирането на тази база работна заплата и се реализира продукт, който да автоматизира голяма част от него. Разработената програма е предназначена за конкретна фирма и е съобразена със спецификациите, поставени от нея.

Основата на отчитане на труда в тази фирма е броя на отработените часове за седмица на всеки от работниците, като в края на седмицата се изготвя отчет от ръководителя на бригадата. Въз основа на тези отчети трябва да се формира и заплащането, за всеки работник, като се вземат предвид и редица други фактори. Тъй като фирмата договаря за всеки обект различни цени за труд и различни срокове на изпълнение, едно от най-важните условия е договорената сума да не надхвърля изплатената на работниците, дори пусковият срок да е изтекъл. Това се постига чрез въвеждане на коефициент за всеки обект, чрез който могат да се коригират договорените надници за час работа. Друго основно нещо, което влияе на сумата за разпределение е използването на механизация, т.е. на различни машини за автоматизиране на някои дейности. Също така, ако се използват външни работници или фирми подизпълнители, съответно това намалява сумата за собствените работници, от която се изплащат на другите. Предвидено е да има възможност да се налагат персонални глоби и поощрения, които допълнително влияят на заплатата. Има и други фактори вземани в предвид, като извънредния труд над осем часа на ден, който се заплаща по 1.5 от нормалната часова ставка, през събота и неделя, както и официалните

празници часовата ставка е два пъти по-голяма. Другата възможност, която се предлага е заплатите на администрацията да са обвързани със заплатите на работниците.

Програмата поддържа база данни с обектите, на които работи фирмата, имената на работниците, машинния парк на фирмата, седмичните отчети и изчислените заплати за всеки месец. На базата на тези данни могат да се правят различни справки и съответно да се разпечатват като отчети. Също така се предлага и експорт на данни в CSV формат, който е доста универсален и така да могат да се импортват в Excel, счетоводни програми или други база данни.

Предимствата от използването на разработената програма са, че чрез нея може много по-гъвкаво да се определя работната заплата на всеки според заработеното и според това как върви изпълнението на обекта и да се правят детайлни справки, от които може да се установи дали нещо не върви според плана и да се вземат навременни мерки. Тази функционалност ще допринесе за по-добро планиране на средствата и да се избегнат загуби от някои обекти.

Настоящата дипломна работа е структурирана в четири глави. В първата глава се описани използването на информационните технологии в строителството, разгледани са съществуващи приложения с подобна функционалност и подробно са разяснени възможностите на разработваната програма. Във втора глава е обяснен избора на среда за разработване на приложението и е направен преглед възможностите на избраната среда, в случая – Delphi. Трета глава описва проектирането на базата и потребителския интерфейс и предлага клас диаграма на обектния модел на системата. Четвърта глава е разгледано инсталирането на вече готовата система и са описани функционалните възможности и как се работи с приложението от потребителска гледна точка.

ГЛАВА 1. Необходимост от системата и приложение.

1.1 Развитие на технологиите и информационното общество

Разпространението на информационните технологии през последните 30 години е всеобхватно и глобално, те започнаха да се прилагат масово в науката, военното дело, медицината и в много други сфери на живота, като се разпространиха в целия свят. Революцията в областта на интегралните схеми през последните 10 години осигури технологична мощност за универсално доставяне на информация и в други области. Това се почувства най-силно в областта на бизнеса. От текстообработката до счетоводството, от управлението на проекти до компютъризираното изготвяне на сметки за плащане – въздействието на информационните технологии революционизира начина, по който работи бизнесът. Основно е изискването за осигуряване на информация за бизнеса. Компютъризираните информационни системи дават тази информация. За да бъдат проектирани ефективно подобрени системи движещи сили трябва да бъдат информационните нужди на бизнеса.

През последното десетилетие използването на информационните технологии в бизнеса нараства бързо и те проникват във всички дейности на организациите.

Научно-техническият прогрес е генератор на предпоставки, които се обективизират във фактори за развитие и промяна на обществото. Такива предпоставки и фактори са новите комуникационни и информационно-изчислителни среди. Характерно за тези формиращи се среди е утвърждаването на нов ресурс. Това е информационният ресурс. Днес той реално се изравнява по място и значение с другите ресурси в корпоративните системи. Оформя се и тенденция за неговото бързо налагане като един от основните ресурси на обществото. Той, заедно с реализиращите го комуникационни и информационно-изчислителни среди, е основополагащ за определяне на бъдещото общество като информационно.

Индикацията на бъдещото общество като информационно има както формален, така и същностен характер. Определящ смисъла на тази индикация е подходът към характеристиката на обществото. Самото общество може да се самоопределя по

различен начин. От една страна, това може да е подход за определяне, ориентиран към видимите и извеждани формално на преден план явления и процеси в него. Така то акцентира върху една или друга своя видимост и насочва общественото внимание към нея с цел да се усили обществения ѝ интегритет или също така целево да се депозитивира. Такова насочване на вниманието е характерната за съвременния етап на развитие растяща информатизация на обществото.

От друга страна обществото може да се самоопредели с подходящи, характеризиращи хармонията му обективно действащи закони на неговото развитие. Те определят същностната характеристика на обществената формация.

Големият и истински фундаментален въпрос, който вече е актуален днес и ще става все повече актуален в близкото бъдеще, е: "Създадени ли са и съществуват ли обективни основания да се говори за формиране и определяне на нова обществена формация или става въпрос за поредна определителна характеристика, отразяваща реално действително нови технологии, среди, явления и процеси в обществото?" Въпросът е действително голям и сложен. Той е с висока научна фундаменталност и практическа значимост за определяне на обществото и неговото развитие.

Безспорна е обективността на обстоятелството, че благодарение на научно-техническото развитие обществото днес се отличава от това, което е било вчера. Сега то е по-различно от преди. Утре ще бъде по-различно от сега. Обществото и като цяло, и като органически диференцирани части е и ще става по-технологично, по-развито, по-информационно и по-различно ресурсно. След получените широка популярност понятия като изчислителна и комуникационна техника, кибернетика, комуникационни и информационно-изчислителни технологии и други такива, в социалния и най-вече в технологичния прожектор на обществото се открояват, налагат и придобиват популярна приложимост и нови понятия и категории. Специално внимание заслужават тези за информационен ресурс, информационност и информационна осигуреност на процесите на всички нива на обществото.

Голямо е и бързо се увеличава множеството от явления и процеси, които формират обективни основания за анализ и оценка на степента и формите на готовност за развитие на обществото. Достигнатата степен на натрупвания и активираната готовност за развитие позволяват да се твърди, че обществото видимо навлиза във фаза на частни и фундаментални промени. В основата на тези промени се проявява обусловената зависимост между производителните сили и производствените отношения. В съвременните научно-технологични условия на утвърждаване на

качествено новите продукти от симбиозата на комуникациите и изчислителната техника, на растящото пресичане, взаимно проникване и сливане в тях на новосъздаващата се комуникационна и информационно-изчислителна среда се изменят старите и се създават нови производителни сили. Те се реализират ефективно чрез реалното вграждане на тези нови технологии и среди в конкретните работни (производителни) машини, в организацията на производството и управлението на обществото.

По този начин се поставя необходимото обективно начало за навлизане в поредния нов етап от процеса на научно-техническия прогрес. Натрупват се предпоставки и на тяхна основа се формира тенденция за обективизация и необратимост на процеса към нова фаза на научно-техническата революция. Тя от своя страна оказва силно и многостранно влияние за създаване на условия за проектиране образа на нова обществена формация.

Тази обществено-икономическа формация ще създава и определя бъдещото съществуване на човечеството. Най-вероятната характерна черта на очертаващото се бъдещо общество е неговата повсеместна информатизация. Тя е едно от извежданите на преден план основания още отсега, макар и с много висока степен на условност, обществото да се определи като информационно общество.

Опорните точки за повсеместна информатизация, а оттам и за информационно индикиране на обществото, са формиращите се и бързо разширяващи се информационно-комуникационни среди. Те се създават върху основата на бързо развиващите се комуникационно-информационни и информационно-изчислителни технологии и ефективната организация и функциониране на корпоративни и глобални информационни системи. Технологичното и организационно обвързване на комуникационните канали и мрежи с циркулиращия в тях и поддържан в обслужваните корпоративни и глобални системи информационен ресурс дава основания за условно обособяване на съответни информационно-комуникационни среди.

Определящо място в тези среди заема информационният ресурс, както начините и способностите за неговото ефективно използване. Най-доброто професионално и обществено възприемане и утвърждаване на тези среди е те да се базират и изграждат върху такава мрежова комуникация, която да създава и поддържа работна среда за реализиране на информационно-комуникационни и изчислителни технологии от нов тип. За създаването на такива технологии е необходима истински адекватна на целта

среда, а не ограничаването на решенията към един или друг вид комуникации и технологии в мрежа.

Натрупаният опит в информационното осигуряване на различни по характер организации и стопански субекти позволява да се прогнозира и открие тенденция за преход от индивидуално и локално обработване и използване на информацията към създаване и ефективно експлоатиране на технологии за обработка и използване на информационния ресурс във вътрешноорганизационни (корпоративни) информационно-комуникационни среди. На корпоративно равнище функциите на такава технико-технологична организация могат да бъдат реализирани от информационно-комуникационна среда от тип „интранет“.

Определящото в тази организация не е акцентът върху достигнатите високи експлоатационни характеристики на изчислителната техника и нейната информационна комутативност, а усиленото внимание към ефективната организация за създаване, придобиване, предоставяне и работа с информационния ресурс. Така се поставя фундамент не само за действително ефективна обработка на информационния ресурс, но и за неговото добро стопанисване и рационално използване в организацията. В резултат на качествените изменения в информационно-комуникационната среда се формират основания за нов етап на информатизация на производствените и управленски процеси в организациите. Генерират се предпоставки както за активно влияние върху производителните сили и производствените отношения, така и за превръщането на новоформиращата се среда в тяхна органическа част.

Бъдещото развитие на философията и реализацията на информационни технологии и създаването на нови информационно-комуникационни среди ще обективизира тяхното утвърждаване като фактор за информатизация на обществото. Информатизирайки обществото, информационно-комуникационните среди реализират стъпка в неговото развитие. Стъпка, която може да го характеризира и дори да го индикира като информационно общество.

1.2 Информационните технологии в сферата на строителството.

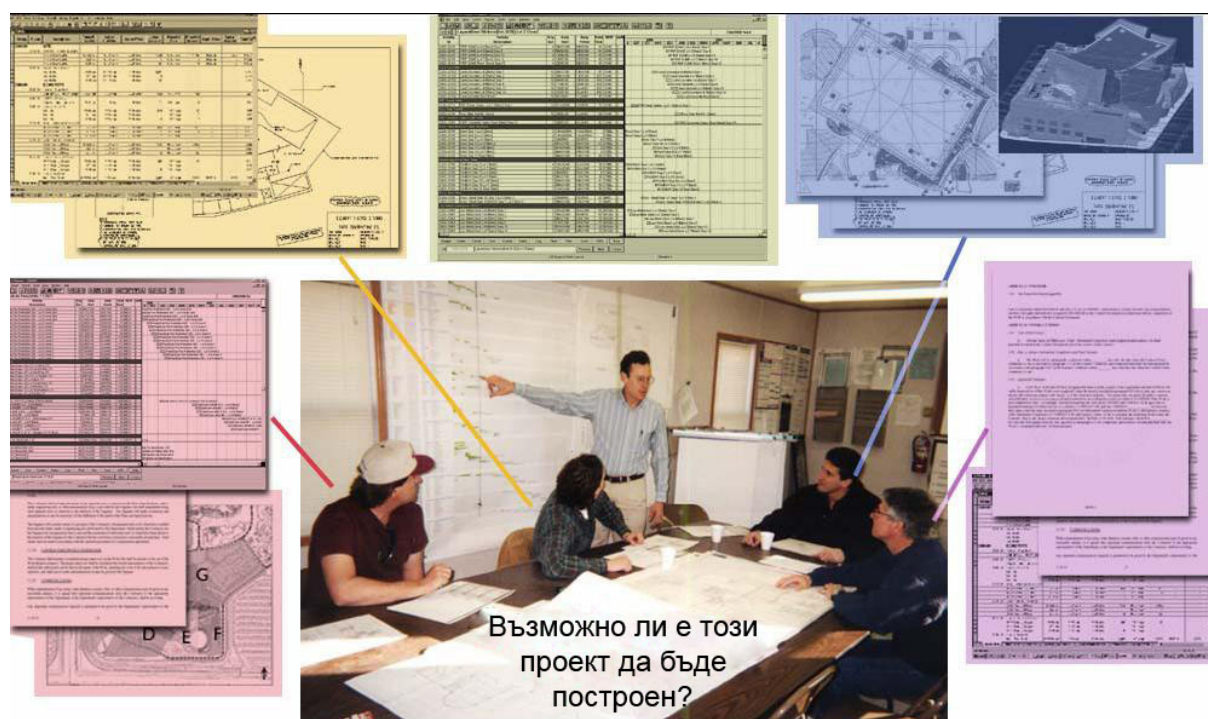
Информационните технологии променят из основи начинът, по който ръководим бизнеса. Строителната индустрия не е изключение, въпреки че цялостното развитие

при технологията е бавно. Някои предприемачи в индустрията са доста напреднали в употребата на информационните технологии. За съжаление повечето от тях не са. Широкото разпространение на промените ще отнеме време докато предприемачите разберат потенциала на информационните технологии и пожелаят да се възползват от него. Въпреки това, информационните технологии се променят значително бързо в много бизнес сектори. Важно е това да не се забравя [5].

В много отношения строителната индустрия е интензивна откъм информация. Това се дължи на много фактори: сложността и големината на крайния продукт; нуждата от визуализация и технически анализ на етап дизайн; разнообразието на различни ноу-хау и материали, необходими за издигането на сграда както и големият брой различни участници при създаването на проект. Като следствие количеството и разнообразието на получената информация, създадена по време на типичното създаване на проект е доста важна. Използването на информационни технологии за обработка, управление и прехвърляне на данни би предоставила големи потенциални ползи на тази индустрия, както при създаването на по-ефективен процес на проектиране и изграждане, така и при осигуряването на по-добро качеството на построените сгради.

През последните 20 години се осъществиха драматични подобрения и широка употреба на информационните технологии при описанието и документирането на работата на много дисциплини, включени при изграждането на проекти. Днес практически цялата информация по проектите е вкарана в софтуерни инструменти или е създавана от компютърни програми и е представена в много и различни формати, използвани от дисциплините, включени в проекта. Софтуерните инструменти могат да бъдат по-обща, като например електронни таблици и софтуер, обработващ текст или пък специфични инструменти като механичните САD програми или софтуер, оценяващ разходите. Както е показано на фиг. 1, форматите, които обикновено се използват за представянето на информация при строителството включват текстови документи, 2D и 3D рисунки, стълбовидни графики и други формати, различни диаграми и схеми, таблици, и др. За повечето решения относно проект, инженерите от различни области, като тези показани на картинката на типична работна среща (фиг. 1) (дизайнер, мениджър на проекта, оценител на разходите, отговорник за графика, и МЕР (координатор по механиката, електричеството и разпределението) трябва да разпределят тяхната информация с другите от екипа по проекта. Целта на срещата, показана на фиг. 1, е да се координират подробния проект и методите на изграждане, разходи и график за офис сграда. На тази среща всеки инженер е формирал в главата си

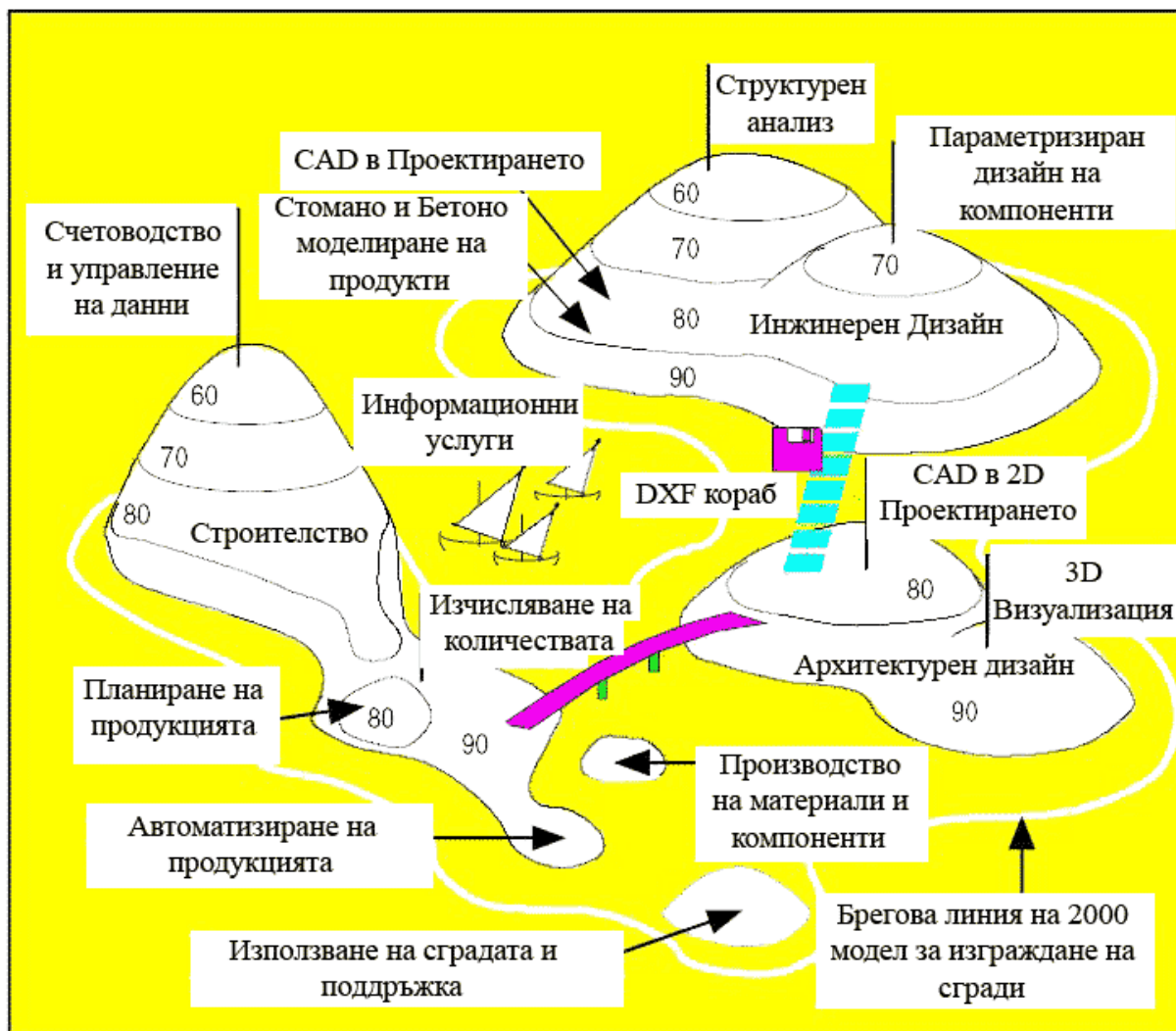
образ на текущото състояние на проекта и представи за бъдещите ситуации, базирани на неговите интерпретации на документите от други инженери. Тези интерпретации са формирали базата за дискусии и решения относно най-подходящата идея за съоръжението и неговите части, кога, как и от кого трябва да бъде построено, колко дълго ще продължи целият проект или част от него, колко ще струват нещата, и т.н. По този начин голяма част от планирането и координирането на проекта се е появило първоначално в главите на инженерите и не е било подпомогнато от информационните технологии. Тъй като главно решенията са базирани на личностни и човешки интерпретации на информацията, която е събрана от много инженери от различни области, процесът на вземане на решение и последващите действия и резултати не са постоянни и повтарящи се от среща на среща и от проект на проект. Като резултат е трудно да се предскаже резултата от текущия план и процеса на изграждане, а информационните технологии допринасят малко за по-надеждното предсказването на резултата от проектите.



Фиг. 1: При всеки проект няколко специалисти от различни области се събират, за да планират и придвижат проекта. Всички специалисти документират своята работа като използват различни информационни системи и формати, за да представят информацията, от която се нуждаят при тяхната работа.

Тъй като повечето от тези дискусии и решения изискват участието на инженери от различни области, от първостепенна важност е документите на различните специалисти да бъдат базирани на една и съща информация и тя да е координирана и предадена ефикасно. Координирането и обединяването на информация от различните области по време на няколко фази от проекта е станало изключително трудно и скъпо, тъй като количеството на електронна информация, която всяка област генерира, се е увеличило.

Най-ранните приложения на информационните технологии в строителството засягали техническите пресмятания, област, в която употребата на компютри днес е задължителна. През 80-те години, употребата на компютри за обикновени строителни работи като теглене, писане на спецификации или пресмятане на разходите станали всекидневие, особено след появата на персоналния компютър. Съотношението на документите, създадени на компютър (към всички строителни документи) се е увеличило значително през последните 5 години. В някои ключови области, като създаване на архитектурни скици, употребата на компютър става по-скоро правило, отколкото изключение.



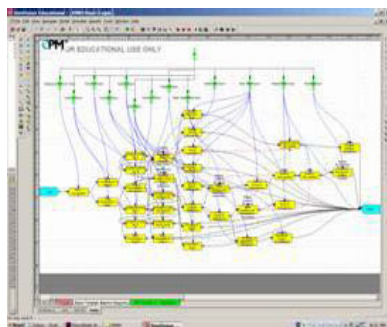
Фиг. 2 Интегрирано използване на информационни технологии в строителството, представено чрез схема на “острови”, като са показани различни сфери и дейности от процеса на строителство и взаимовръзката между тях [5].

Тук на кратко сме направили преглед на бъдещата роля и обхват на информационните технологии в строителството и въвежда интегрирано ПОП (продукт, организация, процес) моделиране в подкрепа на предизвикателствата, отбелязани по-горе и определя виртуален дизайн и изграждане (VDC) като метод за дизайн за по-ефективно действие на информационните технологии в подкрепа на интегрирания ПОП дизайн (Фиг. 2). На Фиг. 3 са показани прозорци на някои софтуерни приложения, участващи в реализирането на този модел.

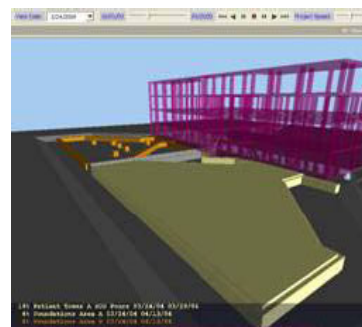
3D/ADT ArchiCAD



SimVision



CP4D



Фиг.3: Моделът Продукт, Организация, Процес, който използва няколко комерсиални софтуерни инструменти.

(1) Роля на информационните технологии

В подкрепа на тези виждания предприемачите ще използват информационните технологии, за да симулират, анализират и оценят очакваното изпълнение на дизайна на съоръжението, проектирането на улеснения при процеса на доставка (план на проектиране и конструкция) и устройството на организацията, която ще изпълни процеса. Тези симулации, анализи и оценки трябва да бъдат базирани върху интегриран модел, описващ проектираното съоръжение, организация и процес. Така че резултатите от симулацията, анализа и оценката трябва да бъдат визуализирани, за да стане ясно какви са връзките между оптимизирането на съоръжението, организацията и процеса за конкретна сфера в сравнение с цялостния проект в широк кръг от критерии, които обикновено са част от строителните проекти. Информационните технологии трябва също да поддържат автоматизацията на процеса на въвеждане на симулация, анализ и оценка и да автоматизира симулациите, анализите и оценките колкото е възможно повече. Накрая информационните технологии ще поддържат оптимизацията на структурата на процеса от гледна точка на много области [8].

(2) Обхват на информационните технологии

Както е илюстрирано в малкия пример за срещата между различните инженери, обхватът на информационните технологии трябва да бъде мултидисциплинарен, т.е. информационните технологии трябва да поддържат интегрирането на информация и перспективите за алтернативите на проекта в много области. Информационните технологии трябва също да обхващат и дизайна на продукта (гъвкавост, обхват на проекта), организацията по осъществяване на дизайна и конструкцията и процесът

(графикът) по изпълнение на проекта. Ние наричаме този обхват (“интегрират ПОП дизайн”), където ПОП, както вече споменахме означава продукт, организация, процес. Както сочат примерите, много решения включват размяната между продукт, организация и дизайн на процеса. Според нас дизайнът на проекта не е цялостен докато продукта, организацията и процесът не са проектирани и взаимоотношенията между тези 3 области разбрани. Причината за създаването на продукта, организацията и обработката на проекта като част от главната сфера на информационните технологии е, че лицата, които отговарят за продукта могат да решат какво да построят, кой и как да го построи, кога да го построи, т.е., продуктът, организацията и планът на процеса са независими променливи величини на проекта. В такъв случай тези решения водят до специфично изпълнение на интегрирания ПОП дизайн по отношение на разходи, безопасност и други критерии на проекта. Тези предвиждания на изпълнението служат като критерии за оценяване на относителните и абсолютните достойнства на определен проект. Такъв интегриран ПОП проект изисква моделиране на системите и компонентите, които изграждат продукта, участниците, работните колективи, разпределението на задачи, и други организационни аспекти, както и дейностите, които обхващат процесите на планиране, изграждане и изпълнение. Дейностите осигуряват главната връзка между проектирането на продукта и организацията, тъй като всеки компонент на проектирането на продукта води до една или няколко дейности по проектирането, изграждането и изпълнението и на всеки участник или колектив в организацията на проекта са разпределени една или няколко задачи [8].

1.3 Описание на системата и необходимост от нейното използване.

През последните две-три години строителството в България е в голям подем, като непрекъснато се увеличават новоизграждащите се обекти и фирмите, които ги изпълняват. Това от своя страна води до увеличаване на персонала, зает в този сектор. За да бъдат конкурентно способни и гъвкави на пазара, строителните фирми трябва да използват освен съвременни технологии и материали за строителството на обектите, също така и съвременни технологии за проектиране и управление.

Поради тези причини на някои фирми от строителния бранш се налага да използват специализирани решения за управление на своята дейност, отразяващи техни

специфични изисквания за начина им на работа. Разработваната програма е предназначена за конкретна фирма и е съобразена със спецификациите, поставени от нея. Фирмата поръчител на програмата има нужда от информационна система за отчитане и управление на труда и формирането на тази база работна заплата. За това ние ще анализираме методите за изчисление на заплатите, по които до момента се ползват във фирмата и ще реализираме продукт, който да автоматизира голяма част от този процес.

Тъй като фирмата има много специфичен начин за формиране на заплатите не е намерила подходящ масов продукт, който да използва за тази дейност, което ще стане ясно от разгледаните в следващата подточка няколко от най-известните и разпространени приложения в тази сфера. Поради тази причина в последните няколко години е използвала набор от свързани по между си екселски таблици. Това от своя страна има някои недостатъци в дадения случай, тъй като се налага за някои от изчисленията да се ползват данни от предходния месец или повече назад, което ще стане видно от описанията на модела по-надолу. Ексел-ът е много добра програма за електронни таблици, в която могат да се правят различни изчисления, но няма вградена функционалност за запазването им като база данни. Ако не се разработи специално приложение на "Visual Basic for Application", което да съхранява данни от таблиците в база данни, друг начин за автоматично пазене на въведената информация за предишни месеци няма. Остава ръчното им записване под различни имена, но тъй като са няколко файла и са взаимно свързани това прави този процес труден и досаден. А също така предполага и възникването на грешки или евентуална загуба на информация при неправилно записване или пропускане да се запише файла с друго име, за да са съхранят прежде въведените данни. Така с непрекъснатото увеличаване на данните и все по-трудното им обработване възниква необходимостта да се създаде специализирано софтуерно решение, което да улеснява и автоматизира този процес, а също така и данните да се съхраняват в база от данни.

