

Методические указания
по курсовому проектированию
по курсу “ **Технология разработки программного обеспечения** “

для студентов специальности
230100.62 Информатика и вычислительная техника

2014

СОДЕРЖАНИЕ

| | |
|---|-----------|
| 1. Тематика курсовых работ..... | 3 |
| 2. Структура курсового проекта..... | 4 |
| 3. Правила оформления курсовой работы..... | 20 |
| 4. Литература..... | 22 |

1. ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ

1. Инженерный калькулятор.
2. Электронные часы с будильником.
3. Часы со стрелками.
4. Изображения графиков различных функций.
5. Вывод диаграмм табличных значений.
8. Медиаплеер: проигрывание аудио- и видео- файлов.
9. Проигрыватель Audio CD.
10. MP3 Player.
11. Программа тестирования.
12. Программа анкетирования.
13. "Записная книжка".
14. "Ежедневник".

2. СТРУКТУРА КУРСОВОГО ПРОЕКТА

Примерная структура курсового проекта:

- Титульный лист
- Содержание
- Введение
- 1. Разработка эскизного и технического проекта программы
 - 1.1. Назначение и область применения
 - 1.2. Технические характеристики
- 2. Разработка рабочего проекта
 - 2.1 Разработка программы
 - 2.2 Спецификация программы
 - 2.3 Текст программы
 - 2.4 Описание программы
 - 2.5 Тестирование программы
- 3. Оценка программы
- Заключение
- Список использованной литературы
- Приложения

Содержание.

В нем содержится название глав и параграфов с указанием страниц.

Введение.

Во введении обосновывается актуальность выбранной темы, ее важность. Формируются цели и задачи курсового проекта. Желательно, чтобы было отражено значение рассматриваемых вопросов, помимо этого во введении должна обосновываться структура курсового проекта.

1. Разработка эскизного и технического проекта программы.

Данный параграф состоит из:

1.1. Назначение и область применения.

В данном разделе указывают назначение программы и краткую характеристику области применения программы.

Например: Данная программа является развлекательной игровой программой. Областью применения данной программы является организация досуга человека, работающего с компьютером. Так как, поставлена задача, разработать приложение под операционную систему Windows, то использоваться программа может только в среде 32 битной операционной системы Windows 9x/ME/NT/2000/XP.

1.2. Технические характеристики.

Данный пункт должен содержать следующие подпункты:

1.2.1. Постановка задачи.

Здесь дается точное описание исходных данных, условий задачи и целей ее решения. На этом этапе условия задачи, записанные в форме различных словесных описаний, необходимо выразить при помощи модели предметной области (МПО). В этом подпункте могут быть описаны основные приемы программирования и типы данных, используемые при решении аналогичных задач. Далее описываются возможные пути решения задачи с указанием их достоинств и недостатков. Выбирается и обосновывается метод решения задачи.

1.2.2. Описание алгоритма.

Здесь дается обобщенное словесное описание алгоритма решения поставленной задачи, излагаются основные требования к алгоритму и пути их реализации. Приводится схема алгоритма, состоящая из укрупненных модулей. Дается пояснение назначения и состава каждого модуля.

На следующем этапе каждый модуль детализируется.

Например: Общий не детализированный алгоритм выполнения данной программы можно представить в виде блок-схемы изображенной на рис. 1.