Основата на отчитане на труда в тази фирма е броя на отработените часове за седмица от всеки от работниците, като в края на седмицата се изготвя отчет от ръководителя на бригадата. В едни бланки се попълват имената на всеки работник и колко часа е работил съответно за всеки ден даденият работник. Един служител може да е работил на повече от един обект през седмицата и следователно името му ще фигурира в повече отчети. Също така в тези бланки се попълват използваните машини на обекта и каква работа са извършвали и за колко часа, ако са осигурявали

транспортни услуги или придвижването до обекта се заплаща също се записва. Освен за работниците и механизацията в отчета влизат и данни за използваните евентуално външни работници или фирми подизпълнителки. Въз основа на тези отчети трябва да се формира и заплащането, за всеки работник, като се вземат предвид и редица други фактори. Тъй като фирмата договаря за всеки обект различни цени за труд (най-често цената е за час на човек, т.е. в човекочасове) и различни срокове на изпълнение, едно от най-важните условия е договорената сума да не надхвърля изплатената на работниците, дори пусковият срок да е изтекъл. Всеки месец фирмата изплаща месечна заплата на работниците от договорените пари за обектите, но изпълнението на някои от обектите може да се забави и тогава се стига до ситуация, в която парите по договор са изчерпани, а трябва да се плащат заплати за просрочените месеци. Така фирмата вече ще е изплатила на работниците всичките пари, договорени за труд по обекта и после ще трябва да плаща от собствените си резерви. Избягването на това положение може да се постигне с въвеждане на коефициент за всеки обект, чрез който могат да се коригират договорените надници за час работа, които се изплащат текущо през месеците на работа по обекта. Така ако работата изостава на даден обект, часовата надница договорена за него се умножава по число по-малко от единица и по-този начин се намалят текущо плащаните суми и остават средства и за просрочваните месеци. Съответно ако работата тече по-бързо от предвиденото този коефициент може да се приравни на число по-голямо от единица и така работниците ще получат договорените суми и няма да бъдат ощетени. Друго основно нещо, което влияе на сумата за разпределение е използването на механизация, т.е. употребата на различни машини за автоматизиране на някоя дейности по обекта. При договарянето на цената за обекта, освен за работници се предвижда сума за използване на машини, която ако бъде превишена, трябва да се удържи от сумата за предвидена за работниците, иначе фирмата отново ще е на загуба, но ако не бъде използвана до край ще трябва да се преразпредели между работниците. Също така ако се използват външни работници или фирми подизпълнители, съответно това намалява сумата за собствените работници, от която се изплащат на другите. Предвидено е да има възможност да се налагат персонални глоби и поощрения, които допълнително влияят на заплатата. Те представляват фиксирана сума в левове, която се прибавя или изважда от заплата съответно при добри прояви или направени трудови нарушения или други провинения от съответния работник или всички работници. Освен тези форми за стимулиране на работниците е добавен и процент като параметър за увеличаване и намаляване на заплата

по преценка на управителя. Това се налага поради факта, че някои работници работят по-малко, за сметка на други, които работят повече и по-бързо. Има и други фактори, взети предвид, като извънредния труд над осем часа на ден, който се заплаща по 1.5 от нормалната часова ставка, през събота и неделя, както и официалните празници часовата ставка е два пъти по-голяма. Другата възможност, която се предлага е заплатите на администрацията да са обвързани със заплатите на работниците. За целта трябва да се изчислява средната часова ставка на всички работници, работили на обекти, след което са умножава по броя на работните часове за месеца. Отделно програмата трябва да отчита разлика в заплащането между обикновен служител и ръководител, т.е. заплатите на ръководителите за съответния отдел или строителна бригада трябва да са със фиксиран процент по-големи от заплатите на подчинените.

Също така приложението трябва да има възможност за преизчисляване на оставащите суми за работниците, механизацията и външната работа, както и общата сума за съответния обект. Тъй като няма разработен метод и поради относителността на работата по обекта не може да бъде изчислен автоматично коефициентът за обекта. Той трябва да бъде определен от човека, използващ програмата на базата най-вече на оставащата сума и субективната преценка за скоростта на дейностите по обекта. При промяна на коефициент съответно всички свързани суми трябва да се преизчислят включително и заплатите.

Програмата трябва да поддържа база данни с обектите, на които работи фирмата, имената на работниците, машинния парк на фирмата, седмичните отчети и изчислените заплати за всеки месец. На базата на тези данни могат да се правят различни справки и съответно да се разпечатват като отчети. Също така трябва да се предлага и експорт на данни в CSV формат, който е доста универсален и така да могат да се импортват в Excel, счетоводни програми или други база данни. Тъй като фирмата има отделна програма за счетоводство и изплащането на парите за заплати минават през нея това ще допринесе за по-лесното интегриране на двете програми.

Предимствата от използването на разработена по поръчка програма специално за дадената фирма са, че чрез нея може много по-гъвкаво да се определя работната заплата на всеки според заработеното и според това как върви изпълнението на обекта и да се правят детайлни справки, от които може да се установи дали нещо не върви според плана и да се вземат навременни мерки. Тази функционалност ще допринесе за по-добро планиране на средствата и да се избегнат загуби от някои обекти.

1.4 Сравнителни характеристики на подобни системи и тяхната ефективност.

Ще разгледаме някои от основните характеристики на част от предлаганите на българския пазар програми за управление на строителството и/или управление на труда и работната заплата. По-долу са представени пет от по-популярните приложения в тези области. Те имат доста възможности както ще видим от изложението. Данните за програмите и техните възможности са взети от сайтовете на производителите, а също така са разгледани и пробни версии на програмите, включващи пълна функционалност, но ограничени за ползване във времето.

Building Manager е нова генерация програма за управление на строителството [4]. На фиг. 1 е показан главния прозорец на програмата.

Номер	Поз.ПСД	Шифър	Наименование	мярка	К-во	Изп. к-во	цена	стойност	изп.ст-с
2	1 1	8101106222	ИЗКОП С ОГР.ШИРИНА ДО 1.2М - РЪЧН...	м3	200.00	0.00	0.75	149	
3	2 2	8102051320	КОФРАЖ НЕАРМИРАНИ БЕТОНОВИ СТ...	м2	200.00	0.00	0.32	65	
4	3 3	8102070150	ПОЛАГАНЕ БЕТОН М200 НЕАРМ.В ОСН...	м3	200.00	0.00	54.40	10 881	
5	4 4	8102082180	ТУХЛ.ЗИДАРИЯ НАД 1/2 ТУХЛА ОТ КУХИ ...	м3	200.00	0.00	2.36	471	
6	5 5	8102131010	ВЪТРЕШНА ВАРОВА МАЗИЛКА ПО ТУХЛ...	м2	200.00	0.00	0.13	26	
7	6 11	8102145201	ВЕНЕЦИАНСКА МОЗАЙКА ПО ПОДОВЕ...	м2	1.00	0.00	3.04	3	
8	7 12	8102121010	ФАЯНSOBA ОБЛИЦОВКА НА ВАРОЦИМ...	м2	1.00	0.00	0.72	1	
9	8 15	8102162160	ТРИКР БОЯДИСВАНЕ НОВИ СТЕНИ И Т...	м2	1.00	0.00	0.13	~0	
10	9 12	8412750000	РЪЧНО ТЪРКАНЕ НА МОЗАЙКА - РЦ	м2	1.00	0.00	0.28	~0	
11	10 14	8410130000	ПРЕХВЪРЛЯНЕ ЗЕМНИ МАСИ И СТР.ОТ...	м3	1.00	0.00	0.20	~0	
12	11 15	8102161200	ПОСТНО БОЯДИСВАНЕ ПО ШПАКЛОВ...	м2	1.00	0.00	0.02	~0	
13	12 17	8102145213	ПОЛИРАНЕ ОБИКНОВЕНИ МОЗАЕЧНИ ...	м2	1.00	0.00	0.51	1	
14	13 21	8311111631	МОНТАЖ КОНЗОЛИ,ПОДПОРИ И ДР. ДО...	тона	1.00	0.00	41.65	42	
15	14 26	8102070240	ПОЛАГАНЕ БУТОБЕТОН М150 В ОСНОВ...	м3	1.00	0.00	0.51	1	
16	15 27	8116503204	ЗАВАРКА ПЛАНКА С ВИНКЕЛ КАТЕТ 12 ...	м	1.00	0.00	0.45	~0	
17	16 28	8311112811	МОНТАЖ ГРЕДИ ДО 1 Т. ВИС. ДО 15 М. ...	тона	1.00	0.00	5.05	5	
18	17 29	*000000001	ДОСТАВКА НА МЕТАЛНА КОНСТРУКЦИЯ	тона	1.00	0.00	33.60	34	
19	18 30	8102145180	МОНТАЖ ПЛОТОВЕ ЗА ПРОЗОРЦИ И СТ...	м2	1.00	0.00	0.36	~0	
20	19 32	*000000002	ДОСТАВКА МОЗАЕЧНИ СЪПАЛА	м2	98.00	0.00	0.55	54	
21	20 34	8102131120	РОЗЕТКИ ЗА КОМИНИ	бр.	1.00	0.00	0.08	~0	
22	21 34	8430241000	ИЗМАЗВАНЕ ОТВОРИ ЗА ИНСТАЛАЦИИ ...	бр.	1.00	0.00	0.10	~0	
23	22 37	8253133200	КОЛЕНА ОТ СЕГМ. ЗА ВЦТ ДО 100 АТМ. ...	бр.	1.00	0.00	0.59	1	

Наименование	р.н-ма	мярка	цена	вал цена	об цена	коэф	ст-ст
1 ИЗКОПЧИЯ	1.924	ч.ч.	0.06	1 степ.	0.060	1.00	~0
2 ИЗКОПЧИЯ	3.855	ч.ч.	0.06	2 степ.	0.060	1.00	~0

105 поз.ПСД, 102 поз. СМР, 183 поз.мат, 16 поз. мех, 13 поз. труд

Фиг. 1 Главният прозорец на Building Manager

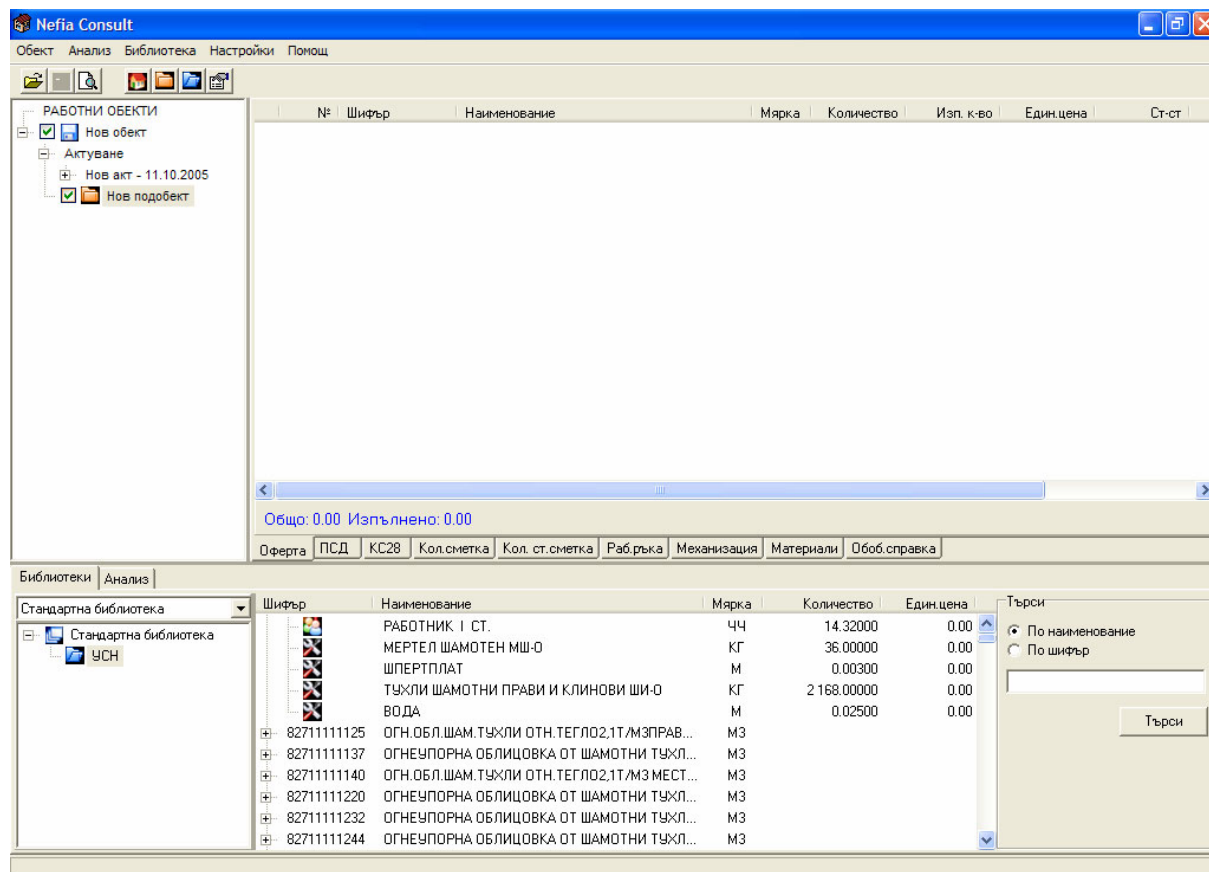
Основните възможности на Building Manager са:

- ◆ Създаване на количествени сметки по конструктивните елементи и технологията за изпълнението им
- ◆ Предоставят се данни за 28 000 вида операции по УСН и 45 000 вида по ТНС
- ◆ Остойностяване на количествените сметки. Цените на ресурсите могат да се вземат от реалните доставки от Work Flow.
- ◆ Извеждат се искания за полагащите се материали, които могат да се прехвърлят автоматично към Work Flow за осчетоводяване.
- ◆ Извеждат се маршрутни карти за полагащия се труд по операции и изделия. Отчитане на изпълнението по имена на хора и изчисление на работна карта и заплата на работника. Осчетоводяване на заработената сума по изделия.
- ◆ Отчитане на изпълнението на строителство и всички необходими справки.
- ◆ Извеждат се Отчет за изпълнение на работната стойностна сметка и Отчет за полагащите се ресурси и фактическите разходи по счетоводен път.
- ◆ Готови номенклатури със УСН и ТНС шифрирани по ЕКП / Единен каталог на продукцията.
- ◆ С лекота става въвеждането, допълването и изменянето на първичните номенклатури, които са необходими при описването и въвеждането на данните за проектите /обектите/.
- ◆ Въвеждане на номенклатурата на стандартизираните изделия и суровини т.е. тези, които се закупуват готови от външни фирми. По-нататък ще ги наричаме материали;
- ◆ Предоставя се готова номенклатура на материали шифрирана по ЕКП.
- ◆ Импорт от Excel на материали.
- ◆ Номенклатурата на изделията, които се произвеждат в самата фирма или се получават готови изработени /по индивидуален проект/ от други фирми. Импорт от Excel.
- ◆ Номенклатура на специалностите труд
- ◆ Импорт от Excel на специалностите труд
- ◆ Номенклатура на видовете услуги. Импорт от Excel.
- ◆ Номенклатура на машините.
- ◆ Импорт от Excel на наличните машини.

- ◆ Номенклатура на операциите /дейностите/, необходими за реализация на проекта или обекта. Те от своя страна могат да съдържат до пет вида ресурси – труд, машини, материали, полуфабрикати и услуги с неограничен брой от всеки вид.
- ◆ Импорт от Excel на операции.
- ◆ Създаване на конструкцията на проекта /обекта/ заедно с всички съставлящи го конструктивни елементи – подобекти, части, под части или общо наречени възли – 28 нива
- ◆ Всеки възел има собствено количество.
- ◆ Създаване на операция от възел.
- ◆ Създаване на изделия от възел.
- ◆ Поддържане на номенклатура на възлите, независимо от тяхната сложност.
- ◆ Динамично управление на структурата на проекта – местене, изтриване, копиране, размножаване, изместване по възли, преименуване и др.
- ◆ Импорт от Excel на конструкцията на проекта т.е. възлите с техните връзки.
- ◆ Въведената конструкция да се запълни със съдържание, видимо на екрана т.е. да се въведе технологията за изпълнение на всеки елемент, а това са всички операции, материали, полуфабрикати, машини, услуги и труд с техните количества необходими за реализация на даден елемент от проекта и като частен случай на изделие.
- ◆ Въвеждане на данни по няколко начина – избор от много обекти и бази едновременно, влачене с мишката от друг обект или база, избор от каталог на ресурсите.
- ◆ При избор от списък се показват всички елементи.
- ◆ От така въдените данни за проекта /обекта/ да се изчислят и изведат необходимите за реализацията му ресурси материали, труд, машини, изделия и услуги.
- ◆ Извеждане на исканията за материалите и полуфабрикатите. Исканията за труда се наричат маршрутните карти.
- ◆ Извеждане на маршрутните карти.
- ◆ Изчисляване себестойността на проекта или обекта. Извеждане на всички необходими стойностни справки.
- ◆ Динамично управление на показаната на екрана сметна документация.

- ◆ Остойносттаването става на база индивидуалните цени на ресурсите и допълнителните начисления върху тях, като за всяко количество на един и същи ресурс цената може да е различна.
- ◆ Отчитане на изпълнението на проекта или обекта /актуване/ с извеждане на всички необходими количествени и стойностни справки по отделните отчети /актове/ и обобщено за избрани.
- ◆ Програмата Билдинг Мениджър 2005 е тясно интегрирана /свързана/ със счетоводната програма Work Flow. От тази интеграция идват следните допълнителни възможности:
- ◆ При създаване на първичните номенклатури /анализи/ в БМ2005 може директно да се използват съответните номенклатури от счетоводството, като в този случай те автоматично се изравняват и могат да се използват за общи справки.
- ◆ Изведените искания за материали се предават електронно към счетоводството и там могат да се осчетоводят или от тях да се генерира заявка за доставка. Освен това се дава много ценна информация за наличните количества по складове, заявените за доставки, превишения на нормите, неотчетени искания и др.
- ◆ От изведените маршрутни карти /наряд за работа/ за труда се получават работните карти по хора. Информацията от тях се предава електронно към програмата за Заплати на Бизнессофт. Това също спестява много ръчен труд особено във фирми, където трудът се заплаща на работка. От заплатите от своя страна има автоматично осчетоводяване на сумите по обектите.
- ◆ Директно извеждане на справки за необходимите /полагащите се/ и наличните по складове материали.
- ◆ Извеждат се сравнителна справки между полагащите се материали по проект и действително вложените по счетоводен път.
- ◆ Извеждат се сравнителни справки между полагащите се материали по отчет /акт/ и действително вложените по счетоводен път.
- ◆ Извеждат се справки за полагащите се и вложените услуги.
- ◆ Извежда се стойностна сравнителна справка между планираното с БИЛДИНГ МЕНИДЖЪР 2005 и отчетеното с Work Flow.
- ◆ При калкулацията /стойносттаване/ на проекта /обекта/, цените на ресурсите могат да се вземат директно от доставните цени по счетоводен път т.е. получава се калкулация по цени на придобиване или през междинни файлове.

Nefia Consult е програмен продукт за управление на строителството[3]. На фиг. 2 е показан основния прозорец на приложението.



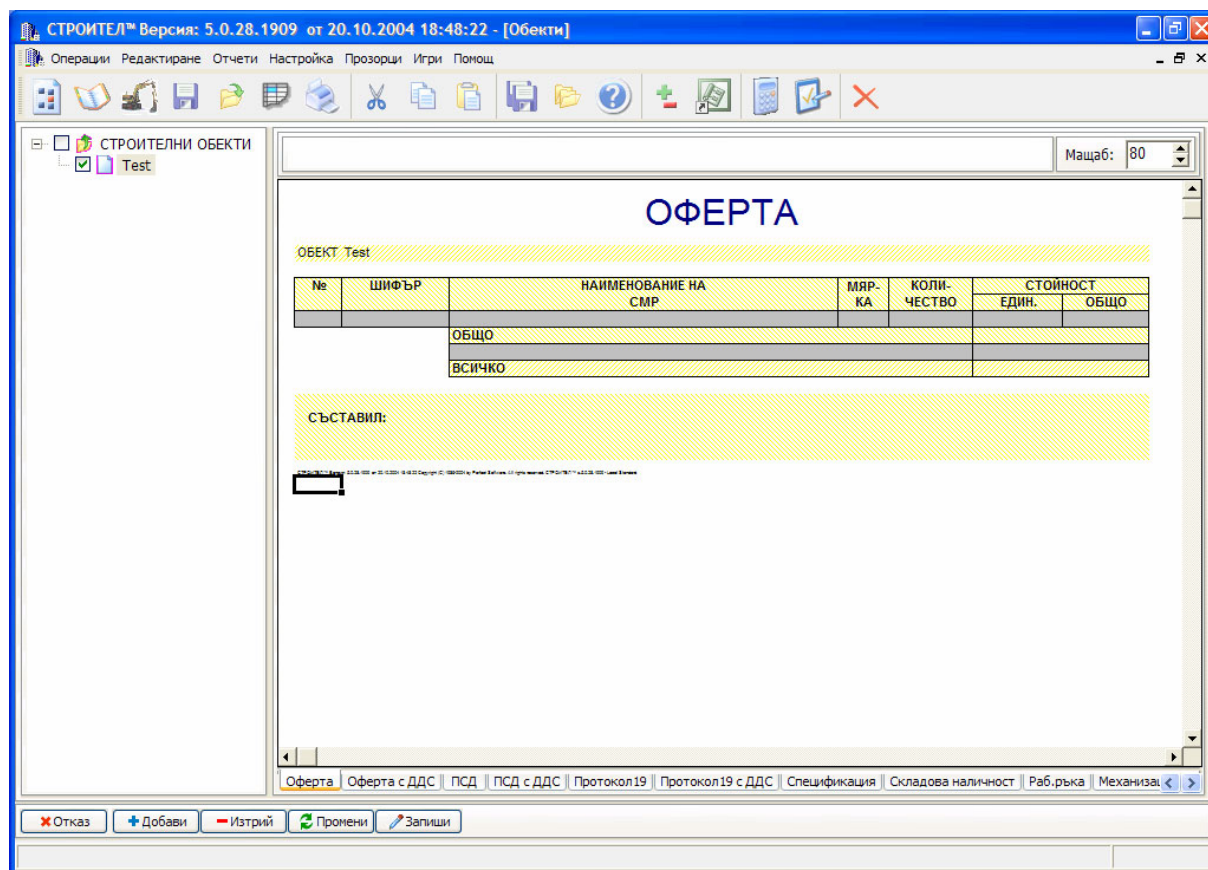
Фиг. 2 Главен прозорец на NefiaConsult

Характеристики на Nefia Consult :

- ◆ Извеждане на справки за обекта - Оферти, КС28, ПСД и др.
- ◆ Отчитане на изпълненото строителство с Протокол обр.19.
- ◆ Създаване на нови Строително Монтажни Работи (СМР), редакция и записване във фирмени сборници.
- ◆ Работа с повече от един обект едновременно.
- ◆ Нагледност и скорост на работа.
- ◆ [ново] Запис на документа като RTF. Редакция в Microsoft Word, OpenOffice.org и др.

СТРОИТЕЛ™ е първият програмен продукт в България за управление на строителството, работещ под Windows. Досега са се появили 4 версии на СТРОИТЕЛ, с многобройни модификации на всяка от тях, като всяка нова версия се отличава качествено от предишната [1].

Програмата е носител на златен медал от Международен Пловдивски панаир Есен'1998г. На фиг. 3. е показан основният прозорец на програмата.



Фиг. 3 Основен прозорец на СТРОИТЕЛ

СТРОИТЕЛ има много уникални характеристики:

- ◆ Изключителна нагледност и скорост на работа;
- ◆ Работа с документи, с външен вид идентичен на стандартните бланки;
- ◆ Позволява работа с произволни по сложност строителни обекти;
- ◆ Освен стандартните справки могат да бъдат извеждани обобщени справки за няколко обекта или за цялостната дейност на фирмата;
- ◆ Програмата предлага избор на видове СМР по УСН, ТНС, СЕК и РЕМ както и възможност за добавяне на фирмени сборници;

- ◆ Могат да бъдат създавани и укрупнени анализни цени;
- ◆ Търсенето на СМР е по произволна част от шифъра или наименованието му, като може да се търси по няколко ключови изрази, разделени с "?" или "*";
- ◆ Задаването на изчисления е решено по един изключително удобен и елегантен начин. Изборът им се извършва от списък с предварително въведени най-често използвани начисления, само с едно щракване на мишката. След това можете да промените процента на избраното начисление. Ако е необходимо и наименованието му, при което то се добавя автоматично в списъка. По този начин се избягва въвеждането на сложни и объркващи формули;
- ◆ Строителството се отчита с Протокол обр.19. При това, ако е въведена сметна документация за обекта, протоколите се генерират автоматично по оставащи количества. Програмата позволява превишаване на офертирани количества, както и добавяне на нови извънредно възникнали СМР. Може да се направи съпоставка между офертиране и изпълнение;
- ◆ Всеки обект или протокол съхранява цените на ресурсите и разходните норми на СМР влизачи в него. Възможна е промяна в разходните норми на анализ към обекта, без тази промяна да се отрази на същия анализ в сборниците със СМР;
- ◆ Цените на ресурсите в базата на програмата се актуализират автоматично в процеса на работа;

В СТРОИТЕЛ е заложен механизъм за създаване на подробни количествени сметки от проекта, който няма аналог по лекота и удобство на използване. Въведени са основните геометрични фигури и формулите за техните периметри, лица и обеми;

Към стандартния набор от документи и отчети могат да бъдат добавяни неограничен брой нови, както и да бъдат променяни съществуващите с редактора на документи. Програмата има пряка връзка с MS Excel (XLS формат) и MS Word;

За създаване на линеен или мрежов график програмата може да извежда автоматично обектите в MS Project. Това е специализирана програма за изготвяне на графици - явяващи се световен стандарт в тази област. С нейна помощ вие можете да планирате строителния процес и да извеждате диаграми на капиталовложенията и ресурсите

Microinvest ТРЗ и ЛС 3.40 е програмен продукт за управление на личния състав и заплащането на труда [2]. На фиг. 4 е показан основният прозорец на приложението.

Microinvest ТРЗ и Личен състав - [Личен картон]

Файл Операции Редакция Справки Други Прозорец Помощ

Личен картон

Трите имена: **Стоян Иванов** ЕГН:

Област: **Не е зададена** Град/село: Пощенски код:

Адрес:

Номер на ЛК: Изд. на:

Дата на раждане: Пол: **Мъж**

Постъпил на: **28.10.2005** Важи до: **28.10.2005**

Основание: **чл. ал. т.**

Кодове по НКИД: Код 1: Код 2: Код 3:

Други условия:

Отдел:

Длъжност:

Договор:

Образование:

Професия:

Обект:

Данни за октомври 2005 г.

Данни	Стойност
Основна заплата	0
Отработени дни	23
Аванс	0
Категория труд	3
МОД	0
Роден след 31 дек 1959г	НЕ
Диференциран процент ТЗПБ	0
Дневни отработени часове	8

Няма трудов стаж до момента!

Категория за сортиране:

00:24

Фиг. 4 Главен прозорец на Microinvest ТРЗ и ЛС

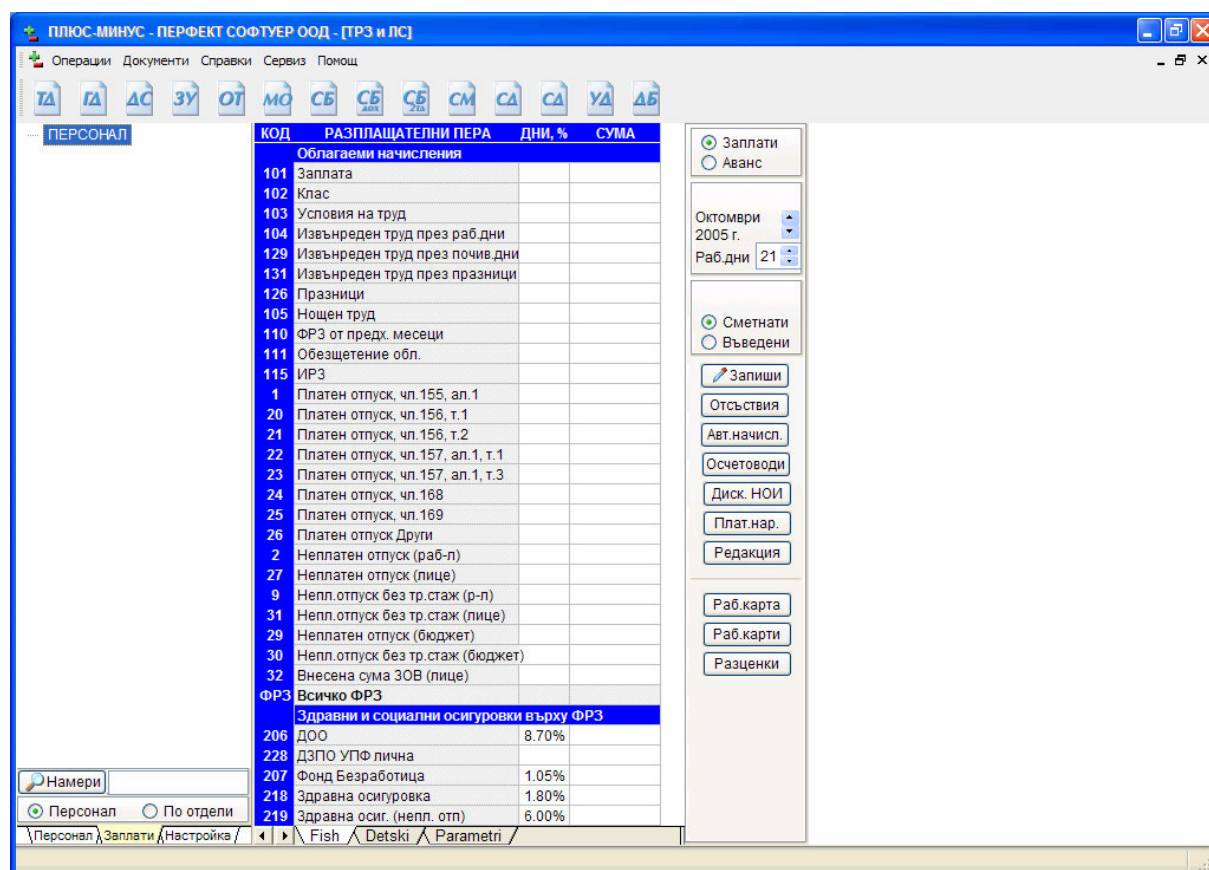
Microinvest ТРЗ и ЛС е универсална система за управление на личния състав и заплащането на труда. Програмата е многофирмена, позволява поддържане картотека на персонала, извеждане на фишове, ведомости за заплати и рекапитулации за лицата, наети по трудови правоотношения, както и възможност за настройка на системата за обработка на сумите за получаване, данъците и осигуровките за лицата, наети по граждански договори, договори за управление и контрол и самоосигуряващите се лица и автоматичен експорт на данните за осигурените лица във формата на НОИ.

Microinvest ТРЗ и ЛС притежава следните функционални възможности:

- ◆ Поддържане на лични картони с данни за работниците и служителите.
- ◆ Начисляване на заплати по стандартно зададени алгоритми.
- ◆ Възможност за задаване на потребителски показатели (начисления, удържки и др.) и схеми за тяхното изчисляване.
- ◆ Обработка на болнични и отпуски.
- ◆ Извеждане на фишове, стандартни ведомости и рекапитулации.

- ◆ Възможност за модификация на фишовете и създаване на потребителски справки, ведомости и рекапитулации.
- ◆ Експорт на дискети на декларациите за осигурените лица във формата на НОИ.
- ◆ Автоматично преизчисляване на дължимия данък на годишна база и извеждане на Служебна бележка.
- ◆ Връзка с текстовия редактор на Microinvest Офис Организатор за създаване на трудови договори, допълнителни споразумения и други документи, касаещи личния състав на фирмата.
- ◆ Програмата притежава удобен механизъм за генериране на потребителски справки с информация за минали месеци и периоди и възможност за записване на шаблона на справките за многократно използване. Microinvest ТРЗ и ЛС работи с всички принтери, инсталирани под Windows. С помощта на модула Microinvest Експортни филтри може да експортирате всички документи и справки в Word, Excel или друг текстов редактор.
- ◆ Програмата поддържа система за защита с пароли.

ПЛЮС-МИНУС™ е съвременен, професионален и напълно интегриран финансово-счетоводен офис пакет, включващ СЧЕТОВОДСТВО, СКЛАД, ТРЗ, ЛС и СЕБЕСТОЙНОСТ на произведената продукция. Всички модули са многофирмени [1]. На фиг. 5 е показан главният прозорец на модула за ТРЗ и ЛС.



Фиг. 5 Плюс-Минус – модул ТРЗ и ЛС

Модул ТРЗ и ЛС е предназначен за изчисляване на работна заплата и водене на личния състав. Той осигурява автоматизирано изчисляване на начисленията и удържките по произволни алгоритми, формиране на фишове за заплати от произволен тип; изчисляване на индивидуални и групови начисления; формиране на ведомости за заплати и рекапитулации, изчисляване на болнични и отпуски; изчисляване на щатни заплати и нормени заплатки; създаване на дискета за НОИ, статистически справки, автоматичен печат на платежни нареждания за внасяне на здравни и социални осигуровки; автоматично осчетоводяване на начислените заплати; формиране на файлове към различни банки за изплащане на заплати чрез дебитни карти; справки за личния състав; печат на трудови и граждански договори, допълнителни споразумения, молби и заповеди за отпуск, сметки за изплатени суми, служебни бележки и много други документи.

Както виждаме разгледаните програми имат доста функционални характеристики и възможности, но въпреки това не отговарят напълно на модела на работа и специфичните изисквания на фирмата, за която се разработва приложението.

Така че вместо да се пригаждат към възможностите на съществуващите програми по-добро решение е да се напише програма, отговаряща изцяло на техните спецификации на работа свързани с управлението на трудовите ресурси и определянето на работната заплата на служителите. Това ще направи процеса по-гъвкав и няма да изисква допълнителни усилия за усвояване или преминаване към нови методи за отчитане на труда и изчисляване на работната заплата.

ГЛАВА 2. Средства за реализация на системата.

2.1 Избор на среда за разработка.

Delphi – една от най-добрите разработки на фирмата Borland свързва визуалните и обектно-ориентираните принципи на програмирането с приятелска среда за разработката и като резултат дава на програмистите компоненти, които се използват многократно. Това открива нови възможности пред тях като им позволява да правят бързи разработки на усъвършенствани, качествени приложения за Windows.

Delphi е един от най-популярните продукти за RAD разработване (Rapid application development – бързо разработване на продукти) и използването му за всеобхватно разработване на приложни програми е крачка в нова епоха. Професионалните програмисти получават инструмент за виртуална разработка на програми, предназначени за крайния потребител. А за случайните програмисти – е достъпна цялата мощ на компилиращата среда. Delphi осигурява едновременно продуктивност и скорост и всичко това в една от най-съвременните среди за работа.

Операционната система Windows е най-разпространена сред работещите с персонални компютри, но разработчиците на приложения се сблъскват традиционно с трудностите на програмирането, защото организацията е необикновено сложна. За да направят програмирането в Windows достъпно за широк кръг от програмисти, много фирми, произвеждащи компилатори за програмни езици, пуснаха Windows версии на своите продукти. Фирмата Borland създаде Delphi, предлагайки визуален компилатор, построен на базата на популярния език Borland Pascal.

Разбира се на пазара съществуват и доста други възможности за разработването на едно приложение и поради тази причина в някои случаи изборът на среда и език е доста сложен. Освен многото предложения за езици трябва да се вземат под внимание и редица други фактори, а също така и клиентски изисквания. Например на каква операционна система ще се изпълнява програмата, има ли вероятност в бъдеще да се смени операционната система или да се поддържат няколко операционни системи едновременно Също така до колко критично е бързото изпълнение на някои дейности от работата на продукта и до каква степен може да бъде толерантно приложението към

грешки. Друг фактор, който също е от голямо значение е времето, за което трябва да се напише приложението и съответно опита или избора на екип за разработката. Например, ако дадена фирма има екип с дългогодишен опит в използването на някоя среда за разработване, естествено е везните да натежат в нейна посока, а не да се сформира нов екип, за да се пише приложението на друг език, макар че и това решение не е изключено в зависимост от другите фактори, влияещи на избора. Друг фактор, който може да повлияе, е типът на самото приложение, т.е. дали ще е само десктоп базирано или ще е уеб базирано, а понякога може да има модули и от двата типа.

Изискванията на клиента в нашия случай са приложението да работи под Windows операционна система. За сега няма вероятност за мигриране към друга операционна система или поддръжка на няколко едновременно. И разбира се приложението да се разработи колкото е възможно по-бързо. Няма изисквания за бързодействието на самата програма, но тъй като не е ясно колко данни ще се обработват, а пък от друга страна ще има доста изчисления върху тези данни, разбира се ще е добре и програмата да работи възможно по-бързо. Също така приложението ще е изцяло десктоп базирано с връзка към сървър за база данни и за сега не се предвиждат уеб базирани модули.

Като вземем предвид горните изисквания можем да обосновем защо Delphi е доста подходяща среда за разработка в случая. Както вече споменахме Delphi е една от най-добрите среди за RAD разработване на приложения и предоставя на програмистите компонентен модел, който е изключително добър и лесен за разработване на потребителски интерфейс, докато например при някои езици като Visual C++ правенето на потребителски интерфейс е доста трудно и отнема много усилия и ресурси. Разбира се доста популярен и добър конкурент в тази насока е Visual Basic, но той има съществен недостатък, че не предлага използването на обектно-ориентиран модел на програмиране. Разбира се може да се използва комбинация от Visual Basic и Visual C++, но това предполага доброто познаване и на двата езика или съответно повече ресурси за два отделни екипа, разработващи отделно интерфейса и функционалността на приложението. Докато Delphi съчетава едновременно и двете – бързото и лесно разработване на потребителския интерфейс и същевременно мощен компилатор с ООП.

Други възможности са модерните и широко използвани в последните няколко години езици Java и C#. Основното предимство на Java е, че може приложения, писани на него да се изпълняват на всякакви операционни системи. Но в нашия случай това не е необходимо за сега, а дори евентуално да се наложи миграция към друга операционна

система, има възможни решения, които да покрият поне следващата по-популярност операционна система Linux. Borland предлагат средата Kylix, която е еквивалент на Delphi за Linux. Така приложения, писани за Windows с Delphi много лесно могат да бъдат прекомпилирани за Linux почти без никакви промени на кода, евентуално могат да се наложат малки пренаписвания на части от кода, но това зависи от конкретната ситуация и използваните компоненти. Така че при евентуална промяна на мнението на клиента или неговите изисквания относно операционната система, най-логичния избор би бил Linux, а това както виждаме няма да представлява голяма трудност и да отнеме много време, даже почти никакво да се прехвърли приложението да работи под Linux. Друга много полезна характеристика на Java е, че могат да се пишат доста лесно уеб приложения, но в нашия случай отново клиентът няма такива нужни относно приложението. Така че и това предимство на Java няма да е от полза за конкретното приложение, което ще разработваме. Само ще отбележим, че Delphi също предлага някои възможности за разработване на уеб приложения, може би не толкова добри и универсални като Java, но ако евентуално се наложи да има в бъдеще и модули, които са уеб базирани, също могат да се използват възможностите на Delphi или разбира се да се избере друг език или технология, които ще се съвместяват с направената разработка. И да споменем, но не на последно място, че продуктът на Borland JBuilder – среда за разработка базирана на Java, също предлага доста добри възможности за правене на потребителски интерфейс и компонентен модел, но тези характеристики са присъщи на Delphi и са може би по-добре направени при него.

Един от основите недостатъци на Java е, че езикът е смесица от интерпретатор и компилатор, което му осигурява платформено независимостта, но за сметка на това го прави бавен за изпълнение в сравнение с другите езици. При нашето приложение бързодействието не е от критично значение, но все пак добре би било сметките, които се правя да се изчисляват по бързо.

C# е най-новият език за програмиране разработен от Microsoft, но въпреки скорошното му появяване той стана доста модерен и широко използван. Целта на Microsoft е да предложи алтернатива на Java, която по един или друг начин заплашва да разбие хегемонията им на част от пазара. Главният архитект на C# е човекът, разработил Delphi за Borland, когото Microsoft са привлекли за създаването на “най-добрия” език до монета. В C# се забелязват много от най-добрите черти взети от C++, Delphi и Java. А както може да се очаква от това, че човекът, стоящ в основата на разработването на Delphi, е в основата и на разработването на C#, езикът предлага

компонентен модел. С# е също като Java, смесица от интерпретатор и компилатор, което прави приложенията, писани на него до известна степен бавни за изпълнение. Но С# не е платформено независим, като Java. Въпреки това има разработки, които се стремят да направят С# работещ и на други операционни системи, каквато е почти готовата Mono.

Тъй като С# е сравнително нов език и все още няма достатъчно много компоненти, предлагани за него, за разлика от Delphi. Също така Delphi предлага много повече компоненти за връзка и работа с база данни като стандартни, разпространяващи се със средата и такива на други производители, което поне за сега го прави по-добър избор за приложения работещи с база данни. Също така С# понеже е продукт на Microsoft, предлага предимно компоненти за връзка и работа с продукти на Microsoft за база данни. А ние сме се ориентирали като сървър за базата данни да използваме MySQL, тъй като е безплатен и е най-разпространен. Също така е бърз и лесен за поддръжка.

Поради някои от недостатъците, споменати по-горе на изброените езици и поради конкретните спецификации на приложението изисквани от клиента, смятаме, че някои от предимства и характеристики на Delphi го правят най-добрия избор за разработването на тази програма.

2.2 Разработка на приложения с Delphi

2.1.1 Характерни черти на Delphi

Delphi притежава широк набор от възможности, започвайки от проектирането на форми и завършвайки с поддръжката на всички формати на популярните бази от данни [7]. Характерни черти на Delphi:

1. Многократно използвани и разширяващи се компоненти. Delphi отстранява необходимостта да се програмират такива компоненти на Windows с общо предназначение, като икони, знаци и дори диалогови прозорци. Предоставеният от Delphi внушителен списък от обекти го поставя начело сред системите, които предлагат архитектура за повторно използване на компоненти.
2. Поддръжка на стандарта VBS (Visual Basic Extensions).

3. Шаблони за приложения и форми.
4. Настройка на средата за разработване според желанията на програмиста.
5. Компилирани програми.
6. Широки възможности за достъп до данни.

В Delphi е вграден механизмът BDE за работа с (Borland Database Engine). BDE е внимателно обмислена система, която е стандартен междинен слой, използван за достъп до всички популярни формати на съвременните бази данни. BDE се използва и в системите “клиент/сървър”. BDE дава съществено предимство в производителността за сметка на по-тясната връзка с форматите на база данни.

2.1.2 Програмиране на Delphi

Преди всичко една програма на Delphi трябва да се разглежда като приложна среда, която представлява гръбнака на едно функциониращо приложение и в нея няма нищо по-конкретно. Приложната среда дава всичко необходимо за програмирането на дадено приложение.

Ключовият момент за опознаването на начина на работа на Delphi е идеята за подаване на заявки и получаване на техния отговор. Когато потребителят натисне бутон, той подава заявка или изпраща съобщение на програмата, че е настъпило дадено събитие, в резултат на което се изпълнява програмен код, който е поставен за отговор. Заявката се нарича събитие, а отговорът – процедура. Delphi генерира събития на всяко едно действие на потребителя. Цялата философия на програмирането на Delphi е създаването на програмен код, който се изпълнява като отговор на настъпването на някакво събитие [7].

Трябва да се отбележи, че не всичко, което не е видимо за нас е автоматично генериран код на Delphi. Голяма част от него на практика се включва в Windows. Програмния код за неща, като стандартни диалогови прозорци и стандартни бутони, се предоставя директно от Windows. Delphi просто ги използва чрез обръщане към съответните готови DLL библиотеки, които са инсталирани на всяка система, работеща под Windows.

Когато настъпи дадено събитие управлението на програмата се подава на подготвената процедура. Тя се изпълнява и управлението отново попада в невидимата част на приложната среда на Windows. Ако няма процедура, която се изпълнява при

настъпване на дадено събитие, просто нищо не се случва. Процедурите за обработване на настъпило събитие почти винаги са кратки и точни. Ако след настъпването на дадено събитие трябва да се запише под формата на отделна програмна библиотека. Това става като дадена процедура се обръща към процедура, записана в отделна библиотека. Така се свършва повече работа, без да се увеличава обемът на програмата, а също се намалява дублирането на кода. Т.е. така се оптимизира програмата по скорост и по големина.

Разгледаното до тук се отнася за създаването на програми по конвенционалния начин. Съществува и друга възможност да се създават собствени компоненти в Delphi.

2.1.3 Средствата на Delphi

2.1.3.1 Термини

В Delphi обектите са елементи (компоненти) – бутони, знаци, прозорци за съобщения, полета – т.е. това, което е създадено за ползване в програмите. Обектите са елементи, от които се построява нашата програма, независимо от мащаба на разглежданата задача. Обект може да бъде наречен както прост елемент /напр. рамка/, така и сложна форма, която съдържа няколко елемента. Обектите се разполагат във формата и отговарящият за тях код автоматически се записва в изходния файл на съответния програмен модул. Този код се компилира и осигурява по-голяма производителност в сравнение с визуалните среди, при които информацията се интерпретира по време на изпълнение на програмата.

- Обект – компонент или група компоненти във вида, в който съществува при изпълнение на програмата. Компонентите от палитрата са основата при построяването на обектите. Всеки обект има област на действие и тип.

- Среда (environment): 1. програмна среда – команден или пакетно-ориентиран компилатор; 2. среда за разработка – предлага завършен набор от инструменти в рамките на една програма.

- Изключение (exception) – непланирана грешка.

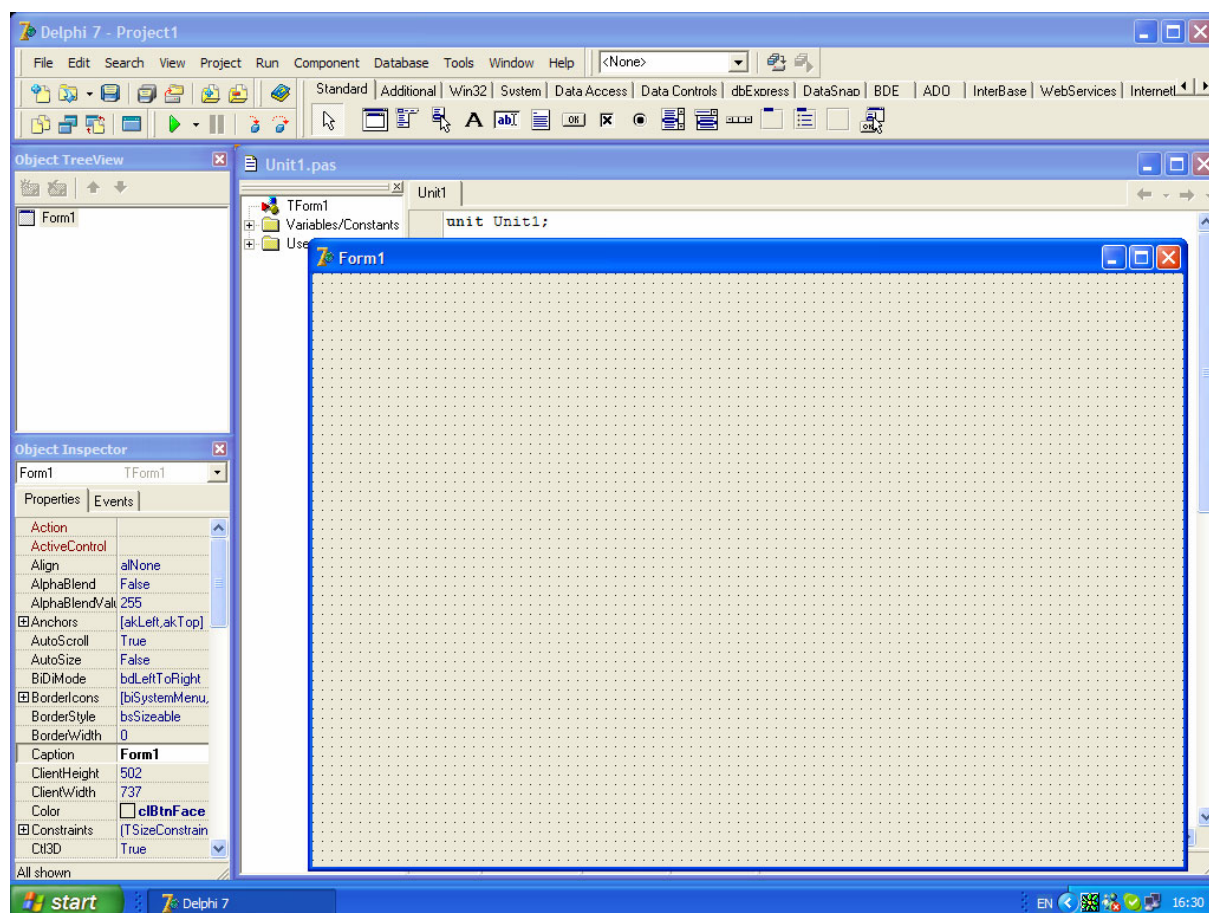
- Свойства – указват не само как изглежда обектът, но и оказват влияние върху поведението и невидимите му черти. Свойствата са непосредствено свързани с обекта – като атрибут, които описват подробно неговата структура.

- Събитие – изпълнението на голяма част от кода, написан с Delphi, ще се предизвика от събития, генерирани от потребителя или от ОС Windows. Процедурата, която се извиква при възникване на събитие, се нарича обработчик на събитие. Той е инструментът, който позволява да се свърже нашия код със събитията, протичащи около програмата.

2.1.3.2 Среда за разработка

Delphi се стартира като: ако има извадена иконка на Desktop-а се щракне върху нея, ако няма такава, тогава от Start бутона на Windows се избере менюто Programs, след което се избира програмната група на Delphi и от там се щраква върху Delphi 7.

При първоначалното зареждане на екрана се появяват четири прозореца, управлявани като многопрозоречно приложение с еднодокументен интерфейс (single document interface SDI):



Фиг. 1. Първоначален екран на Delphi

- главен прозорец на Delphi, с име “Delphi-Project 1”
- прозорец Object Inspector
- формата Form1
- прозорец с програмите озаглавен Unit1.pas

Бърз достъп към всеки от отворените прозорци на приложението: от меню View с команда Windows List се избира прозорец със списък на прозорците /Window List Dialog/ или клавиши ALT + O.

- Главен прозорец – центърът за управление на разработките в Delphi. Състои се от три области, които имат богати функционални възможности: менюто, панелът за бърз достъп /Speed Bar/ и палитрата с компоненти / Component Palette/. Менюто дава възможност за контрол на всички прозорци в средата за разработка. То е статично, т.е. не се изменя при преход от един прозорец към друг. Панелът за бърз достъп е разположен вляво от палитрата с компоненти. Той предоставя редица бутони за бърз избор на елементи от менюто с помощта на мишката. Той може да се настрои, така че да включва определени бутони. Палитрата с компоненти е каталог на визуалните и приложни обекти, които могат да се включват във формите и приложенията. Те са разположени в 8 групи: стандартна /Standart/, допълнителна /Additional/, за достъп до данни / Access/, за управление на данни /Data Controls/, диалози /Dialogs/, системна група /System/, Visual Basic (VBX) и група на примерните компоненти /Samples/. Всяка от тези групи е представена на отделна страница в палитрата. Същността на обектно-ориентираните свойства на Delphi се състои в множеството компоненти, с чиято помощ се получава достъп до стандартните възможности на Windows. В Delphi те са достъпни, чрез компоненти, “опаковани” във формите. Поместеният един път компонент става истински обект, готов за изпълнение на всяка набелязана инструкция.

- Object Inspector – той е в основата на всички действия и манипулации с обекти в Delphi. Прозорецът му се състои от следните елементи: комбиниран панел /Combo box/ за избор на обект; страница на свойствата / Properties Page/, съдържаща полета със стойностите на свойствата за текущо изображение във формата обект; страница на събитията / Events Page/, съдържаща полета с връзките между програмния код и вградените събития за избрания обект.

- Прозорец на формата – съдържа информацията, необходима на потребителя за взаимодействие с приложението по време на неговото изпълнение. В него се отразяват резултатите от работата на приложението. Всички програми на Delphi са длъжни да имат поне един обект форма.

- Редактор на програмния код – дава възможност да се редактира хода на програмата. Той е многостраничен и позволява да се осъществява преход между изходните файлове /файлове с разширение PAS/ на проекта чрез избор на съответен етикет. Обръщайки се към заглавието на прозореца, Delphi в действителност адресира файла с изходния код на формата, който се явява модул.

2.3 Релационни база данни и Delphi.

2.3.1 Основи на бази данни в Delphi

В Delphi създаването на програми за работа с бази от данни е леко и просто. Delphi предоставя всички инструменти, необходими за създаването на мощни приложения – програми, които могат да работят с множество различни типове бази от данни, реализирани на различни платформи. За създаване на бази от данни в Delphi се използват обектно-ориентирани компоненти – точно както при другите типове приложения [7]. За да се създаде приложение, работещо с бази от данни, трябва да се поместят във формата съответните компоненти и да се установят техните свойства. В Delphi може да се създават и манипулират таблици и данни, като се дописва програмен код на Object Pascal или с помощта на SQL (Structured Query Language – структуриран език за заявки/ - стандартния език за разработка на сложни операции с базите от данни.

При компютрите данните за бъдеща употреба, включително и тези от базите данни, винаги се съхраняват във файлове. Най-разпространените два подхода за такова съхранение са да се запише цялата база данни в един файл или да се запази всяка таблица, индекс и други елементи в отделни файлове, обикновено в една директория.

Delphi може да използва и двата подхода, или по-точно, използва собствен подход, който работи добре с всяка от двете структури. Построеното в Delphi приложение може да създаде данни и да получава достъп до тях самостоятелно или с помощта на други допълнителни програми за работа с бази от данни. Таблиците в Paradox и dBase записват базите данни като директории и всяка таблица е отделен файл. Access, Inter Base и повечето SQL сървъри използват един голям файл, съдържащ цялата база данни с всичките нейни индекси и таблици.

Приложенията за работа с бази данни в Delphi не взаимодействат директно с данните, към които се обръщат. Delphi взаимодейства с Borland Database Engine (BDE),

който има директен достъп до някои източници на данни като dBase, Paradox, ASCII, Fox Pro и Access. BDE може да използва SQL Links на Borland – средство, което осигурява достъп до някои локални и отдалечени SQL сървъри. Delphi се разпространява с Local Inter Base за Windows . Отдалечените сървъри включват Oracle, Sybase, Informix, Inter Base и DB2. BDE се инсталира автоматично, заедно с пакета на Delphi, който предлага версия на инсталационна програма. Файловете на BDE са задължителни – приложните програми няма да работят без тях.

2.3.2 Компоненти в Delphi за работа с бази данни

Приложенията, използващи бази от данни, работят подобно на всички други типове приложения, създадени чрез Delphi. Разликата е, че те използват някои допълнителни управляващи елементи, компоненти и класове. Компоненти са онези малки икони, които се избират от палитрата с компоненти и се поместват във формата, за да се осигурят необходимите функционални свойства на нашето приложение.

Може да се използват много компоненти на Delphi за създаването на управляващи елементи за Windows – това са интерактивни обекти, които работят подобно на обичайните диалогови елементи. Името на такъв управляващ елемент е същото като името на компонента, но без буквата T /например елемент, създаден на базата на компонента TDBEdit, ще бъде DBEdit/ всеки компонент от нашата програма използва клас на Object Pascal със същото име. Тъй като те са със същите названия като компонентите – започват с буква T /например клас на Object Pascal, използван от компонента TTable, се нарича TTable/.

В Delphi има и една специална категория компоненти за работа с бази от данни, нар. компоненти за управление на данни. Те са достъпни от страницата Data Controls в палитрата с компоненти. Те създават управляващи елементи, изглеждащи като обичайните диалогови управляващи елементи, но имат специални възможности за работа с данни.

Всеки компонент на Delphi е свързан с набор от определени методи и свойства. Когато се избере компонент във формата, неговите свойства се визуализират в инспектора на обекти.

Компонентите на Delphi за работа с бази от данни са представени в две страници на палитрата с компоненти: Data Access (фиг. 3) и Data Controls (фиг. 4)



Фиг. 3 Стр. с компоненти за достъп до данни.



Фиг. 4 Стр. с компоненти за управление на данни

Страницата Data Access съдържа компоненти за достъп до бази от данни. Тук са представени TTable, TQuery и някои други. Стр. Data Controls съдържа визуални компоненти за редактиране и разглеждане на информацията в базите данни. Тези контроли се наричат data-aware контроли.

Всяко приложение за работа с бази от данни обикновено се създава от проектна форма, съдържаща поне следните три компонента за работа с бази от данни:

1. Компонент за достъп до данните – например TTable или TQuery – който съединява нашето приложение с таблица в базата от данни / фиг. 3 /

2. На фиг. 3 това е TTable

3. Компонент TDataSource, който осигурява на приложението достъп до информацията, съдържаща се в таблица на базата данни. Всяка форма, съдържаща един или повече компоненти за управление на данни, с които потребителят може да взаимодейства, трябва да съдържа и поне 1 компонент TDataSource. Той работи като проводник между компонента TTable /или TQuery/ и един или няколко компонента за управление на данни. Когато това се наложи, те трябва да си взаимодействат чрез TDataSource. За достъп до една и съща таблица от различни форми е нужно да се помести във всяка от тях компонент TDataSource. Той ще може да използва таблици, поместени в произволна от формите.