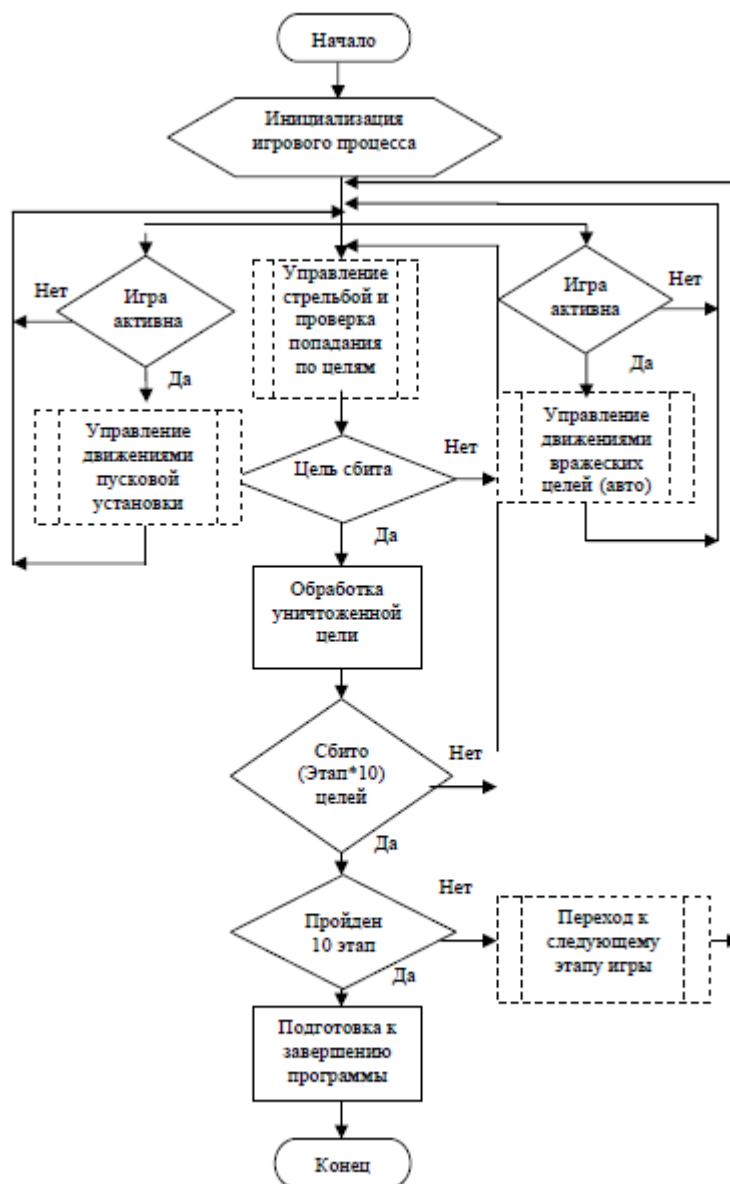


Рис. 1. Общая блок-схема игры «Сбей самолет».

Словесно эту блок-схему можно описать как следующую последовательность действий:

1. Начало;
2. Инициализация игрового процесса (подготовка данных, параметров и т.д.);
3. Активация трех параллельных процессов управления;
 - 3.1. Управление движениями пусковой установки (с участием пользователя);
 - 3.1.1. В случае поступления команды перемещения пусковой установки, пусковая установка перемещается в соответствии с командой и возвращает управление пункту 3;
 - 3.2. Управление стрельбой по самолетам и проверка попадания по ним;
 - 3.2.1. В случае поступления команды на выстрел, происходит запуск объекта «ракета», которая движется вертикально, пока не выйдет за пределы видимой области, либо пока не попадет в цель. Если цель не поражена,

управление возвращается пункту 3. В случае если цель поражена, управление передается пункту 4;

3.3. Управление движениями вражеских целей (в автоматическом режиме)

3.3.1. Автономный процесс управления движением самолетов контролирует перемещение целей по экрану. В случае выхода цели за пределы видимой области реализуется «разворот» цели и ее движение в обратном направлении. Здесь же рассчитываются скорости движения и количество отображаемых целей. По завершении процесс передает управление пункту 3. И так далее.

В общем алгоритме можно выделить основные подзадачи:

1. Управление движениями вражеских целей.
2. Управление стрельбой, проверка попадания по целям.
3. Управление движениями пусковой установки.
4. Переход к следующему этапу игры.

Поскольку наиболее важными и сложно реализуемыми частями программы являются именно эти процедуры, было принято решение изобразить их более детализированные алгоритмы в виде блок-схем.

То есть здесь, выбранные модули детализируются.

Для оформления данного подпункта используется **ГОСТ 19.701-90 ЕСПД**. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.

1.2.3. Организация входных и выходных данных.

Здесь содержится описание и обоснование выбора метода организации входных и выходных данных.