4. Един или няколко компонента за управление на данни – TDBEdit, TDBGrid, TDBNavigator. Потребителят на нашето приложение въвежда данни, редактира ги и изпълнява други подобни операции, използвайки предоставените от нас компоненти за управление на данни. На фиг. 4 това е TDBGrid. Той създава управляващ елемент DBGrid, който не само изобразява информацията от таблицата, но и предоставя възможности за нейното редактиране.

Някои компоненти за работа с бази от данни се явяват невизуални – например на фиг. 3 TTable и TDataSource. Те са невидими за потребителя на програмата при нейното изпълнение. TTable и TDataSource са невизуални, защото тяхната работа се състои във взаимодействието на приложната програма с BDE. Компонентът TDataSource не отговаря за диалога с потребителя, а за свързването на компоненти за достъп до данни /такива като TTable или TQuery/ с компоненти за управление на данни /напр. TDBGrid/. Компоненти за управление на данни винаги се явяват визуални, защото представляват визуален интерфейс за работа с базите от данни. Те могат да бъдат не само мрежи с данни и елементи за редактиране и управление, но и други видове обекти – списъци, комбинирани панели, бутони и др.

Когато във формата се помести компонент TDataSource и се свърже с компонент TTable и с един или няколко компонента за управление на данни, потребителят получава възможност да преглежда, редактира и манипулира данните именно с помощта на предоставените от нас визуални компоненти за управление на данни.

Компонентата TDataSource, обаче не се занимава с данните директно, а е свързана с компонентата TDataSet. Тя може да представлява таблица, резултат от изпълнение на заявка, резултат от изпълнение на процедура на сървър, данни, получени от отдалечен сървър или друг собствен източник на данни. След като се постави TTable или TQuery във формата се използва характеристиката TDataSet на компонентата TDataSource за обръщение към нея. За тази характеристика Object Inspector показва всички налични източници на данни в текущата форма или в други форми, свързани с текущата.

Най-простият начин да се зададе достъп до база данни в Delphi е да се използва компонентата TTable. Обектът TTable просто сочи към таблица в базата данни. Когато се използва компонентата TTable, трябва да се зададе името на базата данни, която ще се използва като стойност на характеристиката Data Base Name. Може да се въведе или псевдоним на база данни, или пътя до директорията, съдържаща таблицата. Object Inspector показва наличните имена на бази данни, които са псевдоними на бази данни, инсталирани в BDE. Трябва да се зададе и стойност на характеристиката TableName. Object Inspector показва всички налични табл. в текущата база данни /или директория/, така че първо се избира стойност на DataBaseName и след това на TableName.

Вторият източник на данни в Delphi е компонентата TQuery. Заявката обикновено е по-сложна от таблица, защото тя изисква команди на езика SQL. Компонентата TQuery има характеристика DataBaseName, както TTable, но няма

характеристика TableName. Таблицата се задава в SQL командите, съхранявани в характеристиката SQL. Например може да се напише следния прост израз на езика SQL:

Select * From Country, където Country е име на таблица и символът, * показва да се използват всички полета от таблицата. Компонентата TTable обикновено е по-бърза за локални таблици, докато TQuery е по-бърза при работа с SQL сървъри. Ефективността на таблиците или заявките зависи от базата с данни, с която се работи.

Един важен компонент, който обикновено работи “зад кулисите”, е TField, представляващ сам по себе си обектно-ориентирана инкапсулация на колона от таблица в базата от данни. Той има свойства, описващи типа на данните в колоната, текущата стойност, формата на изхода, формата за редактиране и някои други атрибути. Когато с помощта на TTable или TQuery се получи таблица от бази данни, във всяка нейна колона има отделен компонент TField, който Delphi създава автоматично и който се основава на физическата структура на конкретната таблица.

Тъй като Delphi създава компонентите TField динамично, обектите TField, описващи колоните, се сменят всеки път, когато се сменя и структурата на таблицата. Например, ако дадена колона се премахне от таблицата, с която работи нашето приложение, то при следващото му стартиране в компонента TField на таблицата ще отсъства информация за тази колона. Това означава, че програмата ни може да работи неправилно, опитвайки се да получи данни от несъществуваща колона. За да се избегнат такива ситуации, може за всяка използвана таблица да се създадат статични компоненти TField. Така Delphi няма да генерира динамично обектите TField, а ще използва зададените от нас. За създаване на статични компоненти TField Delphi предоставя специалния инструмент DataSet Designer. С негова помощ могат да се избират колоните, които да се изобразяват в мрежата с данни и да се управлява наредбата им при показване.

При работа с източник на данни в Delphi може да се работи в различни състояния, показвани от специална характеристика State, която може да приема няколко различни стойности:

= dsEdit показва, че източникът на данни е в режим на редактиране.

Това става когато програмата извика метода Edit или характеристиката AutoEdit на компонентата TDataSource има стойност True и потребителят е започнал да редактира съдържанието на контролата, свързана с базата данни /например TDBGrid и

TDBEdit/. Когато променените записи се изпратят към базата данни, източникът на данни излиза от състоянието dsEdit.

= dsBrowse показва, че източникът на данни е в нормален режим за разглеждане, използван за търсене на данни и обхождане на записите.

= dsInsert показва, че към източникът на данни ще се добави нов запис.

Това става чрез метода Insert при преместване на последния ред на компонентата TDBGrid или чрез използването на съответната команда на компонентата TDBNavigator.

= dsInactive показва, че източникът на данни е затворен.

= dsSetKey показва, че се подготвя търсене в източника на данни.

Това е състоянието между извикването на метода SetKey и извикването на методите GotoKey и GotoNearest.

= dsCalcFields е състояние на полетата, тоест по време на извикването на обработчика на събитието OnCalcFields.

= dsNewValue, dsOldValue, dsCurValue са състояния на източника на данни, когато в момента се извършват обновявания на кеша.

= dsFilter е състояние на източника на данни при задаване на филтър, т.е. по време на извикването на обработчика на събитието OnFilterRecord.

Освен компонентите за връзка с бази данни TTable, TQuery и TDataSet има и няколко други такива в страницата Data Access на палитрата с компоненти:

= TDataBase – за контрол на транзакциите, сигурност и контрол на връзката. Най-общо тя се използва за свързване с отдалечени бази данни в клиент /сървър приложенията или да се предотвратят повтарящи се връзки към една и съща база данни от различни форми. Използва се и за локален псевдоним на базата, използван само в програмата. След като веднъж той е свързан с даден път, компонентите TTable и TQuery в приложението могат да използват локалния псевдоним.

= Session – предлага глобален контрол над връзките с базите данни за едно приложение, включително списък от съществуващите бази данни и псевдоними, както и събитие за промяна на въвеждането на парола за връзка /login/ с базата данни.

= BatchMode – за извършване на пакетни операции като копиране, добавяне, обновяване или изтриване на стойности върху една или повече бази данни.

= UpdateSQL – за писане на SQL команди, за да се извършат различните операции за обновяване на източника на данни, когато се използва read-only заявка.

Data Aware контроли в Delphi от стр. Data Controls:

= DBGrid представлява графична таблица, която може да покаже цялата таблица наведнъж. Тя позволява превъртане и преминаване между записите, както и редактиране на нейното съдържание.

= DBNavigator – набор от бутони, използвани за предвижване и осъществяване на операции върху базата данни.

= DBText показва съдържанието на поле, което не може да бъде променяно. Това е data aware версията на контролата Label.

= DBEdit позволява на потребителя да редактира поле /да променя текущата му стойност/, използвайки текстово поле.

= DBMemo позволява на потребителя да променя големи текстови полета, съхранени в поле от тип Memo или BLOB (Binary Large Object).

= DBImage е разширение на компонентата Image , която показва изображение, съхранено в поле от тип BLOB.

= DBListBox и DBComboBox позволяват на потребителя да избира единични стойности от определено множество.

= DBCheckBox- за показване и превключване на опция, отговаряща на поле от тип Boolean в таблицата. Тя е разширение на компонентата CheckBox.

= DBRadioGroup предлага набор от избори чрез взаимно изключващи се радио бутони, както стандартната компонента RadioGroup.

= DBRichEdit – позволява на потребителя да редактира форматиран текстов файл. Базиран е на стандартната за Window 95 контрола RichEdit.

= DBCtrlGrid е графична таблица за представяне на множество записи, която може да съдържа няколко data aware контроли.

= DBChart е data aware версията на компонентата Chart.

Всички тези компоненти се свързват с източника на данни чрез характеристиката си DataSource. Някои от тях са свързани с цялото множество от данни, като компонентите DBGrid и DBNavigator, докато други се отнасят за конкретно поле на източника на данни, задавано чрез характеристиката DataField. След като веднъж е зададена стойност на характеристиката DataSource, характеристиката DataField ще покаже списък с възможните си стойности в комбиниран списък в Object Inspector.

Също така освен изброените по-горе компоненти, Borland в различните версии на Delphi предложи допълнителни палитри с компоненти за връзка към база данни.

Такива са ADO и dbExpress. Като ADO поддържа технологиите на Microsoft за връзка и работа с база данни, а dbExpress предлага директна връзка към някои сървъри, като например MySQL, Oracle, DB2 и др.

2.3.3 Компоненти на други разработчици.

Работа с BDE има доста предимства, но и някои недостатъци, като например смятаната за бавна връзка през ODBC. Също така друг недостатък е дистрибуцията ѝ, която изисква около 6 MB допълнителна инсталация. Поради тези и причини ние ще потърсим други възможности за връзка към базата данни. Както вече казахме стандартно с Delphi се предлагат ADO компоненти, но за да се вържем с тях към MySQL трябва да използваме или отново ODBC драйвер или OLEDB драйвер. Единственият разработен такъв е MyOLEDB, който за съжаление от няколко години не се поддържа и няма нови разработки за последните версии на MySQL и ADO. Инсталирането на стара версия създава проблеми и не може да тръгне. Друга алтернатива на BDE предлагана стандартно са dbExpress компонентите. Тъй като има някои особености при работа с тях, като например начина на свързване между самите компоненти, а също така част от функционалността, която предлагат е малко по различна, може да се каже че те са по-подходящи за приложения с многослойна архитектура.

Ние ще се спрем на използването на компоненти от други производители, които се инсталират допълнително към Delphi. За връзка и работа с MySQL сървър има над десет различни пакета с компоненти като някои от по известните са: MySQLDAC, ZeosLib, TMySQL, MyDAC и др. Част от тези компоненти са комерсиални разработки и съответно са платени, а друга част са с отворен код и безплатни. Ние ще се спрем на ZeosLib компонентите, не само защото са безплатни и с отворен код, а имат и други предимства пред останалите.

Пакетът с компоненти ZeosLib поддържа няколко различни среди за разработка на Borland, а също така и FreePascal. Компиляторите, за които е наличен, са:

- Delphi 5 - 7 и 9
- Lazarus (FreePascal)
- C++ Builder 5 - 6

- Kylix 2 - 3

Освен че поддържа няколко различни езика и среди, също така освен връзка към MySQL има възможности за работа с доста други сървъри. Директна връзка може да се осъществи към следните сървъри:

- MySQL 3.20 - 4.1
- PostgreSQL 6.5 - 7.4
- Firebird 1.0 - 1.5
- Interbase 5.0 - 7.5
- Microsoft SQL Server 7, 2000
- Sybase ASE 12.0, 12.5
- Oracle 9i
- SQLite 2.8

От този списък половината сървъри са безплатни и с отворен код, така че при евентуална промяна на изискванията на клиента евентуално може да се мине и на друг безплатен сървър без никаква промяна в кода, единствено като се променят свойствата на компонента за връзка TZConnection. Ако въпреки това се наложи използването на друг сървър, който не е в този списък, се предлага възможност за връзка към него през така наречения ADO Bridge.

Освен поддръжката на много сървъри и няколко среди за разработка друго голямо предимство е изцяло покриване на възможностите на TDataSet и предлагане на допълнителна функционалност.

Пакетът съдържа следните компоненти [6]:

1. TZConnection: този компонент е обвивка на връзката с базата и управлението на транзакциите
2. TZReadOnlyQuery: наследник на TDataset служещ за изпълнение на заявки и обработка на данни във режим само за четене.
3. TZQuery: също наследник на TDataset, който позволява модифициране на данните в нормален и кеширан режим.
4. TZUpdateSQL: аналог на стандартния TUpdateSQL компонент за дефиниране на Insert/Update/Delete заявки, които да модифицират даден TDataset.
5. TZStoredProc: този компонент изпълнява SQL според процедури.

6. TZSQLProcessor: компонент за изпълнение на SQL скриптове на различни сървъри.
7. TZSQLMonitor: този компонент следи всички изпълняващи се заявки и също така информация, която може да се записва.
8. TZSQLMetadata: специализиран TDataset компонент, който дава достъп до мета данни, като таблици, колони, индекси и др.

ГЛАВА 3. Проектиране и архитектура на системата

3.1 Проектиране на базата данни и схема на таблиците

Създаването на качествен продукт, отговарящ на всички изисквания на клиента, обуславя проектирането, разработването на логически модел на данните и изграждането на подходяща релационна база данни.

При създаване на модела на данните се извършва организиране на изискванията на системата в едно логическо представяне на базата данни. Моделът на данните се състои от обекти и техните атрибути и ограничения, дефиниции на релации между обектите и ограничения върху тези релации, следователно разработването на логически модел на данните се осъществява чрез определяне на обекти, техните атрибути и ограничения и техните релации. Създаденият по този начин модел позволява да се идентифицират съответните таблици, които трябва да бъдат създадени, колоните, съдържащи се в тях, релациите между тях.

За определяне на обектите е необходимо от изискванията на системата, да се дефинират основните логически подразделения на информацията. В случая системата е поръчана от фирма в строителния сектор за определяне на труда и работната заплата на служителите. Разглеждайки изискванията на системата, дефинираме следните основни обекти: служители, строителни обекти и заплати. Тъй като фирмата разполага със собствени машини, които изискват ресурси, се налага дефинирането на още един обект – машини. Понякога фирмата наема и външни работници, чиито заплати са предварително определени и затова дефинираме пети обект – външни работници. В резултат на това правим следните таблици в дизайна на базата данни, които им съответстват - Workers, Objects, Machines, Salaries, ExtrnWork.

След като сме дефинирали основните таблици, които могат да бъдат определени в този момент, трябва да дефинираме колоните (атрибутите) на тези таблици. Тази информация вземаме директно от изискванията на системата, в които е определено какви данни трябва да се поддържат за обектите.

Обектът служители трябва да съдържа информация за имената на служителите, каква длъжност заемат, към кой отдел са, основна заплата, клас квалификация, средна

заплата на час, брой месеци, през които е работил/работи във фирмата и дали все още работи в нея. Вследствие на това добавяме колони към таблицата Workers за всеки от тези видове информация. Освен това добавяме уникален идентификатор за всеки служител, както се прави при всички нормализирани обекти.

Вглеждайки се по-внимателно в атрибутите на тази таблица, забелязваме, че имаме две колони – длъжност и отдел. В съответната фирма има определен брой длъжности и отдели. В следствие на това създаваме още две таблици - Positions и Departments. В таблицата Positions ще се съхранява информация за названията на длъжностите. Към тази таблица добавяме два атрибута: уникален идентификатор и название на длъжност. В таблицата Departments ще се съхранява информация за названията на отделите. Към тази таблица добавяме два атрибута: уникален идентификатор и название на отдела. Тогава атрибутите за каква длъжност заемат и към кой отдел са служителите в таблицата Workers ще са препратки към съответната таблица, без да е необходимо да се повтаря названието на длъжността и отдела за всеки служител.

Таблицата Objects трябва да съдържа информация за името на обекта, стартова дата, крайна дата, общ брой необходими човекочаса, цена на човекочас, сума за работа на машините, типа и статуса на обекта. Вследствие на това добавяме колони към таблицата Objects за всеки от тези видове информация и уникален идентификатор за всеки обект.

Като се вгледаме по-внимателно в атрибутите на този обект и взимайки под внимание изискванията на системата, забелязваме, че строителните обектите могат да бъдат няколко типа и възможните статуси на строителните обектите също са няколко, на съответен строителен обект може да се ползват или да не се ползват машини, ако е необходимо да се наемат външни работници и да се води отчет на всеки обект за всеки месец. Следователно трябва да имаме препратки от и към други таблици.

На базата на този извод създаваме обектите: статус и тип. В резултат на това добавяме таблици към базата данни, които им съответстват – ObjectStatus и ObjectType. Така атрибутите за типа на обекта и статуса на обекта в таблицата Objects ще са препратки съответно към ObjectStatus и ObjectType.

В таблицата ObjectType ще съхраняваме информация за типа на обекта и в нея създаваме два атрибута: уникален идентификатор, название на типа на обект и описание на типа.

В таблицата ObjectStatus ще съхраняваме информация за статуса на обекта и в нея създаваме два атрибута: уникален идентификатор, название на статуса на обект и описание на статуса.

За да се води месечен отчет на строителните обекти се налага да дефинираме допълнителен обект, който да има препратка към таблицата Objects. Обектът за месечните отчети на строителните обекти трябва да съдържа информация за съответния месец, коефициент за извършената работа (дали има преизпълнение, забавяне или работата върви по план), средства изразходвани за служителите на фирмата, машините и външните работници, остатък по договорената сума за обекта и препратка към таблицата Objects. Като резултат добавяме таблица към дизайна на базата, със съответните колони и уникален идентификатор. Тази таблица наричаме ObjMonthlyData.

Както вече споменахме фирмата ползва и външни работници или фирми подизпълнители за да извършват конкретна работа на даден строителен обект. Обектът, в който се съхранява информация за тези работници или фирми, трябва да притежава следните атрибути: начална и крайна дата на наемането на работника или фирмата, изплатена сума и възможност за коментар. Така ще имаме таблица, в която се пазят данни за единица извършена работа от външен изпълнител, но няма нужда да пазим информация за него, а само за сумата взета от него за конкретната услуга. Вследствие на това добавяме колони към таблицата ExtrnWork, която вече сме създали, за всеки от тези видове информация. Освен това добавяме уникален идентификатор за всеки работник и препратка към таблицата Objects.

Обектът машини трябва да съдържа информация за името на машината, каква е цената за изминат километър от машината, която се заплаща при придвижването и до обекта, каква е цената на час за работа на машината, също така ако машината извършва и други дейности освен основната ѝ – трябва да има цена на час и за друга дейност. Вследствие на това добавяме колони към таблицата Machines за всеки от тези видове информация. Освен това добавяме уникален идентификатор за всяка машина, както се прави при всички нормализирани обекти. Също така трябва да добавим още една колона, която да индикира дали дадената машина е изтрита или не, т.е. това се прави с цел информацията за нея да не се показва на потребителя, но да се пази в базата данни и при необходимост да се използва например при генериране на отчети за предишни месеци, в които тя е участвала преди да бъде извадена от машинния парк на фирмата и съответно да се наложи изтриването ѝ.

Таблицата Salaries трябва да съдържа информация за заплатата на всеки служител за даден месец, средното му трудово възнаграждение на час, заработена сума, глоби, бонуси, допълнителни възнаграждения, допълнителен фактор, крайно възнаграждение и коментар. Вследствие на това добавяме колони към таблицата Salaries за всеки от тези видове информация, уникален идентификатор за всяка заплата и препратка към таблицата Workers.

Вглеждайки се по-внимателно в атрибутите на тази таблица, забелязваме, че имаме колона допълнителен фактор. В съответната фирма има определен брой допълнителни фактори. В следствие на това създаваме нова таблица – SalaryCoeff, в която ще се съхранява информация за тези коефициенти. Към тази таблица добавяме три атрибута: уникален идентификатор, коефициента, който ще се използва в приложението и съответната му стойност, която ще се използва за изчисления. Тогава атрибутът за допълнителния коефициент в таблицата Salaries ще е препратка към таблицата SalaryCoeff.

След като сме определили обектите и техните атрибути в модела трябва да определим релациите между обектите и техният тип. Ще започнем от таблицата Workers. Всеки служител във фирмата заема някаква длъжност, следователно има релация между таблиците Workers и Positions. При това всеки служител може да заема само една длъжност и много служители да имат една и съща длъжност, следователно между таблиците Positions и Workers съществува релация „едно към много”.

Аналогична е ситуацията и с таблиците Workers и Departments. Всеки служител във фирмата е назначен към даден отдел, следователно има релация между таблиците Workers и Departments. При това всеки служител може да е само към един отдел и един или повече служители да са назначени в този отдел, следователно между таблиците Departments и Workers съществува релация „едно към много”.

Всеки служител получава заплата, следователно има релация между таблиците Workers и Salaries. Служителите всеки месец получават заплати, т.е. в таблицата Salaries има няколко записа за всеки служител, но във всеки запис за заплата трябва да има точно едно име, следователно между таблиците Workers и Salaries съществува релация „едно към много”.

Както вече споменахме от таблицата Salaries има препратка към таблицата SalaryCoeff, т.е. има релация между двете таблици. За изчислението на всяка заплата може да се ползва само един коефициент, но един коефициент може да се ползва за

изчислението на много заплати, следователно релацията между таблиците SalaryCoeff и Salaries е „едно към много”.

Понеже фирмата е в строителния бранш, основната част от служителите са строители, които работят на строителни обекти, следователно имаме релация между таблиците Workers и Objects. Един служител може да работи на един или повече обекти (по различно време) и на един обект може да работи един или повече служители, то между тези таблици съществува релация „много към много”. В един нормализиран проект на базата данни релациите „много към много” трябва да се модифицират, като се добави свързваща таблица и между нея и всяка първоначална таблица трябва да се създадат релации „едно към много”.

На тази база в базата данни добавяме нова таблица и предвид особеностите на системата, в нея освен атрибутите за препратки към първоначалните таблици - Workers и Objects – ще добавим атрибути, които да съхраняват информация за начална и крайна дата на заетостта на даден работник на даден обект, работни часове в работно време, работни часове в извънработно време, работни часове в празнични и почивни дни и уникален идентификатор. Както вече казахме, тя ще има релации с таблиците Workers и Objects и тези релации ще са „едно към много”. Тази таблица ще я наречем Earnings.

Таблицата Objects има релация не само с таблицата Earnings, но и с други таблици.

Всеки строителен обект е от някакъв тип, следователно има релация между таблиците Objects и ObjectType. При това всеки обект може да е само от един тип и много обекти да са от един и същи тип, следователно между таблиците ObjectType и Objects съществува релация „едно към много”.

Всеки строителен обект има статус, следователно има релация между таблиците Objects и ObjectStatus. При това всеки обект може да има само един статус и много обекти да имат еднакъв статус, следователно между таблиците ObjectStatus и Objects съществува релация „едно към много”.

За някои строителни обекти могат да се наемат външни работници, следователно има релация между таблиците Objects и ExtrnWork. За даден строителен обект могат да се наемат един или повече външни работници, но те работят само на този обект, за който са наети, следователно релацията между таблиците Objects и ExtrnWork е „едно към много”.

Фирмата води отчети за всеки обект за всеки месец, следователно има релация между таблиците Objects и ObjMonthlyData. Понеже в таблицата ObjMonthlyData има

един запис за всеки обект за всеки месец, но в нея има много записи за всеки обект, но за различните месеци, правим извода, че между таблиците ObjMonthlyData и Objects релацията е „едно към много”.

На различните строителни обекти могат да се ползват (или да не се ползват) машини, следователно има релация между таблиците Objects и Machines. Една машина може да се използва на един или повече обекти (по различно време) и на един обект може да се ползват една или повече машини, то между тези таблици съществува релация „много към много”. За да проектираме нормализиран проект на базата данни тази релация „много към много” трябва да се модифицира, като се добави свързваща таблица и между нея и всяка първоначална таблица трябва да се създадат релации „едно към много”.

На тази база в базата данни добавяме нова таблица и предвид особеностите на системата, в нея освен атрибутите за препратки към първоначалните таблици - Objects и Machines – ще добавим атрибути, които да съхраняват информация за начална и крайна дата на използването на дадена машина на даден обект, брой изминати километри за транспортирането на дадената машина до строителния обект, работни часове на машината, през които се използва за основната си предназначена дейност, работни часове, през които се използва за допълнителни дейности (ако има възможност) и уникален идентификатор. Както вече казахме, тя ще има релации с таблиците Machines и Objects и тези релации ще са „едно към много”. Тази таблица ще я наречем MachWork.

На базата на логическия модел, който създадохме проектираната база данни ще съдържа следните таблици: Workers, Objects, Salaries, SalaryCoeff, Positions, Departments, Earnings, ExtrnWork, ObjMonthlyData, ObjectStatus, ObjectType, Machines и MachWork. Всяка таблица трябва да съдържа атрибути (колони), първичен ключ, външни ключове и релации (връзки) между таблиците,

Атрибутите на таблицата Workers са: ID, Name, IDPosition, IDDepartment, BaseSalary, ClassCF, AveragePH, NumMonths, IsDeleted. Първичният ключ в тази таблица е атрибутът ID, той ни дава уникален идентификатор на всеки служител. Атрибутът IDPosition е външен ключ към таблицата Positions. Релацията между двете таблици е „едно към много”, като таблицата Workers е от страната „много” на релацията, а таблицата Positions е от страната „едно” на релацията. Атрибутът IDDepartment е външен ключ към таблицата Departments, като таблицата Workers е от страната „много”, а таблицата Departments е от страната „едно” на релацията.

Атрибутите на таблицата Positions са: PositionID и Name. PositionID е първичен ключ.

Атрибутите на таблицата Departments са DepartmentID и Name. В таблицата първичен ключ е DepartmentID.

Таблицата Salaries има следните атрибути: IDsalaries, IDworker, Month, AverageWPH, Salary, Penaly, Bonus, Additional, IDfactor, Sum, Comment. IDsalaries е първичен ключ на тази таблица, той определя уникален идентификатор на всяка заплата. Атрибутът IDworker се явява външен ключ към таблицата Workers. Имаме релация „едно към много”, като от страната „едно” на тази релация се намира таблицата Workers, а от страната „много” – таблицата Salaries. Атрибутът IDfactor също е външен ключ. Той служи за релация „едно към много” с таблица SalaryCoeff. Страната „едно” в тази релация е таблицата SalaryCoeff, а страната много е таблицата Salaries.

Атрибутите на таблицата SalaryCoeff са FactorID, KoefDispley и Factor. Първичният ключ на тази таблица е атрибутът FactorID.

Атрибутите на таблицата Earnings са: IDearn, IDworker, IDobject, StartDate, EndDate, Hours, OverHours и HolidayHours. Първичният ключ на тази таблица е IDearn – уникален идентификатор на всяка заработка. Атрибутите IDworker и IDobject са външни ключове. IDobject е външен ключ към таблицата Objects, а IDworker е външен ключ към таблицата Workers. Следователно ще имаме две релации „едно към много”, като от страните „много” на тези две релации ще се намира таблицата Earnings, а от страните „едно” – Workers и Objects.

Атрибутите на таблицата Objects са: ObjectID, Name, StartDate, EndDate, PPH, WPH, Machinery, Status и Type. Първичният ключ на тази таблица е ObjectID, в него се съхраняват уникалните идентификатори на обектите. Тази таблица е в релация не само с таблицата Earnings. Тя седи от страната „много” на релациите с таблиците ObjectStatus и ObjectType, и от страната „едно” на релациите с таблиците: ExtrnWork, ObjMonthlyData и MachWork.

Атрибутите на таблицата ObjectStatus са StatusID, Name и Description. Първичният ключ на тази таблица е атрибутът StatusID. Тази таблица седи от страната „едно” на релацията с таблицата Objects, а таблицата Objects е от страната „много”.

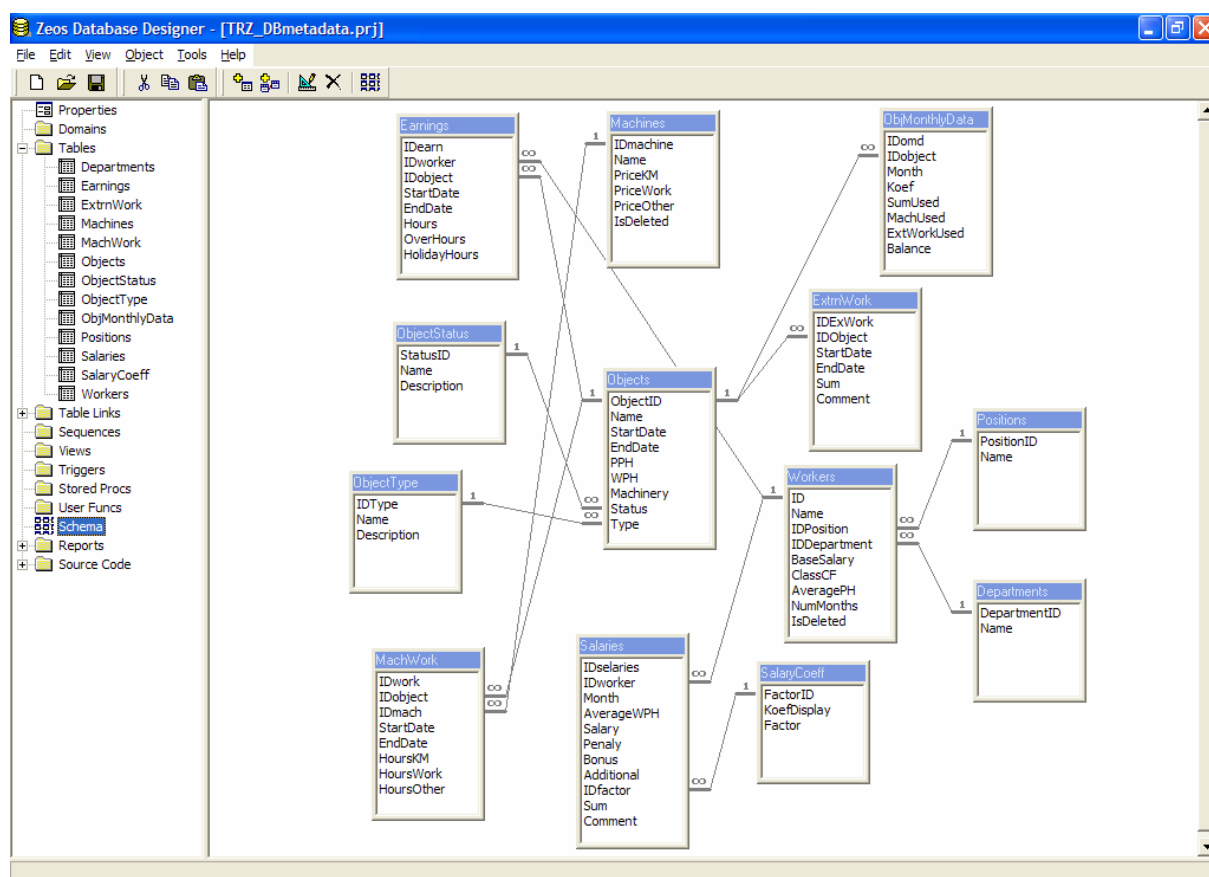
Атрибутите на таблицата ObjectType са IDType, Name и Description. Първичният ключ на тази таблица е атрибутът IDType. Тази таблица седи от страната „едно” на релацията с таблицата Objects, а таблицата Objects е от страната „много”.

Таблицата ExtrnWork има следните атрибути: IDExWork, IDObject, StartDate, EndDate, Sum и Comment. IDExWork е първичен ключ на тази таблица, той определя уникален идентификатор на всяка „външна” работа. Атрибутът IDObject се явява външен ключ към таблицата Objects. Имаме релация „едно към много”, като от страната „едно” на тази релация се намира таблицата Objects, а от страната „много” – таблицата ExtrnWork.

Таблицата ObjMonthlyData има следните атрибути: IDomd, IDObject, Month, Koef, SumUsed, MachUsed, ExtWorkUsed и Balance. IDomd е първичен ключ на тази таблица, той определя уникален идентификатор на всеки месечен отчет за строителните обекти. Атрибутът IDObject се явява външен ключ към таблицата Objects. Имаме релация „едно към много”, като от страната „едно” на тази релация се намира таблицата Objects, а от страната „много” – таблицата ObjMonthlyData.

Атрибутите на таблицата MachWork са IDwork, IDobject, IDmach, StartDate, EndDate, HoursKM, HoursWork и HoursOther. Първичният ключ на тази таблица е атрибутът IDwork. Атрибутът IDobject се явява външен ключ към таблицата Objects. Релацията между двете таблици е „едно към много”. Тази таблица седи от страната „много” на релацията с таблицата Objects, а таблицата Objects е от страната „едно”. В тази таблица има още един външен ключ, това е атрибутът IDmach, той е към таблицата Machines. Релацията между двете таблици е „едно към много”. Таблицата MachWork седи от страната „много” на релацията с таблицата Machines, а таблицата Machines е от страната „едно”.

Таблицата Machines има следните атрибути: IDmachine, Name, PriceKM, PriceWork, PriceOther и IsDeleted. Първичният ключ на тази таблица е атрибутът IDmachine. Тази таблица седи от страната „едно” на релацията с таблицата MachWork, а таблицата MachWork е от страната „много”.



Фиг. 1. Схема на базата данни

3.2 Проектиране на потребителския интерфейс

Една от основните части на приложението е потребителският интерфейс, през който се осъществява комуникацията между програмата и човека. Именно поради тази му особеност той трябва да е максимално удобен и лесен за използване. В наши дни компютрите са достатъчно бързи и мощни, за да позволят проектирането и разработването на добре изглеждащ графичен интерфейс, за разлика от преди няколко десетилетия, когато основно се изискваше бързо действие на алгоритмите в програмата и почти не се обръщаше внимание на интерфейса. Тогава и операционните системи не бяха достатъчно развити и най-доброто, което можеше да се постигне е разработването на псевдографичен интерфейс. Сега те са много по-развити и предлагат лесно изграждане и управление на графичен интерфейс в основата на който стоят така наречените прозорци.

Ние сме избрали за операционна система Windows, под която ще работи приложението и в нея съществуват два основни типа интерфейс: едно-документен интерфейс (SDI – Single Document Interface) и много-документен интерфейс (MDI –

Multiple Document Interface). Едно-документния интерфейс позволява на едно стартирано копие на приложението да отвори само един документ в даден момент, т.е. потребителя може да работи само в един прозорец на програмата и само той да е активния прозорец, в който са заредени данни за обработване. При много-документния интерфейс се предлага възможност да има няколко отворени прозореца в рамките на главния, в които да са заредени толкова на брой документа/изгледа. Това позволява на потребителя да превключва лесно между самите обработвани документи, вместо всеки път наново да ги зарежда. Това в някой случаи е голямо удобство, но в други може да доведе до объркване на потребителя, особено ако той е по-неопитен. Следвайки последните тенденции и препоръки на Microsoft ние ще изберем типа SDI интерфейс. Той е по прост и лесен както за използване, така и за програмна реализация.

Въпреки че се спряхме на реализацията на едно-документен интерфейс ние все пак ще трябва да вмъкнем и елементи от много-документния интерфейс, тъй като ще имаме няколко таблици с които ще трябва да работи потребителя и за негово удобство ще трябва да се показват една или най-много две в главния прозорец на приложението. За да не зареждаме всеки път различни данни в една таблица, което ще означава всеки път да се обръщаме към сървъра, ще използваме за реализацията една прекрасна възможност предоставена от средата на Delphi – така наречените фреймове (frame – рамка, но тъй като този термин се използва в различен смисъл в някой езици и области на информационните технологии, тук ще приемем, че представлява компонент подобен на прозорец в който могат да се поставят други компоненти). Това представлява клас, който може да се използва, като контейнер за други визуални компоненти. Той много наподобява прозорец, но от гледна точка на операционната система не е такъв. Фреймовете също така могат да бъдат записани като компоненти и после да се сложат в палитра с компоненти, което прави последващото им използване много лесно. Друга много важна и полезна характеристика на фреймовете е когато тяхна инстанция бъде сложена в друг прозорец (форма от гледна точка на терминологията в Delphi) или компонент контейнер, при промяна в базовия фрейм автоматично се обновяват всички инстанции или негови наследници. Но на нас за сега ще ни се наложи да използваме основно възможността им да съдържат в себе си други компоненти, чрез което образно казано можем да си направим наш компонент съставен от няколко други разположени както намерим за добре, които можем да разглеждаме като едно цяло.

Използвайки тази техника можем да сложим всяка таблица в отделен фрейм и евентуално други компоненти, като бутони и полета за редактиране, но в главния

прозорец да показваме само един фрейм. Останалите няма да са видими за потребителя и ще съдържат необходимите данни заредени и при необходимост ще визуализираме нужната таблица без да я зареждаме наново или да променяме динамично изгледа на главния прозорец. Този метод също е много полезен и лесен за реализация при смяна на необходимите бутони и други компоненти, реализиращи допълнителна функционалност за дадената таблица.

След като се спряхме на едно-документния интерфейс и уточнихме как ще го реализираме за нуждите на нашето приложение, трябва да обърнем внимание и на основните елементи, съставляващи потребителския интерфейс, част от които са следните:

- главно меню
- подменюта
- модални прозорци
- съобщения (за грешки, диалогови и др.)
- бутони за бърз достъп
- помощ

Главното меню ще бъде съставено от следните секции: “Файл” – за експорт и разпечатване на данни и отчети; “Данни” – за въвеждане на необходимите данни за работници, машини, обекти и заработените часове; “Изчисления” – за приключване на текущия месец и изчисляване на заплатите; “Справки” – за различни отчети; “Настройки” – за определяне на различни параметри на програмата; “Помощ” – информация за програмата и помощна информация, описваща функциите на приложението. Подменютата ще се базират на описаните по-горе ...

Друга особеност на интерфейса, която трябва да вземем предвид, идва от изискването на клиента да наподобява работата със Excel, тъй като до сега дейността за определяне на заплатите се вършела предимно със помощта на Excel. Поради тази причина ще използваме едновременно за визуализиране и въвеждане на данните таблици, които в Delphi се представят с компонента TGrid. За да се придържаме към стила на Excel няма да използваме допълнителни прозорци за въвеждане на данните или за редактирането им, това ще става директно в таблицата. Стандартният компонент TGrid в Delphi предоставя само основна функционалност, а на нас ще ни трябват някои екстри за улеснение на работата и за по-добро представяне на данните, като например при въвеждане на имена на работници в дадена клетка, това да не става ръчно, а клетката да става падащ списък с имената, от който да се избира необходимото само с

едно натискане на левия бутон на мишката. За това ще изберем друг компонент на друг производител, който предлага основната функционалност на TGrid плюс доста други екстри и ние ще се спрем на QuantumGrid на фирмата DeveloperExpress.

Тъй като данните няма да бъдат форматирани по някакъв специален начин и да се извършват различни преобразувания върху тях, ние няма да поддържаме възможността за връщане на дадено действие стъпка назад. Ако се налагат някакви корекции потребителят ще трябва да ги извършва ръчно, но това няма да го затрудни особено понеже няма да има действия, които се прилагат върху всички данни едновременно. Също така данните са разпределени в клетки и при необходимост за корекция може да се редактира само необходимата клетка, което не е голямо количество информация.

Друга допълнителна екстра, с която са свикнали потребителите и е представена в повечето програми са т. н. бутони за бърз достъп до менютата, които са разположени в лента под главното меню. Тъй като при нашето приложение командите на менюто не са голям брой, това не би затруднило потребителят и той би трябвало да се ориентира лесно в менюто без помощта на тези бутони. Поради тази причина за сега в приложението няма да се съдържат бутони за бърз достъп, но при настояване на клиента могат да бъдат прибавени много лесно към интерфейса на програмата благодарение на визуалната среда на Delphi и отличния начин за обработване на събития при компонентите, предоставени от Borland. В Delphi един обработчик на събитие може да бъде "закачен" на няколко компонента и то само с едно натискане на мишката.

Друг елемент, който ще използваме в интерфейса са т. н. модални прозорци, основна характеристика на които е, че се отварят над активния в момента прозорец и потребителят няма достъп до останалата част от приложението докато не затвори модалния прозорец. Такъв прозорец ще използваме за въвеждане на настройките на приложението и за въвеждане на някои други параметри като например сума за глобяване на всички работници едновременно.

Друга важна особеност на потребителския интерфейс е да представя на потребителя подходящи съобщения при възникнала грешка в работата на програмата или при друга необходимост, например да потвърди някое действие. В Delphi има различни методи за показване на съобщения, включително и за диалогови, където потребителят трябва да отговори на някакъв въпрос. Ние сме предвидили на

необходимите места в приложението да се показват съответни съобщения при възникнала грешка или необходимост от потвърждение от потребителя.

При необходимост от сведения за дадена функционалност на програмата е добре потребителят да разполага със съответната помощна информация, затова ние ще предвидим възможност за използването на помощ в приложението.

За улеснение на потребителя ще използваме в интерфейса за въвеждане на някои данни специализирани компоненти, като например където трябва да се въведе дата, това ще става с компонента `TDateTimePicker`, който наподобява календар и от него потребителят с едно или няколко натискания на мишката може да избере необходимата му дата.

3.3 Обектен модел на системата

Приложението е написано на обектно-ориентиран език и за това ще представим UML клас диаграма на проекта. Диаграмата е направена с `ModelMaker`, който се разпространява заедно с дистрибуцията на `Delphi 7`. На фиг. 2 е показан цялостен изглед на клас диаграмата на системата.

Както виждаме от диаграмата всички класове са наследници на класа `TObject`, в `Delphi` съществува конвенция за наследяването според която всички класове трябва да са наследници на този. Той съдържа методи за създаване и освобождаване на обект, който е инстанция на даден клас, както и други методи за управление на инстанциите на обектите.

Класът `TForm` представлява капсулация на прозорец на операционната система `Windows` и предоставя различни методи и свойства за управление на поведението на прозорец в приложението, които иначе трябва да се извикват чрез `API` функции на `Windows-a` (`Application Programmer Interface` – функции предоставени за използване на други програмисти от създателите на операционната система, чийто смисъл е в нашия контекст). Тъй като от гледна точка на `Windows-a` дори бутоните представляват прозорец, ние ще уточним, че тук става дума за прозорец на приложение. Също така предоставя обработчици на събития, които представляват методи, прихващащи изпращането на съобщения от `Windows-a` към дадения прозорец, където можем да сложим код, който да се изпълнява всеки път при настъпването на дадено събитие, например натискане на мишката върху прозореца. В приложението имаме четири класа, които го наследяват и ще ги опишем по-надолу.

Класът TFrame е подобен на TForm, но има някои разлики от него, и по-скоро се доближава до компонентите в Delphi, защото не може да бъде инстанциран като самостоятелен прозорец в приложението, а трябва да бъде поставен в друг прозорец. Основната идея на този клас е да служи като контейнер за други компоненти. Него го наследяват няколко класа, които съдържат различни таблици и компоненти за потребителския интерфейс.

Класът TDataModule представлява контейнер за компоненти, работещи с базата данни и в него не могат да се слагат други компоненти. Той е направен за улеснение на програмистите, за да могат да поставят управлението на базата данни в един модул. Ние сме го наследили в класа TdmCnt и в него сме поставили компонент за връзка с базата и други компоненти за работа с данните. Той съдържа методи, обработващи различни събития, настъпили с базата или със самите данни.

Класът TOptions съхранява настройките на програмата като има свойства, които съдържат техните стойности и методи, които да четат и записват тези стойности в регистрите на Windows-a.

Класът TLogger служи за записване на съобщения за грешки във файл, това е направено с цел по-лесно да се поддържа програмата. Има един метод, който извършва цялата работа.

Класът TfrmAboutBox представлява прозорец, даващ кратка информация за програмата и нейния автор. Той съдържа стандартни компоненти и няма допълнителни свойства и методи, които да добавят друга функционалност.

Класът TfrmOptions е прозорец, съдържащ интерфейс за въвеждане на настройките на приложението. Съдържа методи за обработка на потребителския вход и свойства с инстанции на класовете TLogger и TOptions. През свойството Logger всички останали модули на приложението могат да записват съобщения за грешки. Свойството Options също така служи, за да държи в паметта информация за настройките и при поискване да са достъпни за останалата част от приложението.

Класът TfrmPBSize представлява прозорец, който служи за въвеждане на глобите и поощренията за служителите. Има методи за обработване на потребителския вход и свойство за определяне на знака на стойността, т.е. от него зависи дали въведената сума е глоба или поощрение

Класът TfrmMain е главния прозорец на приложението, в който има методи предимно за показване на инстанции на фреймовете, които ще опишем по-надолу. Тези методи се изпълняват като обработчици на събития възникнали от потребителския

вход, т.е. при избор на различни команди от главното меню. Също така има методи за инициализиране на прозореца и на някои параметри на програмата при нейното стартиране. Тук се правят и някои изчисления, свързани с приключването на текущия месец и попълването на данни в базата за всички обекти, на които е работено през този месец.

Класът TfrWorkers представлява фрейм, в който се показват и редактират данните за служителите. Той няма методи, тъй като всичко се извършва директно през таблицата, показана във фрейма, което много наподобява работата с Excel, само че въведената информация директно се запазва в базата данни. По същия начин е организиран и класа TfrMachines и при него също не се налага писането на допълнителна функционалност, тъй като тя се предоставя от използваните компоненти.

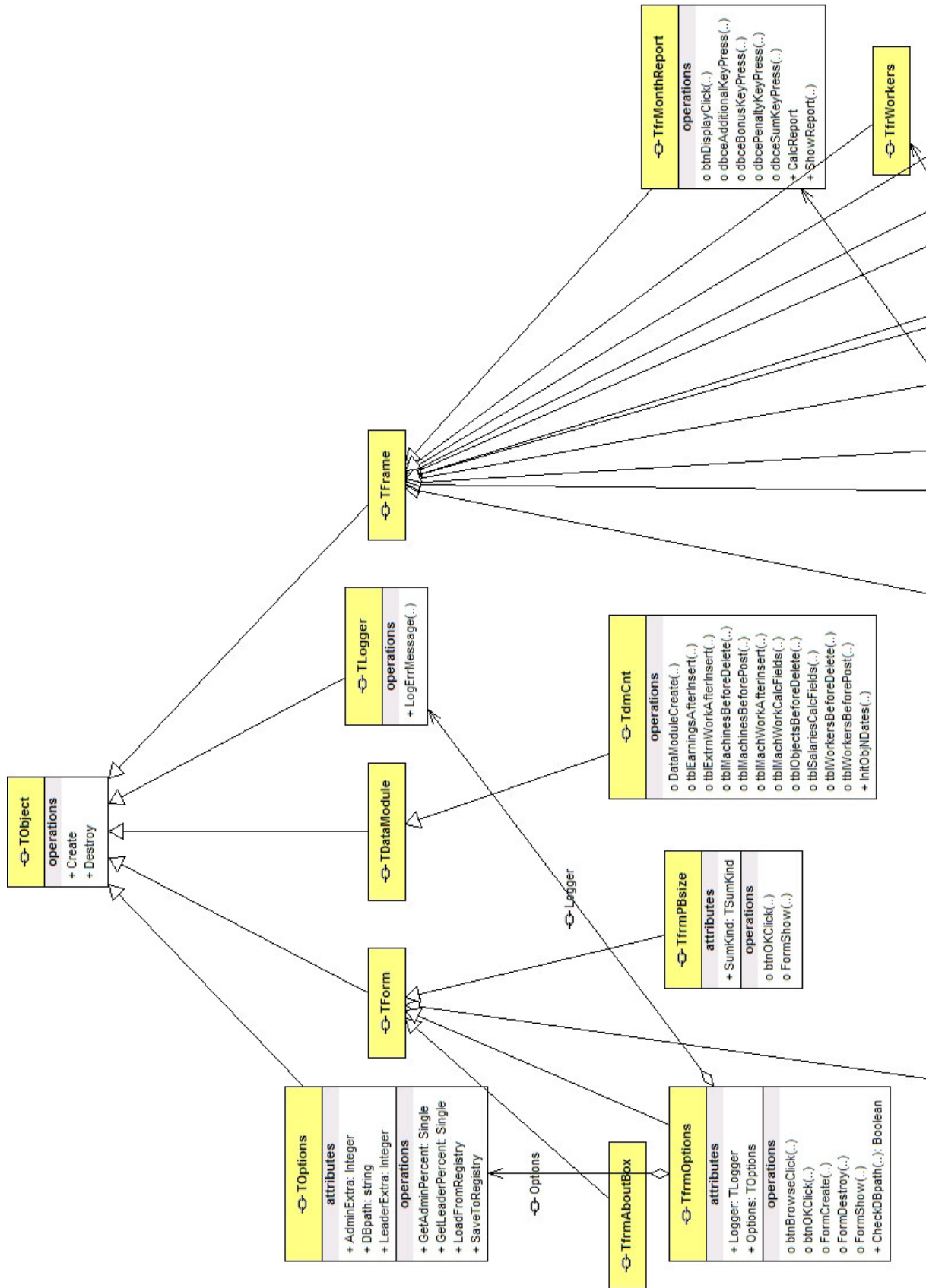
При класът TfrObjects е използван същия подход, но там имаме показани две таблици, които са свързани помежду си. Тук се налага да използваме метод за преизчисляване на някои данни за конкретния обект при смяна на коефициента на обекта за дадения месец, което и прави единствения метод в този клас.

Класът TfrEarnings представлява фрейм, в който се въвеждат заработените часове от служителите. Той съдържа метод за инициализация на показваните данни при първоначално извикване на инстанцията. Другите методи са за филтриране на записите по даден обект и съответната седмица, които представляват обработчици на събития на компоненти за определяне на дата и избор от даден списък, след което необходимата информация за часовете на служителите се въвежда директно в таблица.

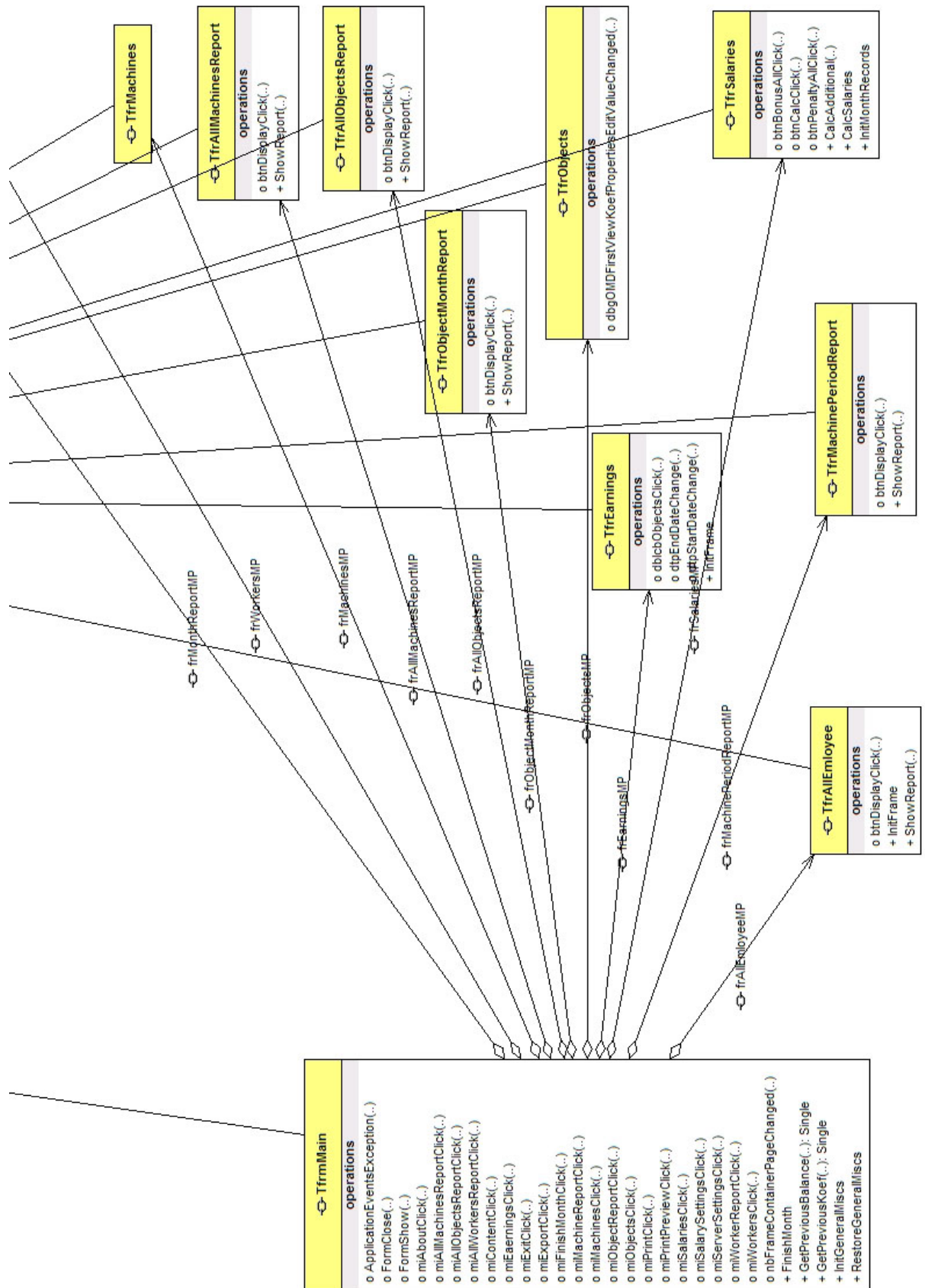
Класът TfrSalaries също представлява фрейм, на който основната функционалност се състои в изчисляване на заплатите на служителите за предходния месец. Има методи за първоначална инициализация на записите в базата данни за всички служители, за изчисляване на заплатите, за прибавяне на глоби и поощрения и за изчисляване на допълнителни надбавки, които се получават при предсрочно завършване на строителния обект.

Останалите шест класа са също фреймове, които показват различни отчети за служителите, обектите и машините. TfrMonthReport включва методи за получаване на отчет за заработките на конкретен служител за избран месец, TfrAllEmployee има методи за получаване на отчет за заплатите на всички служители за даден месец, TfrObjectsMonthReport съдържа методи за извличане на отчет за работата по конкретен обект за избран месец, TfrAllObjectsReport има методи за получаване на отчет за всички обекти за даден месец, TfrMachinePeriodReport включва методи за показване на отчет за

работата на конкретна машина през избран месец, TfrAllMachinesReport съдържа методи за извличането на отчет за всички машини за избран месец. При генериране на даден отчет получената информация не се записва в базата данни, а всеки път се извлича наново от съществуващата вече информация, което съкращава обема на пазената информация.



Фиг. 2 (1) UML клас диаграма на приложението



Фиг. 2 (2) UML клас диаграма на приложението

ГЛАВА 4. Описание на функциониращата система.

4.1 Инсталиране и начин на работа с продукта.

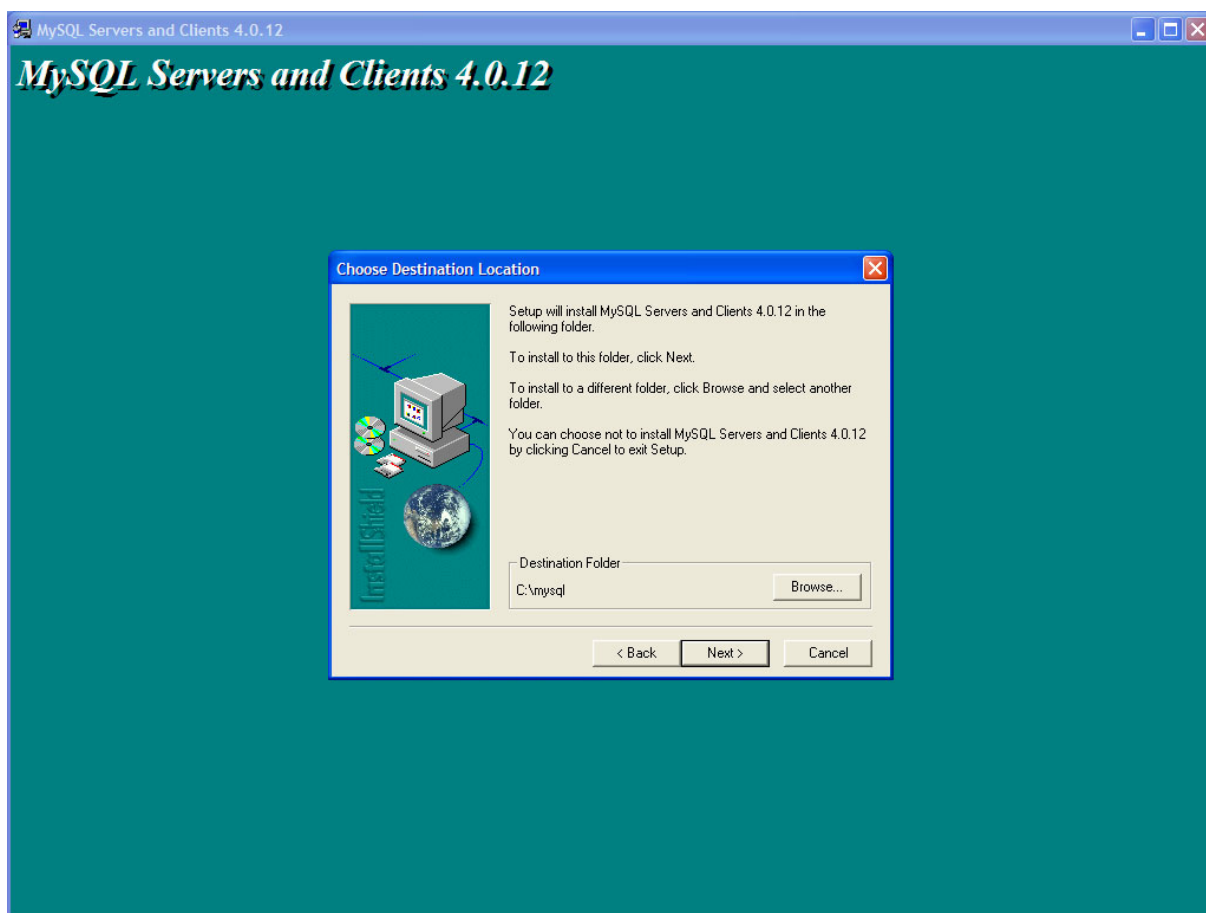
Приложението е разработено по поръчка на един клиент и най-вероятно ще работи на един или два компютъра, поради което не се налага честото му инсталиране. Поради тази причина не е направен инсталатор за програма, но има някои особености при инсталацията, които ние ще обясним накратко тук. Ако някога продуктът бъде предложен за употреба и от други фирми, тогава ще бъде разработена инсталираща програма или използвана готова така за по-лесно разпространени и улеснение на повечето клиенти.