Например: В программе были использованы три типа входных данных:

1. Графические, представленные в виде файлов в формате bmp.
2. Звуковые данные, представленные в виде файлов в формате wav.
3. Видео данные, представленные в виде файлов в формате avi.

Существует два способа ввода этих данных в программу:

- прямая загрузка в режиме разработки форм;
- программная загрузка данных из файла в соответствующий компонент в режиме работы программы.

Выходные данные представляют собой динамически изменяемую графическую информацию, выводимую на дисплей ЭВМ.

1.2.4. Выбор состава технических и программных средств.

В данном подпункте на основании разработанного алгоритма делается вывод о необходимости использования того или иного языка программирования. Перечисляются достоинства выбранной среды программирования. Определяются технические средства, необходимые для оптимальной работы будущей программы.

Например: Исходя из целей курсовой работы, требуется создать программу, работающую под ОС Windows. В связи с тем, что разработанный алгоритм подразумевает использование графических возможностей, звука,

анимации и параллельных процессов взаимодействия с пользователем, было принято решение использовать интегрированную среду разработки программ Borland Delphi 6. Данная среда разработки позволяет достаточно быстро создавать приложения для Windows.

Технические средства, достаточные для оптимальной работы программы:

- Windows 9x/ME/NT/2000/XP.
- Видео память не менее 1Mb. Ускоритель графики не требуется.
- Оперативной памяти не менее 16 Mb.
- Процессор Intel Pentium 166 и выше.
- Объем свободного места на жестком диске не менее 500 Mb.
- Манипуляторы: клавиатура _и мышь.

2. Разработка рабочего проекта.

Данный параграф состоит из:

2.1. Разработка программы.

Проект Windows-окна должен быть представлен в виде графической схемы, на которой расположены все визуальные и не визуальные компоненты, разрабатываемого интерфейса. Компоненты на схеме должны быть пронумерованы. После схемы приводится расшифровка изображенных на схеме компонентов: название и имя компонента, назначение в программе, событие на которое данный компонент откликается.

Интерфейс с пользователем (все виды взаимодействия с пользователем: интерактивный режим, обработка ошибок, автономная работа).

- Определить требования к интерфейсу пользователя.
- Общие спецификации пользовательских функций.
- Описание основных запросов и алгоритмов обработки данных.

Требования к графическому интерфейсу:

- Содержать привычные и понятные пользователю пункты меню, соответствующие функциям обработки.

- Ориентироваться на пользователя, который общается с программой на внешнем уровне взаимодействия.

- Удовлетворять правилу "шести" – в одну линейку меню включать не более 6 понятий, каждое из которых содержит не более 6 опций.

- Сохранять стандартизированное назначение и местоположение на экране графических объектов.

При создании программы в системе Delphi, к ним предъявляются следующие требования:

- многооконное (не менее 2-х окон);
- наличие меню;
- наличие всплывающих подсказок;
- защита от некорректного ввода данных;
- обработка исключительных ситуаций;
- наличие сведений о программе;
- наличие сведений об авторе.

Например: Как уже отмечалось, для разработки приложения игры «Сбей самолет» используется среда визуального программирования Delphi. В данной среде был создан проект с названием HitThePlane.dpr. Данный проект состоит из одного программного модуля и одной формы: Form1: (см. рис. 2)



Рис. 2. Form1. Главная форма программы «Сбей самолет».

Главная форма представляет собой стандартное окно класса TForm и содержит следующие необходимые для функционирования программы компоненты, расположенные ниже, в порядке их размещения:

1. Form1: TForm – Основная форма проекта. Контейнер всех объектов.
2. StartTimer: TTimer – Таймер, активирующий игровой процесс.
3. EnemyTimer: TTimer – Таймер для управления движением самолетов.
4. FireTimer: TTimer – Таймер для контроля стрельбы и движения ракеты.
5. Music: TMediaPlayer – Проигрыватель фоновой музыки.

И так далее.

При написании программы не следует забывать о хорошем стиле программирования. После заголовка процедуры или функции записывается комментарий, содержащий поясняющий текст, а именно: назначение подпрограммы; перечень и назначения параметров; их тип.