Програмата използва за сървър за база данни MySQL и следователно, за да тръгне трябва първо да имаме инсталиран и стартиран такъв сървър, към който да се свържем. MySQL е безплатен сървър, но ако не се разпространява заедно с приложение, а самостоятелно. Затова трябва да го изтеглим предварително и да не го прибавяме към нашия продукт и също така да го инсталираме независимо и отделно от него.

Сървърът MySQL може да се изтегли от официалния сайт на разработчика му, който е <http://www.mysql.com/>. Като се отвори началната страница на сайта, след това трябва от секцията “Downloads” да се избере връзката “MySQL database”, която отваря нова страница с последна стабилна версия на сървъра, която се предлага за изтегляне. На тази страница излиза списък със всички операционни системи, на които може да се инсталира MySQL, а те са доста на брой, затова няма да ги изброяваме тук. След като се избере операционна система и вида инсталацията, трябва да се избере и сървър, от който да се изтегли инсталационният пакет. Да отбележим също така, че се предлага за изтегляне и изходния код на MySQL, който може да се компилира и после да се инсталира сървърът, но ние ще се спрем на готова компилирана инсталация, тъй като няма да правим никакви модификации на кода, а също така инсталацията ще е по-лесна. След като свалим това, което сме избрали, а в нашия случай това е инсталация за Windows, а файлът с пакета се казва mysql-4.1.14-win32.zip, трябва да го разархивираме и инсталираме[9].

Инсталацията съдържа файла Setup.exe, който трябва да изпълним след като сме разархивирали изтегления пакет. Преди това обаче в зависимост от версията на

Windows-а, на която ще инсталираме сървъра, трябва да се уверим, че имаме права да стартираме процес като “сървиз”. Следователно трябва да имаме администраторски парва за потребителя, на който се инсталира MySQL на повечето версии на Windows. Също така трябва да има инсталирана поддръжка на TCP/IP протокола. След като се уверим в горните две изисквания или направим, ако не са изпълнени, можем да стартираме файла Setup.exe, на фиг. 1 е показан изглед на инсталационния прозорец на MySQL.



Фиг. 1 Инсталация на MySQL сървър.

Като следваме инструкциите инсталираме сървъра в директорията, която сме избрали. Еwentуално след това можем да направим някои настройки на сървъра, да създадем потребители и права за достъп. Сървърът може да се управлява с команди от конзолата или чрез използването на различни менажери с графичен интерфейс като MySQLAdmin, който се разпространява стандартно с инсталацията или като

MySQLFront, който е на друг производител и трябва да се изтегли и инсталира допълнително, но пък той предлага повече възможности и функционалност.

След като имаме вече инсталиран и работещ сървър, другото което трябва да направим е да създадем базата данни и таблиците в нея. Това може да се направи автоматично при първото стартиране на програмата или ако има разработена инсталация, но поради същите причини поради които не направихме инсталираща програма не сме включили и при първо стартиране на програмата да се създава базата данни, за това трябва ръчно да я създадем. Това обаче е процес, който ще отнеме доста време за писане и ние ще се възползваме от друга възможност на сървъра, а именно да я създадем от скрипт съдържащ SQL конструкции за създаване на базата и таблиците в нея. Ние сме генерирали този скрипт предварително и сега ще трябва да го изпълним на сървъра, като единият начин е това да стане през конзолата, а другият през програмата, която използваме за управление на сървъра.

След като сме създали базата вече можем да преминем към инсталирането на програмата. То става като се копират файловете на приложението в избраната от нас директория, не са необходими други специални действия при самото инсталиране. След първото стартиране на програмата трябва да видим дали отчетите излизат на кирилица и ако това не е така, да рестартираме компютъра, тъй като приложението прави промяна в регистрите на Windows-а и при някои версии на операционната система промените не се отразяват автоматично. Като проверим това и евентуално при нужда рестартираме, вече програмата е готова за използване.

4.2 Преглед на основните възможности на продукта.

След като вече сме инсталирали сървъра за данни и сме създали структурата на базата данни, можем да стартираме програмата. Една от основните възможности на програмата е да поддържа данни за служителите на фирмата. За да ги въведеме или редактираме трябва да изберем от менюто “Данни” и после подменюто “Работници”, след което се появява таблица със служителите на фирмата (Фиг. 1). В полето “Име” се въвеждат трите имена съответния работник, в полето “Длъжност” има възможност от падащо меню да се избере каква позиция заема работникът, като може да се избира между “служител” и “ръководител”. В полето отдел също от падащо меню може да се избере в кой отдел работи служителя. В тази фирма отделите са следните: “Администрация”, “Електроизграждане”, “Строителство”, “Механизация” и

“Търговия”. В полето “Основна заплата” трябва да се попълни заплата на работника, под която не може да получава по-малко. Тъй като не всеки месец работникът може да бъде ангажиран 100%, а заплата се формира на база заработени часове и часова ставка за даден обект. Също така тази минимална заплата е индивидуална за всеки работник и зависи от времето, в което е работил във фирмата и по какъв начин си е вършил работата. В полето “Заплата на час” се съдържа средната работна часова ставка на дадения работник за цялото време, в което е бил във фирмата. Тя се опреснява всеки месец спрямо заработеното от него. Тя служи да се формира заплащането му за обектите, които са без договор и съответно няма предварително определени часови ставки. Полето “Месеци работа” се използва, за да съхранява информация колко месеца даденият работни е във фирмата. То участва в изчисляването на средната ставка за час.

Редактирането на данните става директно като върху старите се нанесат нови, след което трябва да се мине на друг ред, за да се запишат корекциите в базата. Добавянето на нов служител става като се отиде на последния ред в таблицата и след това със стрелка надолу се създава нов празен ред, който се попълва директно. Изтриването на служител става с комбинация от клавишите “CTRL” + “DEL”, след което излиза диалог за потвърждение на операцията и ако бъде положително се изтрива фиктивно от таблицата, но във базата остава. Там само се променя едно поле, което указва, че запис е изтрит. Това е направено с цел при правене на отчети за по-стари дати да може да участва и този служител.

Име	Длъжност	Отдел	Основна заплата	Заплата на час	Месеци работа
Евгени Христов Георгиев	Служител	Строителство	220.00 лв	0.93 лв	11
Недялко Димитров Тодоров	Служител	Строителство	220.00 лв	1.41 лв	12
Йонко Илиев Петров	Служител	Механизация	220.00 лв	1.75 лв	18
Ангел Цонев Димитров	Служител	Строителство	220.00 лв	1.08 лв	6
Христо Николов Кръстев	Служител	Строителство	220.00 лв	1.34 лв	8
Негайгил Салиев Палата	Служител	Строителство	220.00 лв	1.17 лв	6
Асен Атанасов Филипов	Служител	Строителство	220.00 лв	1.57 лв	11
Борислав Тодоров Матеев	Служител	Строителство	220.00 лв	1.74 лв	10
Кръстьо Иванов Кръстев	Служител	Строителство	220.00 лв	1.71 лв	20
Румен Николов Енев	Служител	Строителство	220.00 лв	1.68 лв	23
Иван Ганчев Иванов	Служител	Строителство	220.00 лв	2.05 лв	14
Йордан Георгиев Денков	Служител	Строителство	220.00 лв	1.40 лв	16
Иван Гатев Иванов	Служител	Електроизграждане	220.00 лв	1.89 лв	24
Стефан Йорданов Стефанов	Служител	Електроизграждане	220.00 лв	1.59 лв	19
Марин Стоянов Геше	Служител	Електроизграждане	220.00 лв	1.21 лв	9
Димитър Илиев Димитров	Служител	Електроизграждане	220.00 лв	1.44 лв	9
Симеон Стефанов Стефанов	Служител	Електроизграждане	220.00 лв	1.69 лв	12
Николай Добрев Минчев	Служител	Строителство	220.00 лв	1.37 лв	10
Христо Христов	Ръководител	Строителство	300.00 лв	2.46 лв	33
Ангел Димитров Чолаков	Служител	Строителство	250.00 лв	1.93 лв	23
Веселин Йорданов Иванов	Служител	Строителство	250.00 лв	1.62 лв	11
Костадин Милков Гърдев	Служител	Строителство	250.00 лв	1.79 лв	43
Юри Русев Найденов	Служител	Механизация	250.00 лв	2.04 лв	38
Марин Стоянов Маринов	Служител	Строителство	250.00 лв	1.54 лв	11
Юри Стефанов Савов	Служител	Механизация	250.00 лв	1.99 лв	18
Асен Георгиев Русев	Служител	Механизация	250.00 лв	1.14 лв	8
Румен Георгиев Русев	Служител	Строителство	250.00 лв	1.78 лв	22
Михаил Игнатов Ситиров	Служител	Електроизграждане	250.00 лв	1.50 лв	17

Фиг. 1 Въвеждане и редактиране на данни за служителите.

Друга основна функционалност на програмата е поддържане и редактиране на данни за строителните обекти, на които работят бригади на фирмата. Достъпът до тези данни става като от менюто се избере “Данни” и после подменю “Обекти”. Показаната таблица на Фиг. 2 съдържа данни за строителните обекти. Полето “Обект” съдържа името или кодовото название на строителния обект. В полето “Начална дата” се въвежда дата на стартиране на строителните работи на обекта, а в полето “Крайна дата” се въвежда дата, на която трябва да приключи изпълнението на обекта. Полето “Общо часове” съдържа броя на договорените с инвеститора на обекта човекочаса, за които трябва да се изпълни работата по обекта. Полето “Цена на час” съдържа определената цена за един човекочас работа. Ако по обекта се работи без предварителна уговорка за цена и брой часове тези полета са нули и след приключване на обекта инвеститорът заплаща обща сума за цялата работа, която се определя впоследствие от фирмата, а не предварително с договор. В полето “Механизация” се попълва сумата, за която ще се използва механична и транспортна техника на обекта. Полето “Статус” отразява дали в

момента се работи по обекта или той вече е приключен, като са възможни две стойности за това поле – “Активен” и “Приключен”, като те се избират от падащо меню. Последното поле в тази таблица “Вид обект” отразява дали обекта се изпълнява по предварително сключен договор или заплащането ще се определя след приключване на работата. Тука също са възможни две стойности: “Договор” и “Без договор”, като те се избират от падащо меню.

Редакцията на данни става директно върху старите. Добавянето на нов обект става като се отиде на последния ред и се натисне стрелка надолу.

При селектиране на даден обект в таблицата под обектите се показват допълнителни данни за избрания обект (Фиг. 2). В полето “Месец” се отразява за кой месец са допълнителните данни. Полето “Коефициент” служи за определяне на часовите ставки за дадения обект като се определя от управителя в зависимост от това как върви изпълнението на обекта за текущия месец. В полето “Извършена работа” се записва каква сума е изплатена за заплати за дадения месец за този обект. Полето “Използвана механизация” съдържа сумата, която е отишла за ползването на машини и транспорт на обекта. Полето “Външна работа” съдържа сумата която, трябва да се изплати на външни работници, т.е. такива които не са служители на фирмата или на други фирми, които са подизпълнители на обекта. Последното поле “Обща сума” съдържа сбора от предходните три полета като отразява общата сума, дадена за работа през конкретния месец за този обект.

Данните в таблицата се попълват автоматично от програмата като единствено може да се променя коефициентът на обекта. След което данните отново се преизчисляват от програмата. По този начин, ако се забавя изпълнението на даден обект може да се влияе на часовите ставки, които се дават на работниците, извършващи дейности на този обект. Така фирмата може да ограничи или да избегне загуба от заплати, поради проточване и забавяне на строителните дейности на даден обект. В същото време, ако има преизпълнение на работата и обектът се приключи предсрочно, останалата сума предвидена за заплати в договора се преразпределя между работниците, които са били на този обект.

Обекти							
Обект	Начална дата	Крайна дата	Общо часове	Цена за час	Механизация	Статус	Вид обект
ОРУ 110 кV Подстанция Горна Оряховица	03.10.2005 г.	31.1.2006 г.	10000	1.80 лв	18 000.00 лв	Активен	Договор
TVS Горна Оряховица	01.9.2005 г.	18.11.2005 г.	7500	1.70 лв	9 500.00 лв	Активен	Договор
Кабел Ср 4 ТП 71-72 Велико Търново	10.10.2005 г.	20.2.2006 г.	0	0.00 лв	0.00 лв		Договор
Газификация	01.6.2005 г.	31.3.2006 г.	0	0.00 лв	0.00 лв	Активен	Без договор

Данни за избрания обект					
Месец	Коефициент	Извършена работа	Използвана механизация	Външна работа	Оставаща сума
01.9.2005 г.	1.00	3 903.30 лв	2 741.70 лв	320.00 лв	29 035.00 лв

Фиг. 2 Въвеждане и редактиране на данни за обектите и допълнителни данни за избрания обект.

Програмата предоставя възможност да се обработват и съхраняват данни за използваните машини за извършване на строителните дейности. От менюто като се избере “Данни” и после подменю “Механизация” се появява таблица с данни за машините (Фиг. 3). В полето “Машина” се въвежда името на машината. Полето “Цена на км” съдържа цената за километър придвижване на дадената машина или ако е транспортно средство (камион, самосвал) цената за превоз на товар за километър. В полето “Цена за час” се въвежда съответната цена за основния вид дейност, която извършва машината. В полето “Цена за час друга дейност” се попълва цена за час, ако машината може да извършва и друга дейност освен основната си.

Редактирането на въведените данни става директно върху вече въведените. Нова машина се въвежда като се мине на последния ред и се натисне стрелка надолу. Изтриването се извършва със комбинация от клавиши “CTRL” + “DEL”.

Механизация / транспорт			
Машина	Цена на км	Цена за час работа	Цена за час други дейности
Автовъзка	2.50 лв	6.00 лв	0.00 лв
Багер Геомакс	4.00 лв	15.00 лв	0.00 лв
Хидравлична преса	3.50 лв	20.00 лв	0.00 лв
ЗИЛ триос	4.00 лв	0.00 лв	0.00 лв
STHLL за рязане на бетон	5.00 лв	12.00 лв	0.00 лв
BT 43-46	3.00 лв	0.00 лв	0.00 лв
Багер РСАТ BT 22-55	4.00 лв	18.00 лв	0.00 лв
BT 78-21	1.80 лв	0.00 лв	0.00 лв
BT 44-86	3.00 лв	0.00 лв	0.00 лв
BT 12-82	2.50 лв	0.00 лв	0.00 лв
BT 78-17	3.00 лв	0.00 лв	0.00 лв

Фиг. 3 Въвеждане и редактиране на данни за наличните машини и транспорт.

За да могат да се изчисляват заплатите, трябва да се въвеждат всяка седмица отчети с данни за всеки работник: колко часа и на кой обект е работил. Също така отчетите съдържат данни за използваната механизация и използваните външни работници. Въвеждането на тези данни става като от менюто се избере “Данни” и после подменю “Заработки”. Първо трябва да се избере обект и след това началната и крайната дата на седмицата, за която ще се въвеждат данни от отчетите. Както се вижда на фигура 4 има три таблици за съответните заплатки. Ако няма вече въведени редове въвеждането на нови данни става като се кликне върху таблицата, за да стане активната контрола и след това се натисне стрелка надолу. При вече въведени редове за нов ред трябва да се отиде на последния ред и да се натисне стрелка надолу. Първата таблица е за данните за работниците на фирмата. В полето “работник” се попълва името на работника. Колко часове е работил съответния работник на избрания обект се въвежда в полето “Часове”. Ако работника има повече от осем часа, останалите над тази цифра трябва да се въведат в полето “Извънредни часове”. Също така ако е работено по време на национални празници или събота и неделя тези часове се въвеждат в полето

“Празнични часове”. В следващата таблица се попълват данни за използваните машини на обекта. Първото поле “Машина” служи за въвеждане на името на машината. В следващото поле “Изминати километри” се попълва колко километра е изминала машината до обекта или ако е някакъв вид транспорт, съответно колко километра е извършвал транспортни услуги. Полето “Часове работа” служи за въвеждане на броя часове, за които е използвана машината. Понеже някои машини са с възможности за двойна употреба, съответно могат да извършват и друга дейност освен основната, за която са предназначени е предвидено и поле “Часове друга работа”, в което се попълват часовете, за които машината е работила по другата си дейност. В третата таблица се попълват данни за извършени дейности по обекта от външни работници, т.е. такива, които не са наети във фирмата или от други фирми подизпълнители. В полето “Извършена работа” се въвежда сумата, за която е работил външният работник или друга фирма. А полето “Коментар” служи за въвеждане на подсещаща бележка относно въведената сума.

Както и при другите таблици и в тези данните могат да се редактират директно, а изтриването става с комбинацията “CTRL” + “DEL”.

Труд и работна заплата

Файл Данни Изчисления Справки Настройки Помощ

Заработки

Обект: ОРУ 110 кв Подстанции От дата: 01. 9 .2005 г. До дата: 04. 9 .2005 г.

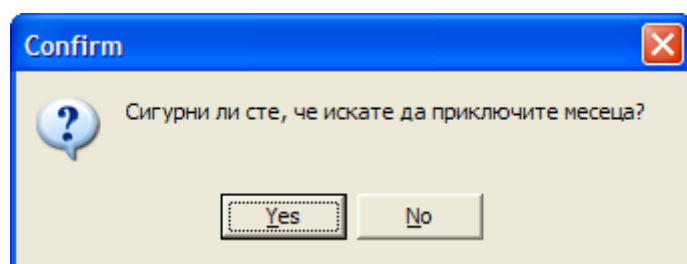
Работник	Часове	Извънредни часове	Празнични часове
Евгени Христов Георгиев	38	0	0
Недялко Димитров Тодоров	38	0	0
Йонко Илиев Петров	38	0	0
Ангел Цонев Димитров	28	0	0
Христо Николов Кръстев	38	0	0
Негайгил Салиев Палата	38	0	0
Асен Атанасов Филипов	38	0	0
Борислав Тодоров Матеев	38	0	0

Машина	Километри пробег	Часове работа	Часове друга работа
Автовъшка	0	14	0
Багер Геомакс	0	26	0
Хидравлична преса	0	8	0
ЗИЛ триос	125	0	0
STHLL за рязане на бетон	0	2	0
BT 43-46	52	0	0

Извършена работа	Коментар
100	Граждански договор - Румен Асенов Русев
120	Граждански договор - Михаил Сотиров Маринов
100	Граждански договор - Юрий Стоянов Савов

Фиг. 4 Въвеждане и редактиране на данни за отработените часове на работници и машини за даден обект в определена седмица.

След като бъдат въведени данните за всички седмици на даден месец може да се пристъпи към изчисляване на заплатите на работниците за този месец. За целта първо от менюто трябва да се избере “Изчисления” и после подменюто “Приключване на месец” (Фиг. 5). Това стартира изчисления за сумите, които са изразходвани за даден обект за този месец. След което данните се попълват във втората таблица показана на Фиг. 2 “Данни за избрания обект”. Това се прави с цел да може да се определи какъв коефициент да се сложи на дадения обект за съответния месец. При първоначалното изчисление се взема коефициента за предходния месец. Ако това е първи месец за обекта се взема коефициент единица. Ако коефициентът бъде сменен съответно наново се преизчисляват сумите в тази таблица. След като бъдат направени тези изчисления може да се пристъпи към определяне на заплатите.



Фиг. 5 Стартиране на изчисления за сумите на обектите след края на месеца.

След приключването на изчисленията за предходния месец и евентуална промяна на коефициентите на някои обекти, вече може да се пристъпи към изчисляване на заплатите на работниците. Това става като от менюто “Изчисления” се избере подменю “Заплати” (Фиг. 6). За да се стартира определянето на заплатите се натиска бутон “Изчисли”, след което в таблицата се появяват съответните стойности. Полето “Служител” съдържа името на съответния работник. В полето “Месец” е записан съответният месец, за който е изчислена заплата. Полето “Средна часова ставка” съдържа стойността на средната ставка за час, понеже работникът може да е работил на няколко обекта през месеца и за всеки от тях да е определена различна такава. От тази средна ставка се актуализират всеки месец данните в таблицата за работниците в полето “Заплата на час”, която се взема при изчисления на заработеното от работника

при работа на обекти, които са без договор. Полето “Заплата” съдържа изчислената заплата за месеца за всички обекти, на които е работил съответния работник. Полето “Глоби” съдържа сума, с която евентуално работникът може да бъде санкциониран за различни нарушения. Полето “Поощрения” съдържа сума, с която може да бъде награден работникът по преценка на ръководителя. Полето “Допълнения” съдържа сума, която се получава като се приключи обектът и средствата предвидени за труд не са изразходвани, така останалата сума се разпределя между работниците, които са участвали само за този обект. Ако работникът е бил на няколко такива обекта всички полагащи му се допълнения от преразпределението се сумират в това поле. Полето “Коефициент” служи за увеличаване или намаляване на заплата на съответния работник с фиксиран процент, в зависимост от неговата производителност на труда. След като се вземат предвид стойностите от полетата “Глоби”, “Поощрения”, “Допълнения” и “Коефициент” и се приложат съответните аритметични действия към стойността, съдържаща се в полето “Заплата” се получава окончателната заплата, която се показва в полето “Крайна заплата”. Таблицата съдържа още едно поле “Коментар”, ако има нужда да се сложи някаква бележка относно формирането на заплатата.

Бутон “Глоби всички” служи да се наложи една глоба на всички работници, като се натисне той, се появява диалогов прозорец, в който се попълва стойността на глобата. Аналогичен е и бутонът “Поощри всички”, за да се даде премия на всички работници.

Служител	Месец	Средна часова ставка	Заплата	Глоби	Пощрения	Допълнения	Коефициент	Крайна заплата	Коментар
Евгени Христов Георгиев	01.9.2005 г.	1.80 лв	190.80 лв	0.00 лв	0.00 лв	0.00 лв	0%	190.80 лв	
Недялко Димитров Тодоров	01.9.2005 г.	1.80 лв	287.10 лв	0.00 лв	0.00 лв	0.00 лв	0%	287.10 лв	
Йонко Илиев Петров	01.9.2005 г.	1.80 лв	272.70 лв	0.00 лв	0.00 лв	0.00 лв	0%	272.70 лв	
Ангел Цонев Димитров	01.9.2005 г.	1.80 лв	174.60 лв	0.00 лв	0.00 лв	0.00 лв	0%	174.60 лв	
Христо Николов Кръстев	01.9.2005 г.	1.80 лв	287.10 лв	0.00 лв	0.00 лв	0.00 лв	0%	287.10 лв	
Негайгил Салиев Палата	01.9.2005 г.	1.80 лв	287.10 лв	0.00 лв	0.00 лв	0.00 лв	0%	287.10 лв	
Асен Атанасов Филипов	01.9.2005 г.	1.80 лв	192.60 лв	0.00 лв	0.00 лв	0.00 лв	0%	192.60 лв	
Борислав Тодоров Матеев	01.9.2005 г.	1.80 лв	254.70 лв	0.00 лв	0.00 лв	0.00 лв	0%	254.70 лв	
Кръстьо Иванов Кръстев	01.9.2005 г.	1.80 лв	201.60 лв	0.00 лв	0.00 лв	0.00 лв	0%	201.60 лв	
Румен Николов Енев	01.9.2005 г.	1.80 лв	164.70 лв	0.00 лв	0.00 лв	0.00 лв	0%	164.70 лв	
Иван Ганчев Иванов	01.9.2005 г.	1.80 лв	189.00 лв	0.00 лв	0.00 лв	0.00 лв	0%	189.00 лв	
Йордан Георгиев Денков	01.9.2005 г.	1.80 лв	292.50 лв	0.00 лв	0.00 лв	0.00 лв	0%	292.50 лв	
Иван Гатев Иванов	01.9.2005 г.	1.80 лв	68.40 лв	0.00 лв	0.00 лв	0.00 лв	0%	68.40 лв	
Стефан Йорданов Стефанов	01.9.2005 г.	1.80 лв	225.00 лв	0.00 лв	0.00 лв	0.00 лв	0%	225.00 лв	
Марин Стоянов Гешева	01.9.2005 г.	1.80 лв	122.40 лв	0.00 лв	0.00 лв	0.00 лв	0%	122.40 лв	
Димитър Илиев Димитров	01.9.2005 г.	1.80 лв	182.70 лв	0.00 лв	0.00 лв	0.00 лв	0%	182.70 лв	
Симеон Стефанов Стефанов	01.9.2005 г.	1.80 лв	84.60 лв	0.00 лв	0.00 лв	0.00 лв	0%	84.60 лв	
Николай Добрев Минчев	01.9.2005 г.	1.80 лв	287.10 лв	0.00 лв	0.00 лв	0.00 лв	0%	287.10 лв	
Христо Христов	01.9.2005 г.	1.80 лв	138.60 лв	0.00 лв	0.00 лв	0.00 лв	0%	138.60 лв	
Ангел Димитров Чолаков	01.9.2005 г.	1.70 лв	146.20 лв	0.00 лв	0.00 лв	0.00 лв	0%	146.20 лв	
Веселин Йорданов Иванов	01.9.2005 г.	1.70 лв	406.30 лв	0.00 лв	0.00 лв	0.00 лв	0%	406.30 лв	
Костадин Милков Гърдев	01.9.2005 г.	1.70 лв	176.80 лв	0.00 лв	0.00 лв	0.00 лв	0%	176.80 лв	
Юри Русев Найденов	01.9.2005 г.	1.70 лв	232.90 лв	0.00 лв	0.00 лв	0.00 лв	0%	232.90 лв	
Марин Стоянов Маринов	01.9.2005 г.	1.70 лв	299.20 лв	0.00 лв	0.00 лв	0.00 лв	0%	299.20 лв	
Юли Стефанов Савов	01.9.2005 г.	1.70 лв	289.00 лв	0.00 лв	0.00 лв	0.00 лв	0%	289.00 лв	

Фиг. 6 Определяне на заплатите на работниците.

След като са извършени изчисленията за месеца и заплатите на работниците могат да се видят различни справки за сумите. Първата е “Отчет за служител за месец”, тя се показва като от менюто “Справки” се избере подменю “Служител за период”. Тя показва таблица с информация за всеки един служител поотделно къде и колко е работил (Фиг. 7). В полето “Обект” се съдържа името на обекта, на който е бил служителят, ако е бил на няколко обекта през месеца, за всеки обект се появява поотделно информация. Полетата “От дата” и “До дата” съответно съдържат информация от кога до кога е работил на дадения обект избраният работник. Полето “Часове работа” показва колко часа е заработил работникът на този обект в съответния период. В полето “Заработка” се съдържа сумата, която му се полага за съответния период. Под таблицата са разположени четири полета с допълнителна информация. Първото отразява дали работникът има получени глоби, второто показва дали има дадени премии, третото съдържа стойността на получени добавки, ако има такива и четвъртото е за крайната сума, която е получил работникът като заплата.

Данните в отчета не могат да се редактират, но могат да се разпечатат. Това става като от менюто “Файл” се избере подменюто “Печат”, което директно ще разпечата отчета. Също така има възможност да се разгледа предварително страницата, която ще се печата, което става от менюто “Файл” и после се избира подменю “Преглед”. Разпечатката от този отчет може да се използва, за да се дава детайлна информация на работниците за получената заплата.

Труд и работна заплата

Файл Данни Изчисления Справки Настройки Помощ

Отчет за служител за месец

Месец: 16. 9 .2005 г. Служител: Евгени Христов Георги

Обект	От дата	До дата	Часове работа	Заработка
ОРУ 110 kv Подстанция Горна Оряховица	01.9.2005 г.	04.9.2005 г.	38	68.40 лв
ОРУ 110 kv Подстанция Горна Оряховица	12.9.2005 г.	18.9.2005 г.	68	122.40 лв

Глоби: Поощрения: Допълнения: Общо:

Фиг. 7 Отчет за даден работник за месеца - какво е заработил.

Друг отчет, който може да се види е за всички служители общо. Той се показва като от менюто “Справки” се избере подменю “Общо служители” (Фиг. 8). След това трябва да се избере месец, за който да се изведат данните и да се натисне бутоната “Покажи”. Тогава в таблицата се появяват данните за избрания месец. Полето “Име” съдържа името на работника. Полето “Месец” е за избрания месец. В полето “Средна ставка” се съдържа стойността на средната ставка, за която е работил служителя през дадения месец. Полето “Заработка” съдържа заработка на работника без начислени допълнителни суми. Полето “Глоби” показва дали има поставени глоби и каква им е

сумата. “Поощрения” – дава стойността на премиите на работника. Полето “Добавки” съдържа сумата на допълнителните пари, преразпредели от обектите, ако има такива. Полето “Заплата” съдържа крайната сума, която трябва да получи работникът за дадения месец. В полето “Коментар” може да има някакви бележки относно заплатата на работника. Този отчет също може да бъде разпечатан и да се види страницата предварително, което става от менюто “Файл” и съответно “Печат” и “Преглед”.

Име	Месец	Средна ставка	Заработка	Глоби	Поощрения	Добавки	Заплата	Коментар
Евгени Христов Георгиев	01.9.2005 г.	1.80 лв	190.80 лв	0.00 лв	0.00 лв	0.00 лв	190.80 лв	
Недялко Димитров Тодоров	01.9.2005 г.	1.80 лв	287.10 лв	0.00 лв	0.00 лв	0.00 лв	287.10 лв	
Йонко Илиев Петров	01.9.2005 г.	1.80 лв	272.70 лв	0.00 лв	0.00 лв	0.00 лв	272.70 лв	
Ангел Цонев Димитров	01.9.2005 г.	1.80 лв	174.60 лв	0.00 лв	0.00 лв	0.00 лв	174.60 лв	
Христо Николов Кръстев	01.9.2005 г.	1.80 лв	287.10 лв	0.00 лв	0.00 лв	0.00 лв	287.10 лв	
Негайгил Салиев Палата	01.9.2005 г.	1.80 лв	287.10 лв	0.00 лв	0.00 лв	0.00 лв	287.10 лв	
Асен Атанасов Филипов	01.9.2005 г.	1.80 лв	192.60 лв	0.00 лв	0.00 лв	0.00 лв	192.60 лв	
Борислав Тодоров Матеев	01.9.2005 г.	1.80 лв	254.70 лв	0.00 лв	0.00 лв	0.00 лв	254.70 лв	
Кръстьо Иванов Кръстев	01.9.2005 г.	1.80 лв	201.60 лв	0.00 лв	0.00 лв	0.00 лв	201.60 лв	
Румен Николов Енев	01.9.2005 г.	1.80 лв	164.70 лв	0.00 лв	0.00 лв	0.00 лв	164.70 лв	
Иван Ганчев Иванов	01.9.2005 г.	1.80 лв	189.00 лв	0.00 лв	0.00 лв	0.00 лв	189.00 лв	
Йордан Георгиев Денков	01.9.2005 г.	1.80 лв	292.50 лв	0.00 лв	0.00 лв	0.00 лв	292.50 лв	
Иван Гатев Иванов	01.9.2005 г.	1.80 лв	68.40 лв	0.00 лв	0.00 лв	0.00 лв	68.40 лв	
Стефан Йорданов Стефанов	01.9.2005 г.	1.80 лв	225.00 лв	0.00 лв	0.00 лв	0.00 лв	225.00 лв	
Марин Стоянов Гешева	01.9.2005 г.	1.80 лв	122.40 лв	0.00 лв	0.00 лв	0.00 лв	122.40 лв	
Димитър Илиев Димитров	01.9.2005 г.	1.80 лв	182.70 лв	0.00 лв	0.00 лв	0.00 лв	182.70 лв	
Симеон Стефанов Стефанов	01.9.2005 г.	1.80 лв	84.60 лв	0.00 лв	0.00 лв	0.00 лв	84.60 лв	
Николай Добрев Минчев	01.9.2005 г.	1.80 лв	287.10 лв	0.00 лв	0.00 лв	0.00 лв	287.10 лв	
Христо Христов	01.9.2005 г.	1.80 лв	138.60 лв	0.00 лв	0.00 лв	0.00 лв	138.60 лв	
Ангел Димитров Чолаков	01.9.2005 г.	1.70 лв	146.20 лв	0.00 лв	0.00 лв	0.00 лв	146.20 лв	
Веселин Йорданов Иванов	01.9.2005 г.	1.70 лв	406.30 лв	0.00 лв	0.00 лв	0.00 лв	406.30 лв	
Костадин Милков Гърдев	01.9.2005 г.	1.70 лв	176.80 лв	0.00 лв	0.00 лв	0.00 лв	176.80 лв	
Юри Русев Найденов	01.9.2005 г.	1.70 лв	232.90 лв	0.00 лв	0.00 лв	0.00 лв	232.90 лв	
Марин Стоянов Маринов	01.9.2005 г.	1.70 лв	299.20 лв	0.00 лв	0.00 лв	0.00 лв	299.20 лв	
Юри Стефанов Савов	01.9.2005 г.	1.70 лв	289.00 лв	0.00 лв	0.00 лв	0.00 лв	289.00 лв	

Фиг. 8 Общ отчет за заплатите на всички служители.

Следващият отчет, който може да се покаже, е за всеки обект поотделно, кои работници са работили на него за даден месец, каква механизация е използвана и какви външни услуги са използвани. Това става като от менюто “Справки” се извика подменюто “Обект за период” (Фиг. 9). За да се извади отчета трябва да се избере обект и после месец, за който да е отчета, след което се натиска бутона “Покажи”. Показват се три таблици, които съдържат данни съответно за работниците, машините и външната работа. Те са аналогични с таблиците, в които се попълват заработките, като са добавени във всяка по две полета: “От дата” и “До дата”, в които се показва периодът,

за който е дадения запис, тъй като отчетът е за месец, а записите са за седмица или по-малко. Също така този отчет може да бъде разпечатан и да се види страницата предварително, което става от менюто “Файл” и съответно “Печат” и “Преглед”.

Труд и работна заплата

Файл Данни Изчисления Справки Настройки Помощ

Отчет за обект за месец

Обект: ОРУ 110 кв Подстанции Месец: 07. 9. 2005 г. Покажи

Работник	От дата	До дата	Часове	Извънредни часове	Празнични часове
Недялко Димитров Тодоров	05.9.2005 г.	11.9.2005 г.	40	9	0
Йонко Илиев Петров	05.9.2005 г.	11.9.2005 г.	40	9	0
Ангел Цонев Димитров	05.9.2005 г.	11.9.2005 г.	29	0	0
Христо Николов Кръстев	05.9.2005 г.	11.9.2005 г.	40	9	0
Негайгил Салиев Палата	05.9.2005 г.	11.9.2005 г.	40	9	0
Асен Атанасов Филипов	05.9.2005 г.	11.9.2005 г.	29	0	0
Борислав Тодоров Матеев	05.9.2005 г.	11.9.2005 г.	40	9	0
Кръстьо Иванов Кръстев	05.9.2005 г.	11.9.2005 г.	29	0	0
Румен Николов Енев	05.9.2005 г.	11.9.2005 г.	40	9	0
Иван Ганчев Иванов	05.9.2005 г.	11.9.2005 г.	19	0	0
Йордан Георгиев Денков	05.9.2005 г.	11.9.2005 г.	30	0	0
Стефан Йорданов Стефанов	05.9.2005 г.	11.9.2005 г.	19	0	0

Машина	От дата	До дата	Километри	Часове работа	Часове друга работа
Автовилка	01.9.2005 г.	04.9.2005 г.	0	14	0
Багер Геомакс	01.9.2005 г.	04.9.2005 г.	0	26	0
Хидравлична преса	01.9.2005 г.	04.9.2005 г.	0	8	0
ЗИЛ триос	01.9.2005 г.	04.9.2005 г.	125	0	0
SHLL за рязане на бетон	01.9.2005 г.	04.9.2005 г.	0	2	0

От дата	До дата	Сума	Коментар
01.9.2005 г.	04.9.2005 г.	100	Граждански договор - Румен Асенов Русев
01.9.2005 г.	04.9.2005 г.	120	Граждански договор - Михаил Сотиров Маринов
01.9.2005 г.	04.9.2005 г.	100	Граждански договор - Юрий Стоянов Савов

Фиг. 9 Отчет за обект за даден месец.

Друг вид отчет, който може да се покаже е за всички обекти общо за даден месец. Това става като от менюто “Справки” се избере подменю “Общо обекти” (Фиг. 10). За да се покажат данните трябва да се избере месец и после да се натисне бутоната “Покажи”. В дадената таблица се появяват данни за всички обекти, на които е работено за избрания месец. Полето “Обект” съдържа името на обекта. Полето “Коефициент” съдържа стойността на коефициента за дадения обект. В полето “Изразходвана сума” се показва стойността, която е отишла за заплащане на работниците, работили на този обект за този месец. В полето “Механизация” се съдържа сумата, която е използвана за машините и транспорта осигурявали услуги на обекта. В полето “Външна работа” се отразява сумата, изразходвана за заплащане на външни работници и фирми

подизпълнители. Полето “Оставаща сума” показва колко пари остават неизразходвани за труд от предварително договорените за дадения обект.

Подобно на другите отчети така и този отчет може да бъде разпечатан и да се види страницата предварително, което става от менюто “Файл” и съответно “Печат” и “Преглед”.

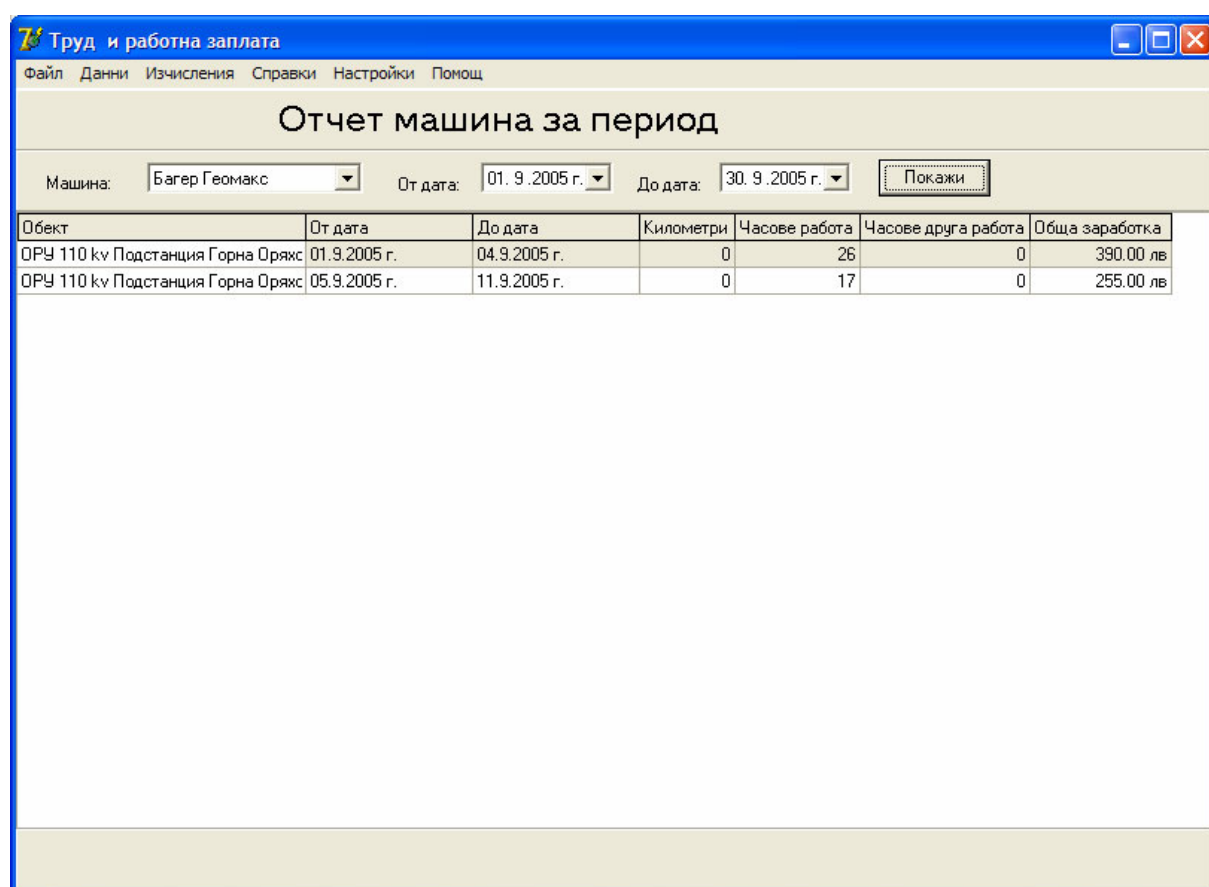
Обект	Коефициент	Изразходвана сума	Механизация	Външна работа	Оставаща сума
ОРУ 110 кв Подстанция Горна Оряховица	1.00 лв	3 903.30 лв	2 741.70 лв	320.00 лв	29 035.00 лв
TVS Горна Оряховица	1.00 лв	3 138.20 лв	0.00 лв	0.00 лв	19 111.80 лв
Кабел Ср 4 ТП 71-72 Велико Търново	1.00 лв	0.00 лв	0.00 лв	0.00 лв	0.00 лв

Фиг. 10 Отчет, показващ сумите, изразходвани за всички активни обекти за даден месец.

Следващият отчет, който може да се покаже, е за дадена машина в избран период каква работа е вършила. Това става като от менюто “Справки” се избере подменю “Механизация за период” (Фиг. 11). За да се изведат данни трябва да се избере първо машина и после границите на периода, за който искаме да е отчета, след което се натиска бутона “Покажи”. В дадената таблица полето “Обект” е името на обекта, на който избраната машина е работила, полетата “От дата” и “До дата” са съответно за границите на периода, за който е направен даден запис в базата за машината, но той е в границите на избрания първоначален период. Полето

“Километри” показва броя на изминатите от машината километри, които се заплащат. Полето “Часове работа” показва броя на часовете за извършената от машината работа. Ако машината е вършила друга работа, различна от основната ѝ дейност, това се отразява в полето “Часове друга работа”. И накрая в полето “Обща заработка” се извежда общата сума в левове за извършената работа и транспорт на машината за дадения обект в избрания период.

Както другите отчети, така и този отчет също може да бъде разпечатан и да се види страницата предварително, което става от менюто “Файл” и съответно “Печат” и “Преглед”.



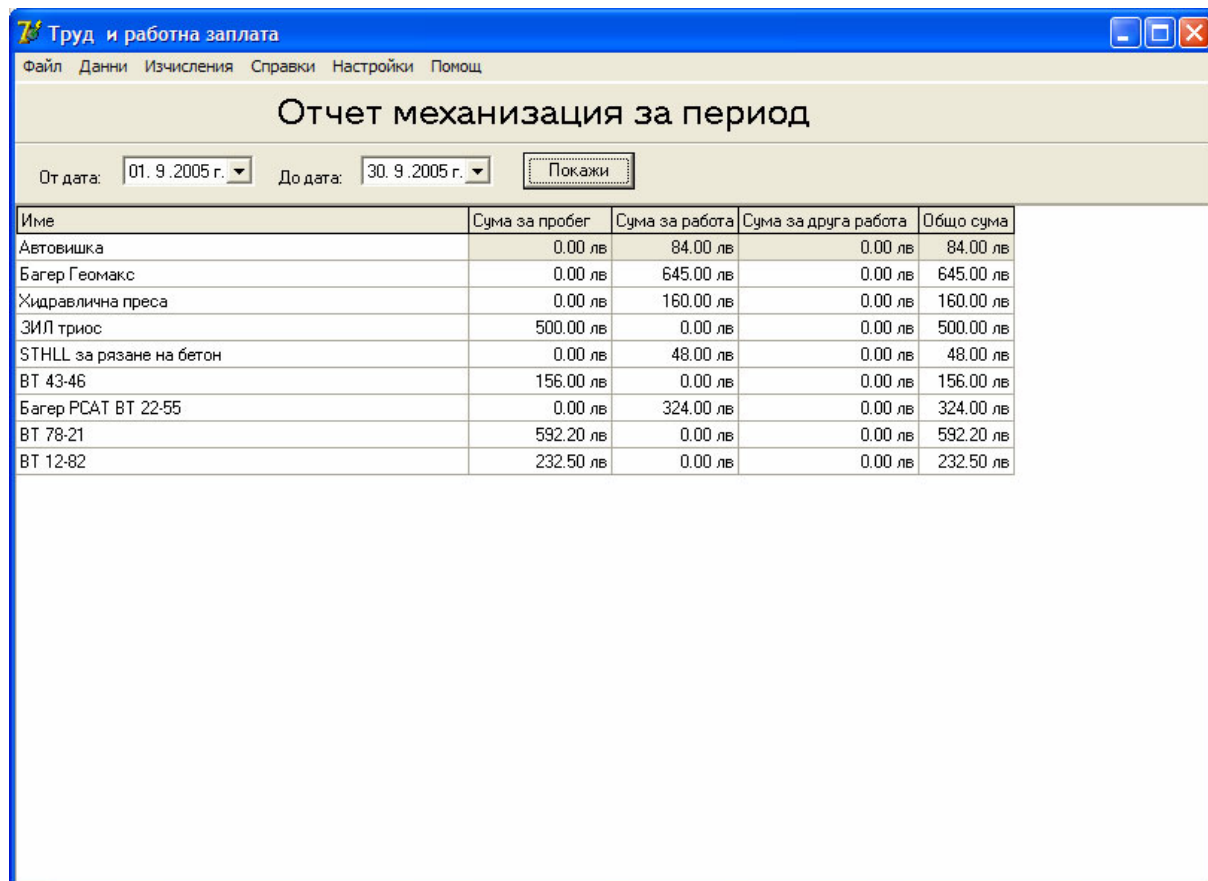
Обект	От дата	До дата	Километри	Часове работа	Часове друга работа	Обща заработка
ОРУ 110 kv Подстанция Горна Орякс	01.9.2005 г.	04.9.2005 г.	0	26	0	390.00 лв
ОРУ 110 kv Подстанция Горна Орякс	05.9.2005 г.	11.9.2005 г.	0	17	0	255.00 лв

Фиг. 11 Справка за дадена машина за извършената от нея работа в избран период.

И последният отчет, който може да се покаже, е общ за всички машини. Той се извежда като от менюто се избере “Справки” и после подменю “Общо механизация” (Фиг. 12). За да се изведат данните се избира първо периода и след това се натиска

бутона “Покажи”. Полето “Име” съдържа името на машината. Полето “Сума за пробег” отразява колко пари е струвало транспортирането на машината или ако е някакъв вид транспорт (камион или самосвал) колко пари е заработил за транспортни услуги. Полето “Сума за работа” съдържа стойността в лева за работа, извършена от машината, на всички обекти за дадения период. Полето “Сума за друга работа” съдържа стойността в лева на извършената работа по странични дейности на машината. И последното поле “Обща сума” съдържа сумата от трите предни полета, като обща за машината.

И последния отчет също може да бъде разпечатан и да се види страницата предварително, което става от менюто “Файл” и съответно “Печат” и “Преглед”.



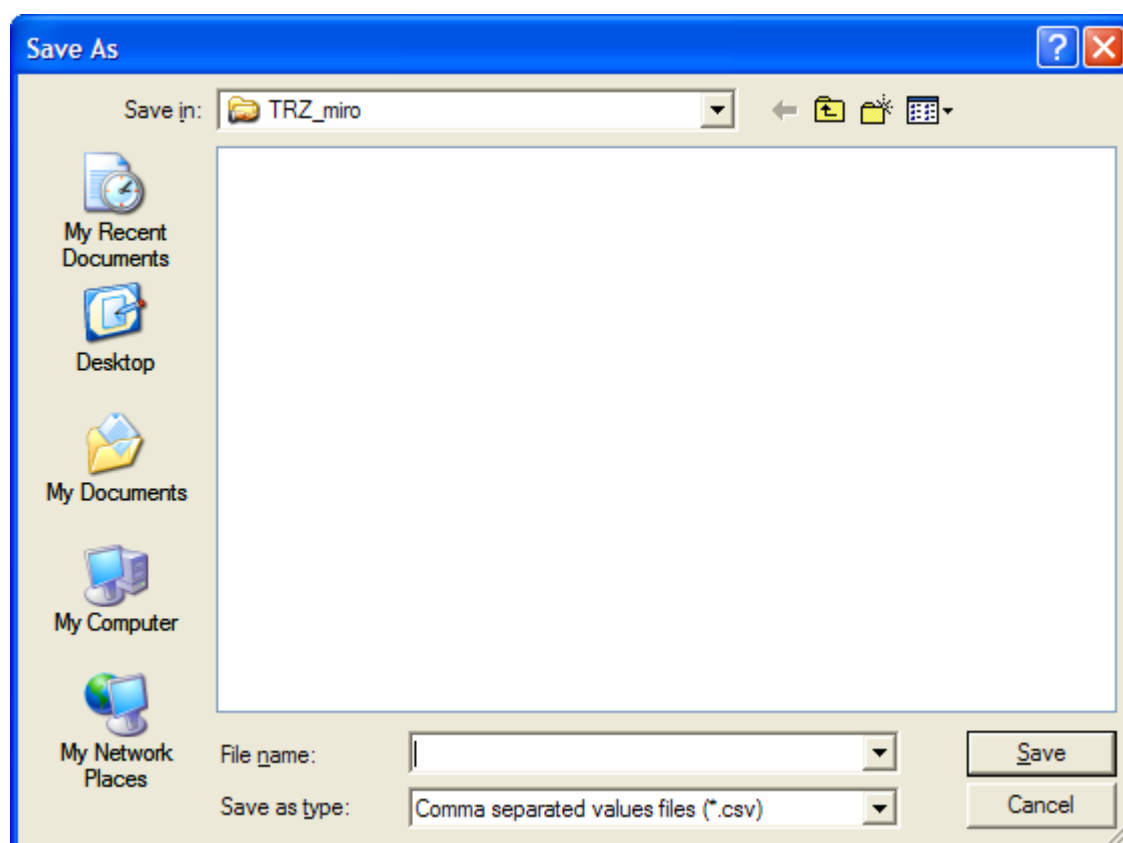
Име	Сума за пробег	Сума за работа	Сума за друга работа	Общо сума
Автовишка	0.00 лв	84.00 лв	0.00 лв	84.00 лв
Багер Геомакс	0.00 лв	645.00 лв	0.00 лв	645.00 лв
Хидравлична преса	0.00 лв	160.00 лв	0.00 лв	160.00 лв
ЗИЛ триос	500.00 лв	0.00 лв	0.00 лв	500.00 лв
STHLL за рязане на бетон	0.00 лв	48.00 лв	0.00 лв	48.00 лв
BT 43-46	156.00 лв	0.00 лв	0.00 лв	156.00 лв
Багер PCAT BT 22-55	0.00 лв	324.00 лв	0.00 лв	324.00 лв
BT 78-21	592.20 лв	0.00 лв	0.00 лв	592.20 лв
BT 12-82	232.50 лв	0.00 лв	0.00 лв	232.50 лв

Фиг. 12 Отчет за всички машини за даден период.

4.3 Допълнителни характеристики и функционалност.

Освен описаните по-горе възможности на програмата ще се спрем и на някои допълнителни като част от тях бяха споменати преди, но тук ще ги разгледаме малко по-подробно.

Всяка таблица, показана в програмата, може да бъде експортирана в Excel формат или в CSV формат. Първо трябва да се кликне върху таблицата, за да приеме фокуса, тъй като на някои места в програмата има показани повече от една таблица едновременно. След това се избира от менюто “Файл” подменюто “Експорт”. Това извиква диалогов прозорец (Фиг. 13), в който трябва да се въведе името на файла, директорията, където да се запише и какъв тип да е файла (xls или csv). За да се запише файлът се натиска бутона “Save”.

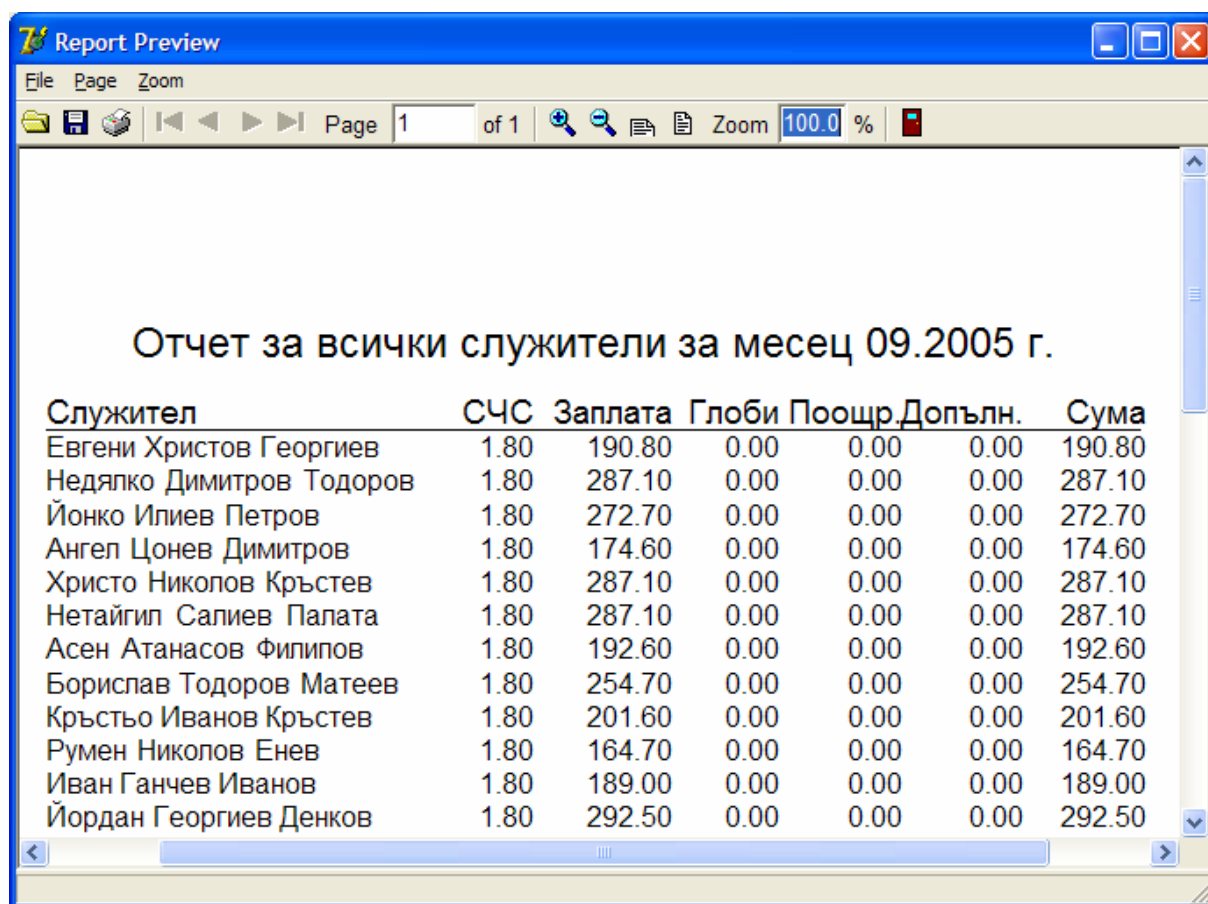


Фиг. 13 Диалогов прозорец за записване на файлове.