2.2. Спецификация программы.

В данном подпункте приводится точное название программы и ее состав. Оформляется в соответствии с ГОСТ 19.202-78 ЕСПД.

Например: Исполнимый файл программы «Сбей Самолет» имеет название Hit The Plane.exe и расположен в каталоге \Сбей Самолет.

Помимо исполнимого файла в проект включены следующие файлы (табл. 2):

Таблица 2

Таблица файлов входящих в проект

| Наименование | Обозначение | Примечание |
|-----------------|------------------------------|-------------------------------------|
| HitThePlane.dpr | Файл проекта | Связывает все файлы приложения |
| HitThePlane.cfg | Файл конфигурации | Конфигурация |
| HitThePlane.dof | Файл параметров проекта | Текущие установки проекта |
| HitThePlane.res | Файл ресурсов | Содержит пиктограммы, версию и т.д. |
| Unit1.pas | Программный модуль для Form1 | Определяет функциональность формы |
| Unit1.dcu | Объектный файл для Unit1.pas | Откомпилированная версия Unit1.pas |
| Unit1.dfm | Файл формы №1 | Список свойств компонентов формы |
| IMAGES\ | Папка с графическими файлами | Следующие 10 файлов находятся в ней |
| beginning.bmp | Файл рисунка | Заставка начала |
| victory.bmp | Файл рисунка | Заставка победы |
| theend.bmp | Файл рисунка | Заставка финала |
| Back.bmp | Файл рисунка | Задний фон |
| plane1.bmp | Файл рисунка | Самолет вправо |
| plane2.bmp | Файл рисунка | Самолет влево |
| jeep.avi | Файл анимации | Анимация пусковой установки |
| SOUND\ | Папка со звуковыми файлами | Следующие 9 файлов находятся в ней |
| BackMusic.mid | Файл полифонии | Фоновая музыка (формат MIDI) |
| Alarm.wav | Звуковой файл | Звук сирены |
| Boom.wav | Звуковой файл | Звук взрыва |
| Start.wav | Звуковой файл | Звук запуска двигателя джипа |
| Engine.WAV | Звуковой файл | Звук работы двигателя джипа |
| Plane.wav | Звуковой файл | Звук самолета |
| Rocket.wav | Звуковой файл | Звук ракеты |
| victory.wav | Звуковой файл | Мелодия после прохождения этапа |
| Theend.wav | Звуковой файл | Мелодия после прохождения игры |

Приводится полный листинг программы вместе с комментариями.

Например:

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, ExtCtrls,
```

```
StdCtrls, ComCtrls, MPlayer, mmsystem, Menus, Buttons;
```

```
type
```

```
TForm1 = class(TForm)
```

```
BackPanel: TPanel;
```

```
Back: TImage;
```

```
EnemyTimer: TTimer;
```

```
Plane1: TImage;
```

Plane2: TImage;
Enemy4: TImage;
Enemy5: TImage;
Enemy3: TImage;
Enemy2: TImage;
Enemy1: TImage;
Rocket: TPanel;
FireTimer: TTimer;
LedPanel: TPanel;
PtsLed: TLabel;

2.4. Описание программы.

Данный подпункт оформляется согласно ГОСТ 19.402-78 ЕСПД и должен содержать следующие подразделы:

- Общие сведения – должны быть указаны: обозначение и наименование программы, программное обеспечение, необходимое для функционирования программы, языки программирования, на которых написана программа.

- Функциональное назначение – должны быть указаны классы решаемых задач и/или назначение программы и сведения о функциональных ограничениях на применение.

- Используемые технические средства – должны быть указаны типы ЭВМ и устройств, которые используются при работе программы.

- Вызов и загрузка - должны быть указаны способ вызова программы с соответствующего носителя данных.

- Входные данные - должны быть указаны: характер, организация и предварительная подготовка входных данных, формат, описание и способ кодировки входных данных.

- Выходные данные - должны быть указаны: характер, организация и предварительная подготовка выходных данных, формат, описание и способ кодировки выходных данных.

Отдельные разделы можно объединять. Некоторые пункты этого раздела повторяют разделы технического проекта. Такие повторения предусмотрены ГОСТом, так как на этапе рабочего проекта возникают некоторые дополнения или изменения в составе технических средств. Здесь приводятся более конкретные и точные данные.

Например:

Общие сведения.

Разработанное приложение является развлекательной игровой программой и называется «Сбей Самолет». Программа разработана в среде Borland Delphi 6 на языке Object Pascal.

Функциональное назначение.

Программа предназначена для развлечения и организации досуга людей, работающих с ЭВМ.

Используемые технические средства.

Для стабильной работы данной программы требуются следующие технические и программные средства:

- Windows 9x/ME/NT/2000/XP.
- Видео память не менее 1Mb. Ускоритель графики не требуется.
- Оперативной памяти не менее 16 Mb.
- Процессор Intel Pentium 166 и выше.
- Объем свободного места на жестком диске не менее 500 Мб.
- Манипуляторы: клавиатура и мышь.

Вызов и загрузка.

Программа запускается на исполнение либо двойным щелчком левой кнопки мыши по файлу HitThePlane.exe в проводнике Windows, либо нажатием правой кнопки мыши по указанному файлу и выбором пункта меню “Открыть”.

Входные данные.

В программе были использованы три типа входных данных:

1. Графические, представленные в виде файлов в формате bmp.
2. Звуковые данные, представленные в виде файлов в формате wav.
3. Видео данные, представленные в виде файлов в формате avi.

Существует два способа ввода этих данных в программу:

1. Прямая загрузка в режиме разработки форм;
2. Программная загрузка данных из файла в соответствующий компонент в режиме работы программы.

Более наглядно входные данные отражены в табл. 3.

Таблица входных данных и их файлов

| Файл | Формат данных | Описание данных |
|---------------|------------------------------|-------------------------------------|
| IMAGES\ | Папка с графическими файлами | Следующие 10 файлов находятся в ней |
| beginning.bmp | Файл рисунка | Заставка начала |
| victory.bmp | Файл рисунка | Заставка победы |
| theend.bmp | Файл рисунка | Заставка финала |
| Back.bmp | Файл рисунка | Задний фон |
| plane1.bmp | Файл рисунка | Самолет вправо |
| plane2.bmp | Файл рисунка | Самолет влево |
| jeep.avi | Файл анимации | Анимация пусковой установки |
| SOUND\ | Папка со звуковыми файлами | Следующие 9 файлов находятся в ней |
| BackMusic.mid | Файл полифонии | Фоновая музыка (формат MIDI) |
| Alarm.wav | Звуковой файл | Звук сирены |
| Boom.wav | Звуковой файл | Звук взрыва |
| Start.wav | Звуковой файл | Звук запуска двигателя джипа |
| Engine.WAV | Звуковой файл | Звук работы двигателя джипа |
| Plane.wav | Звуковой файл | Звук самолета |
| Rocket.wav | Звуковой файл | Звук ракеты |
| victory.wav | Звуковой файл | Мелодия после прохождения этапа |
| Theend.wav | Звуковой файл | Мелодия после прохождения игры |

2.5. Тестирование программы.