Друга възможност, която предоставя програмата, е да се преглеждат страници преди да се разпечатат. Тъй като само отчетите могат да се разпечатват, съответно само

те могат да бъдат видяни как ще изглеждат преди печат. За да се разгледа съответния отчет, когато той е активния прозорец, трябва да се избере от менюто “Файл” подменю “Преглед”, след което се появява нов прозорец, в който се показва как ще изглежда отчетът (Фиг. 14).

За да се разпечата отчетът, трябва той да е активният прозорец и да се избере от менюто “Файл” и после подменю “Печат”. Това направо ще разпечата дадения отчет. Друг вариант да се разпечата отчетът е от прозореца за преглед да се натисне бутон с иконка принтер или от менюто “File” да се избере подменю “Print”.



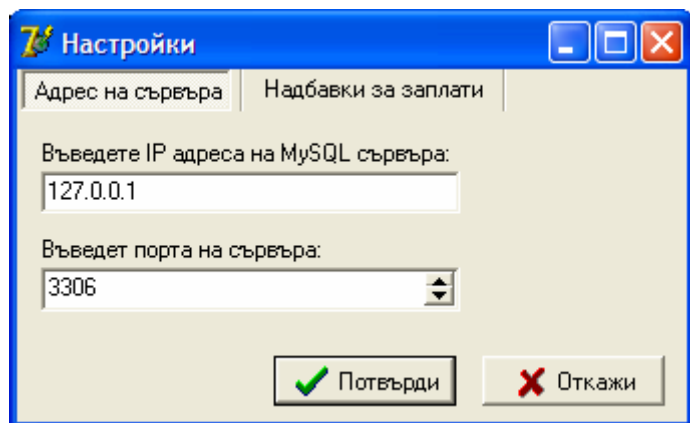
The screenshot shows a window titled "Report Preview" with a menu bar (File, Page, Zoom) and a toolbar. The main content area displays a table with the following data:

Служител	СЧС	Заплата	Глоби	Поощр.	Допълн.	Сума
Евгени Христов Георгиев	1.80	190.80	0.00	0.00	0.00	190.80
Недялко Димитров Тодоров	1.80	287.10	0.00	0.00	0.00	287.10
Йонко Илиев Петров	1.80	272.70	0.00	0.00	0.00	272.70
Ангел Цонев Димитров	1.80	174.60	0.00	0.00	0.00	174.60
Христо Николов Кръстев	1.80	287.10	0.00	0.00	0.00	287.10
Нетайгил Салиев Палата	1.80	287.10	0.00	0.00	0.00	287.10
Асен Атанасов Филипов	1.80	192.60	0.00	0.00	0.00	192.60
Борислав Тодоров Матеев	1.80	254.70	0.00	0.00	0.00	254.70
Кръстьо Иванов Кръстев	1.80	201.60	0.00	0.00	0.00	201.60
Румен Николов Енев	1.80	164.70	0.00	0.00	0.00	164.70
Иван Ганчев Иванов	1.80	189.00	0.00	0.00	0.00	189.00
Йордан Георгиев Денков	1.80	292.50	0.00	0.00	0.00	292.50

Фиг. 14 Преглед на отчет за разпечатване.

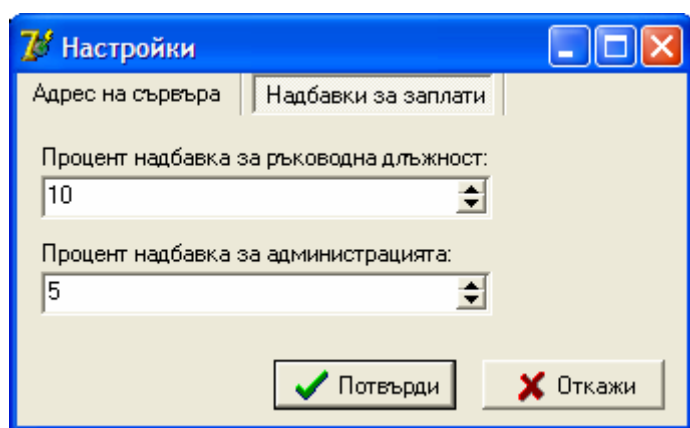
Също така е предвидено възможност да се избира към кой MySQL сървър да се върже програмата. Това може да стане като от менюто “Настройки” се избере “Връзка към сървъра” (Фиг. 15). Тука трябва да се въведе на кой IP адрес се намира сървърът и съответно на кой порт да се осъществи връзка с него. По подразбиране порта на

MySQL сървъра е 3306. От съображения за сигурност той може да се смени, тъй като има някои вируси, които използват уязвимостта на сървъра и правят атаки на този порт, ако има стартиран сървър.



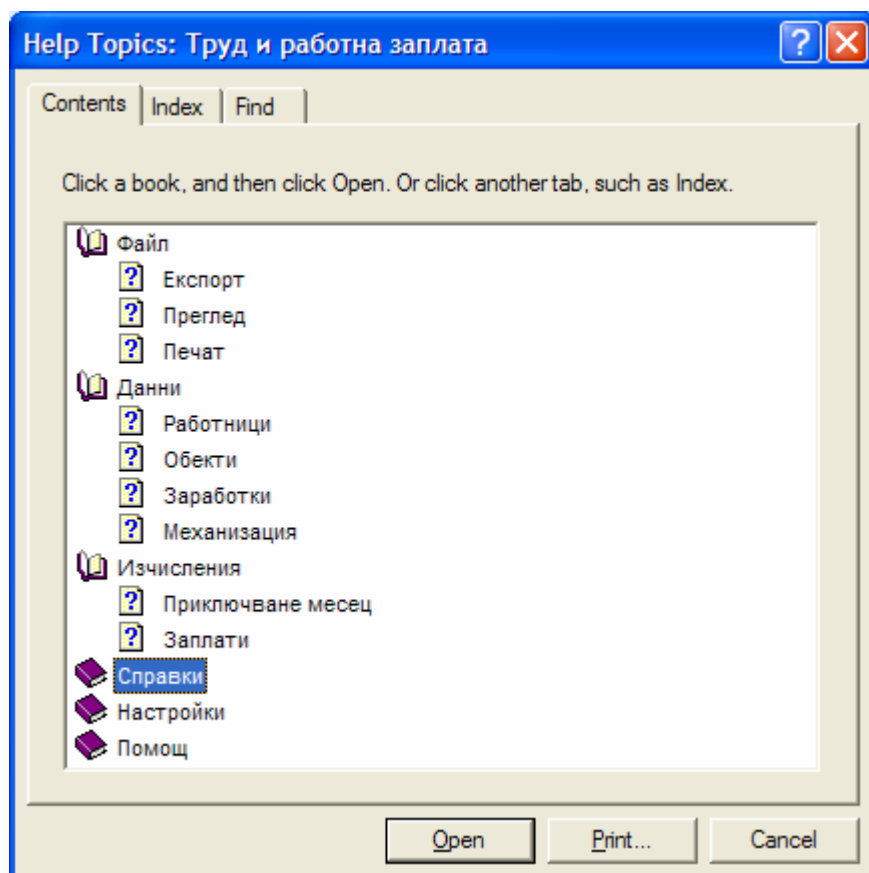
Фиг. 15 Настройки за свързване към сървъра

Програмата предлага да се направят и някои настройки за работната заплата за някои категории служители. Това може да стане като от менюто “Настройки” се избере подменюто “Процент заплати” (Фиг. 16). Възможно е да се зададе процент за увеличаване или намаляване на заплатите на администрацията, която се формира от средната часова ставка на всички работници. Ако този процент се увеличи над нула, тогава получената средна ставка ще се увеличи с толкова процента, колкото е дадената цифра и така ще се изчисли месечната заплата на администрацията, съответно ако се намали под нула заплатите на администрацията ще се намалят с въведения процент под средната ставка. Другата опция е за промяна на заплатите на ръководителите, като за тях може да се задава само положителен процент и той определя колко ще се увеличи заплатата на ръководители на дадени отдели, спрямо заработеното от тях.

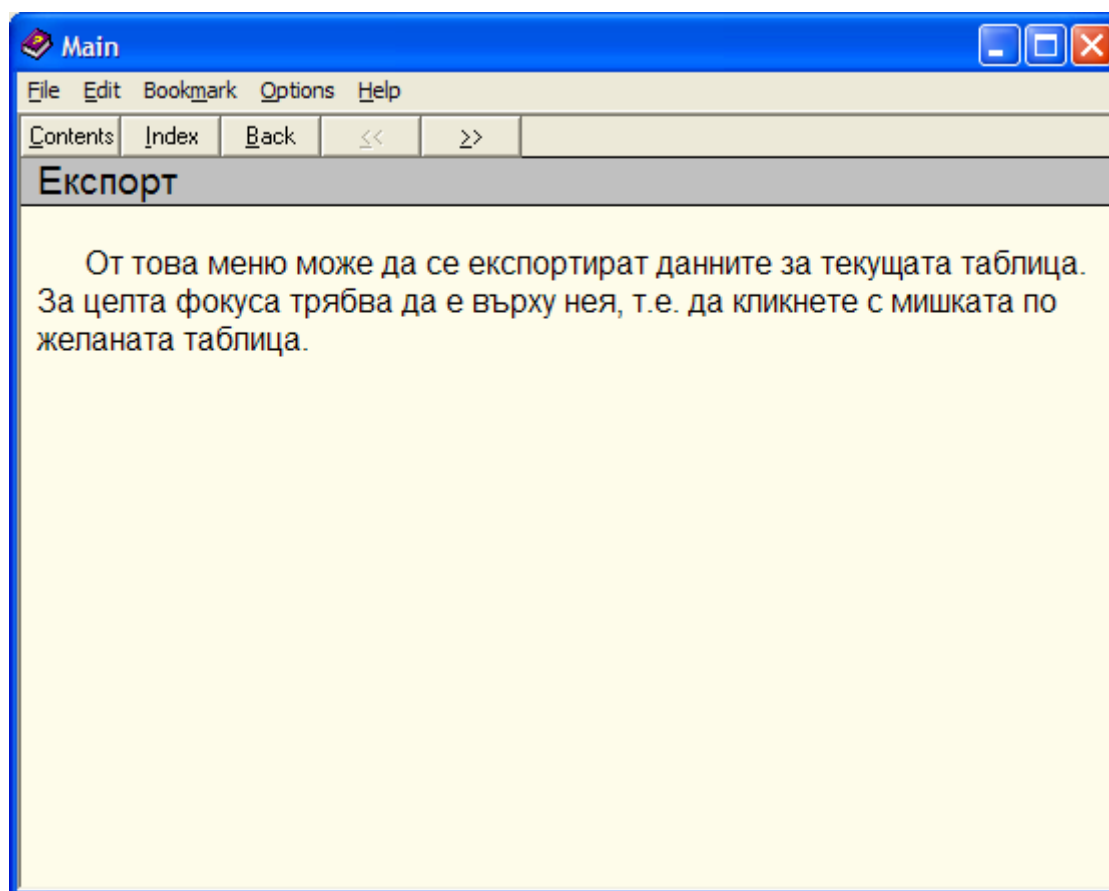


Фиг. 16 Настройки за процента на заплатата на администрацията и ръководителите

От менюто "Помощ" и като се избере подменю "Съдържание" може да се покаже помощният екран на програмата, където има кратко описание на функциите на програмата и начина на работа с нея. На фиг. 17 и фиг. 18 са показани съответно съдържанието на помощния файл и една от темите на помощната информация.

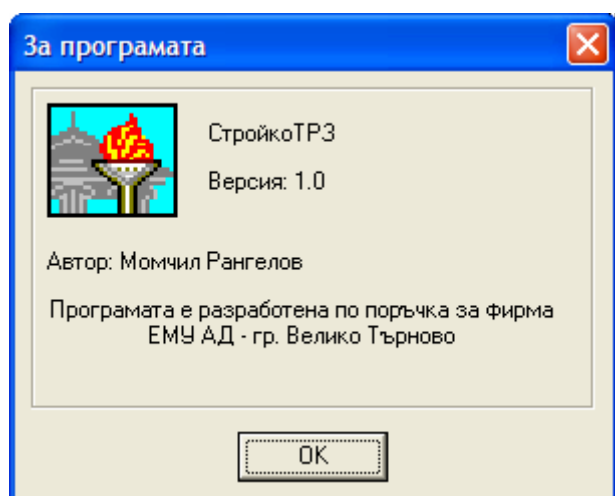


Фиг. 17 Съдържание на помощния файл



Фиг. 18 Кратка информация за експортиране на данните.

От менюто “Помощ” и подменю “За програмата” може да се види кратка информация за самата програма (Фиг. 19)



Фиг. 19 Информация за програмата.

ЗАКЛЮЧЕНИЕ

Според представянето на готовия програмен продукт и неговото тестване дипломантът счита, че са изпълнени основните цели, поставени в настоящата дипломна работа, които включват разработването на функциониращо приложение, което да даде възможност на фирмата поръчител да формира максимално гъвкаво заплатите на работещите в нея служители според заработеното и според това как върви изпълнението на обекта и да се правят детайлни справки, от които може да се установи дали нещо не върви според предварителните разчети и срокове, което да даде възможност да се вземат навременни мерки.

Тази функционалност ще допринесе за по-добро планиране на средствата и да се избегнат загуби при изпълнението на някои строителни обекти. Също така за по-обективно заплащане на работниците в зависимост от тяхната производителност.

Разработеното приложение дава възможност за въвеждане на един модерен и справедлив начин на определяне на работната заплата, който се практикува в много фирми в Европа и Америка. Наред с това ще създаде предпоставки за стимулиране на производителността на работниците, както и може да ограничи в някои случаи финансови загуби на фирмата.

Като недостатък може да се посочи, че приложението е проектирано за използване от един потребител и при работа на повече човека едновременно могат да възникнат проблеми. Също така няма възможност за определяне на различни права на достъп за служители на администрацията.

Изброените недостатъци, като и подобряване и добавяне на функционалност според възникнали нови нужди на клиента и оптимизиране на бързодействието на приложението могат да послужат за по-следваща база за неговото развитие. Друга възможност е да се анализират начините на определяне на заплатите и в други фирми от строителния сектор и да се доразвие приложението, така че да може да се използват от по-широк кръг клиенти в тази сфера.

БИБЛИОГРАФИЯ

- [1] Фирмен сайт на **Перфект Софтуер ООД** – разработка на приложения за строителство, счетоводство и др.: <http://www.persof.com/site/>
- [2] Фирмен сайт на **Микроинвест ООД** – разработка на софтуер за автоматизация на офиса: <http://www.microinvest.net/>
- [3] Фирмен сайт на **Nefia Consult** - програмен продукт за управление на строителството: www.nefia.com
- [4] Фирмен сайт на **БизнесСофт ЕООД** - разработка на софтуер за управление на бизнеса на строителни, производствени и търговски фирми: <http://www.businessoft.bg/>
- [5] Bo-Christer Bjork (1995). Requirements and Information Structures for Building Product Data Models. Dissertation for the degree of Doctor of Technology, Helsinki University of Technology, <http://lib.tkk.fi/Diss/199X/isbn9513860302/isbn9513860302.pdf>
- [6] Documentation for ZeosLib, <http://sourceforge.net/projects/zeoslib/>
- [7] Marco Cantu (2003). Mastering Delphi 7. Sybex
- [8] Martin Fischer and John Kunz (2004). The Scope and Role of Information Technology in Construction. CIFE Technical Report #156, Stanford University, <http://www.stanford.edu/group/CIFE/online.publications/TR156.pdf>
- [9] MySQL 4.1 Reference Manual, <http://dev.mysql.com/doc/refman/4.1/en/index.html>

ПРИЛОЖЕНИЕ

(Някои резултати от работата на програмата)

Отчет за всички служители за месец 09.2005 г.

Служител	СЧС	Заплата	Глоби	Поощр.	Допълн.	Сума
Евгени Христов Георгиев	1.71	249.66	0.00	0.00	0.00	249.66
Недялко Димитров Тодоров	1.71	255.64	0.00	0.00	0.00	281.21
Йонко Илиев Петров	1.71	206.05	0.00	0.00	0.00	220.00
Ангел Цонев Димитров	1.71	355.00	0.00	0.00	0.00	355.00
Христо Николов Кръстев	1.71	300.16	0.00	0.00	0.00	300.23
Нетайгил Салиев Палата	1.71	238.54	10.00	0.00	0.00	228.54
Асен Атанасов Филипков	1.71	206.91	0.00	0.00	0.00	220.00
Борислав Тодоров Матеев	1.71	344.00	0.00	15.00	0.00	359.00
Кръстьо Иванов Кръстев	1.71	182.97	0.00	0.00	0.00	220.00
Румен Николов Енев	1.71	300.16	0.00	0.00	0.00	300.16
Иван Ганчев Иванов	1.71	148.77	0.00	0.00	0.00	220.00
Йордан Георгиев Денков	1.71	248.80	10.00	0.00	0.00	238.80
Иван Гатев Иванов	1.70	210.96	0.00	0.00	0.00	220.00
Стефан Йорданов Стефанов	1.71	165.87	0.00	0.00	0.00	220.00
Марин Стоянов Гешев	1.71	133.38	0.00	0.00	0.00	220.00
Димитър Илиев Димитров	1.71	211.18	0.00	0.00	0.00	220.00
Симеон Стефанов Стефанов	1.71	111.15	0.00	0.00	0.00	220.00
Николай Добрев Минчев	1.71	226.57	0.00	15.00	0.00	241.57
Христо Христов	1.71	300.16	0.00	0.00	0.00	330.25
Ангел Димитров Чолаков	1.70	146.20	0.00	0.00	0.00	250.00
Веселин Йорданов Иванов	1.70	406.30	0.00	0.00	0.00	406.30
Костадин Милков Гърдев	1.70	176.80	0.00	0.00	0.00	250.00
Юри Русев Найденов	1.70	232.90	0.00	0.00	0.00	250.00
Марин Стоянов Маринов	1.70	299.20	0.00	0.00	0.00	299.20
Юри Стефанов Савов	1.70	289.00	0.00	0.00	0.00	289.00
Асен Георгиев Русев	1.70	192.10	0.00	0.00	0.00	250.00
Румен Георгиев Русев	1.70	207.40	0.00	0.00	0.00	250.00
Михаил Игнатов Сотиров	1.70	195.50	0.00	0.00	0.00	250.00
Гошо Стоянов Янков	1.70	229.50	0.00	0.00	0.00	260.00

Отчет за обект ОРУ 110 kv Подстанция Горна Оряхови

Служители

Работник	От дата	До дата	Часове	Извънр.	Празн.
Евгени Христов Георгиев	01.09.2005	04.09.2005	16	4	8
Недялко Димитров Тодоров	01.09.2005	04.09.2005	16	0	0
Йонко Илиев Петров	01.09.2005	04.09.2005	16	4	0
Ангел Цонев Димитров	01.09.2005	04.09.2005	16	0	0
Христо Николов Кръстев	01.09.2005	04.09.2005	16	4	0
Нетайгил Салиев Палата	01.09.2005	04.09.2005	16	4	0
Асен Атанасов Филипов	01.09.2005	04.09.2005	16	4	0
Борислав Тодоров Матеев	01.09.2005	04.09.2005	16	0	0
Кръстьо Иванов Кръстев	01.09.2005	04.09.2005	16	0	8
Румен Николов Енев	01.09.2005	04.09.2005	16	0	0
Иван Ганчев Иванов	01.09.2005	04.09.2005	16	0	0
Йордан Георгиев Денков	01.09.2005	04.09.2005	16	4	8
Иван Гатев ИвановЗ	01.09.2005	04.09.2005	16	0	0
Стефан Йорданов Стефано	01.09.2005	04.09.2005	16	4	8
Марин Стоянов Гешев	01.09.2005	04.09.2005	16	0	0
Димитър Илиев Димитров	01.09.2005	04.09.2005	16	4	8
Симеон Стефанов Стефано	01.09.2005	04.09.2005	16	0	0
Николай Добрев Минчев	01.09.2005	04.09.2005	16	2	0
Христо Христов	01.09.2005	04.09.2005	16	2	0
Йордан Георгиев Денков	05.09.2005	11.09.2005	40	9	0
Недялко Димитров Тодоров	05.09.2005	11.09.2005	40	9	0
Йонко Илиев Петров	05.09.2005	11.09.2005	40	9	0
Ангел Цонев Димитров	05.09.2005	11.09.2005	29	0	0
Христо Николов Кръстев	05.09.2005	11.09.2005	40	9	0
Нетайгил Салиев Палата	05.09.2005	11.09.2005	40	9	0
Асен Атанасов Филипов	05.09.2005	11.09.2005	29	0	0
Борислав Тодоров Матеев	05.09.2005	11.09.2005	40	9	0
Кръстьо Иванов Кръстев	05.09.2005	11.09.2005	29	0	0
Румен Николов Енев	05.09.2005	11.09.2005	40	9	0
Иван Ганчев Иванов	05.09.2005	11.09.2005	19	0	0
Йордан Георгиев Денков	05.09.2005	11.09.2005	30	0	0
Стефан Йорданов Стефано	05.09.2005	11.09.2005	19	0	0
Марин Стоянов Гешев	05.09.2005	11.09.2005	10	0	0
Димитър Илиев Димитров	05.09.2005	11.09.2005	40	9	0
Симеон Стефанов Стефано	05.09.2005	11.09.2005	19	0	0
Николай Добрев Минчев	05.09.2005	11.09.2005	40	9	0
Христо Христов	05.09.2005	11.09.2005	19	0	0
Евгени Христов Георгиев	12.09.2005	18.09.2005	40	8	8
Недялко Димитров Тодоров	12.09.2005	18.09.2005	40	0	0
Йонко Илиев Петров	12.09.2005	18.09.2005	20	6	8

Работник	От дата	До дата	Часове	Извънр.	Празн.
Ангел Цонев Димитров	12.09.2005	18.09.2005	24	0	0
Христо Николов Кръстев	12.09.2005	18.09.2005	40	8	8
Нетайгил Салиев Палата	12.09.2005	18.09.2005	32	0	0
Асен Атанасов Филипов	12.09.2005	18.09.2005	40	0	0
Борислав Тодоров Матеев	12.09.2005	18.09.2005	40	0	0
Кръстьо Иванов Кръстев	12.09.2005	18.09.2005	32	4	4
Иван Ганчев Иванов	12.09.2005	18.09.2005	40	8	0
Йордан Георгиев Денков	12.09.2005	18.09.2005	24	0	0
Стефан Йорданов Стефано	12.09.2005	18.09.2005	40	0	0
Марин Стоянов Гешев	12.09.2005	18.09.2005	40	8	0
Димитър Илиев Димитров	12.09.2005	18.09.2005	32	0	0
Симеон Стефанов Стефано	12.09.2005	18.09.2005	24	4	0
Николай Добрев Минчев	12.09.2005	18.09.2005	40	8	4
Христо Христов	12.09.2005	18.09.2005	40	8	0
Ангел Цонев Димитров	19.09.2005	25.09.2005	40	4	0
Христо Николов Кръстев	19.09.2005	25.09.2005	24	0	0
Асен Атанасов Филипов	19.09.2005	25.09.2005	24	4	0
Недялко Димитров Тодоров	19.09.2005	25.09.2005	40	0	0
Евгени Христов Георгиев	26.09.2005	30.09.2005	40	0	0
Нетайгил Салиев Палата	26.09.2005	30.09.2005	32	0	0

Механизация

Машина	От дата	До дата	Пробег км	Работа	Др. работа
Автовишка	01.09.2005	04.09.2005	0	14	0
Багер Геома	01.09.2005	04.09.2005	0	26	0
Хидравличн	01.09.2005	04.09.2005	0	8	0
ЗИЛ триос	01.09.2005	04.09.2005	125	0	0
STHLL за ря	01.09.2005	04.09.2005	0	2	0
BT 43-46	01.09.2005	04.09.2005	52	0	0
Багер Геома	05.09.2005	11.09.2005	0	17	0
Багер PCAT	05.09.2005	11.09.2005	0	18	0
STHLL за ря	05.09.2005	11.09.2005	0	2	0
BT 78-21	05.09.2005	11.09.2005	329	0	0
BT 12-82	05.09.2005	11.09.2005	93	0	0

Външна работа

От дата	До дата	Сума	Коментар
01.09.2005	04.09.2005	100.00	Граждански договор - Румен Асенов Русев
01.09.2005	04.09.2005	120.00	Граждански договор - Михаил Сотиров Ма
01.09.2005	04.09.2005	100.00	Граждански договор - Юрий Стоянов Саво

Отчет за всички машини
от 01.09.2005 г. до 30.09.2005 г.

Име на машина	Сума пробег	Работа	Др. работа	Всичко
Автовишка	0.00	84.00	0.00	84.00
Багер Геомакс	0.00	645.00	0.00	645.00
Хидравлична преса	0.00	160.00	0.00	160.00
ЗИЛ триос	500.00	0.00	0.00	500.00
STHLL за рязане на бетон	0.00	44.00	0.00	44.00
BT 43-46	156.00	0.00	0.00	156.00
Багер PCAT BT 22-55	0.00	324.00	0.00	324.00
BT 78-21	592.20	0.00	0.00	592.20
BT 12-82	232.50	0.00	0.00	232.50

Microsoft Excel - WorkersList.xls

File Edit View Insert Format Tools Data Window Help

A1 = Име

1	A	B	C	D	E	F	G	H
	Име	Длъжност	Отдел	Основна заплата	Заплата на час	Месеци работа		
2	Евгени Христов Георгиев	Служител	Строителство	220	1.399285674	28		
3	Недялко Димитров Тодоров	Ръководител	Строителство	220	1.578214288	28		
4	Йонко Илиев Петров	Служител	Механизация	220	1.727941155	34		
5	Ангел Цонев Димитров	Служител	Търговия	355	1.226249933	8		
6	Христо Николов Кръстев	Служител	Администрация	220	1.508666635	15		
7	Нетайгил Салиев Палата	Служител	Строителство	220	1.551428676	21		
8	Асен Атанасов Филипов	Служител	Строителство	220	1.646153808	26		
9	Борислав Тодоров Матеев	Служител	Търговия	344	1.728333354	12		
10	Кръстьо Иванов Кръстев	Служител	Строителство	220	1.709428668	35		
11	Румен Николов Енев	Служител	Администрация	220	1.682666659	30		
12	Иван Ганчев Иванов	Служител	Строителство	220	1.872758746	29		
13	Йордан Георгиев Денков	Служител	Строителство	220	1.54935503	31		
14	Иван Гатев Иванов	Служител	Електроизграждане	220	1.815933943	39		
15	Стефан Йорданов Стефанов	Служител	Електроизграждане	220	1.642352939	34		
16	Марин Стоянов Гешев	Служител	Електроизграждане	220	1.519999981	24		
17	Димитър Илиев Димитров	Служител	Електроизграждане	220	1.6049999	24		
18	Симеон Стефанов Стефанов	Служител	Електроизграждане	220	1.699999928	27		
19	Николай Добрев Минчев	Служител	Строителство	220	1.5704	25		
20	Христо Христов	Ръководител	Администрация	300	2.329500198	40		
21	Ангел Димитров Чолаков	Служител	Строителство	250	1.842105389	38		
22	Веселин Йорданов Иванов	Служител	Строителство	250	1.664615273	26		
23	Костадин Милков Гърдев	Служител	Строителство	250	1.765517354	58		
24	Юри Русев Найденов	Служител	Механизация	250	1.946226478	53		
25	Марин Стоянов Маринов	Служител	Строителство	250	1.63538456	26		
26	Юри Стефанов Савов	Служител	Механизация	240	1.860606194	33		
27	Асен Георгиев Русев	Служител	Механизация	250	1.504347801	23		
28	Румен Георгиев Русев	Служител	Строителство	250	1.746756792	37		
29	Михаил Игнатов Сотиров	Служител	Електроизграждане	250	1.59124589	33		
30	Гошо Стоянов Янков	Служител	Механизация	260	2.14466691	60		
31	Мартин Иванов Асенов	Служител	Строителство	250	2.160000086	22		
32	Андрей Христов Андреев	Служител	Строителство	240	1.730000019	18		
33	Сали Сюлейманов Яшаров	Ръководител	Строителство	240	1.830000043	15		
34	Мишо Владимиров Добрев	Служител	Механизация	250	1.730000019	8		
35	Стефан Камачев Маринов	Служител	Строителство	260	2.070000071	8		

WorkersList/

Ready

Експортирани данни в Excel