Кратко описывается среда программирования. Проводятся основные команды, выполняемые при вводе и редактировании программы, команды записи программы на диск, чтение с диска. Перечисляются и описываются средства отладки.

Например: При разработке программы использовалась среда программирования Borland Delphi 6. Эта среда представляет собой набор средств быстрой разработки и отладки приложений для Windows.

При разработке приложения применялся визуальный редактор форм, редактор программного кода, палитра компонентов и инспектор объектов.

Основными операциями в процессе разработки являлись: создание форм, размещение компонентов, редактирование свойств этих компонентов, формирование визуального пользовательского интерфейса, написание кода программы для обработки событий, сохранение и компиляция проекта.

Помимо указанных операций применялись дополнительные операции для отладки созданного приложения. Для этого в составе Delphi 6 имеется интегральный отладчик, который позволяет обнаруживать и исправлять ошибки как на стадии написания кода, так и на стадии отладки работающей программы.

3. Оценка программы.

В данном параграфе проводится оценка созданного программного продукта.

Затраты на создание программного продукта складываются из расходов по оплате труда разработчика программы и расходов по оплате машинного времени при отладке программы:

$$Z_{СПП} = Z_{СПП}^{ЗП} + Z_{СПП}^{МЕ} + Z_{ОБЩ},$$

где

$Z_{СПП}$ - затраты на создание программного продукта;

$Z_{СПП}^{ЗП}$ - затраты на оплату труда разработчика программы;

$Z_{СПП}^{МЕ}$ - затраты на оплату машинного времени;

$Z_{ОБЩ}$ - общие затраты.

Расходы на оплату труда разработчика программы.

Расходы на оплату труда разработчика программы определяются путем умножения трудоёмкости создания программного продукта на среднюю часовую оплату программиста (с учётом коэффициента отчислений на социальные нужды):

$$Z_{СПП}^{ЗП} = t \cdot T_{\text{час.}}$$

Расчёт трудоёмкости создания программного продукта.

Трудоёмкость разработки программного продукта можно определить следующим образом:

$$t = t_O + t_A + t_B + t_{П} + t_D + t_{ОТ}$$

где

t_O - затраты труда на подготовку описания задачи;

t_A - затраты труда на разработку алгоритма решения задачи;

t_B - затраты труда на разработку блок-схемы алгоритма решения задачи;

$t_{П}$ - затраты труда на составление программы по готовой блок-схеме;

t_D - затраты труда на подготовку документации задачи;

$t_{ОТ}$ - затраты труда на отладку программы на ЭВМ при комплексной отладке задачи;

Составляющие затрат, в свою очередь можно вычислить через условное число операторов Q . Например, в нашем случае число операторов в отлаженной программе $Q = 6327$.

Расчёт затрат труда на подготовку описания.

Оценить затраты труда на подготовку описания задачи не возможно, т.к. это связано с творческим характером работы, вместо этого оценим затраты труда на изучение описания задачи с учётом уточнения описания и квалификации программиста определяются:

$$t_{П} = Q \cdot B / (75 \dots 85 \cdot K),$$

где

B - коэффициент увеличения затрат труда вследствие недостаточного описания задачи, уточнений и некоторой недоработки, $B = 1, 2 \dots 5$;

K - коэффициент квалификации разработчика, для работающих до 2 лет $K = 0, 8$;

В связи с тем, что при изучении описания данной задачи потребовалось много уточнений и доработок в описании коэффициент B принимаем равным 4.

Таким образом, получим:

$$t_H = 4000 \cdot 4 / (80 \cdot 0,8) = 250 (\text{чел} / \text{час}).$$

Расчёт затрат труда на разработку алгоритма.

Затраты труда на разработку алгоритма решения задачи:

$$t_A = Q / (60 \dots 75 \cdot K) = 4000 / (70 \cdot 0,8) = 71,43 (\text{чел} / \text{час}).$$

Расчёт затрат труда на разработку блок-схемы.

Затраты труда на разработку блок-схемы алгоритма решения задачи вычислим следующим образом:

$$t_B = Q / (60 \dots 75 \cdot K) = 4000 / (70 \cdot 0,8) = 71,43 (\text{чел} / \text{час}).$$

Расчёт затрат труда на составление программы.

Затраты труда на составление программы по готовой блок-схеме вычислим по формуле:

$$t_D = Q / (60 \dots 75 \cdot K) = 4000 / (70 \cdot 0,8) = 71,43 (\text{чел} / \text{час}).$$

Расчёт затрат труда на отладку программы.

Затраты труда на отладку программы на ЭВМ при комплексной отладке задачи:

$$t_{OT} = 1,5 \cdot t_{OT}^A,$$

где

t_{OT}^A - затраты труда на отладку программы на ЭВМ при автономной отладке одной задачи;

$$t_{OT}^A = Q / (40 \dots 50 \cdot K) = 4000 / (45 \cdot 0,8) = 111,11 (\text{чел} / \text{час}).$$

Отсюда

$$t_{OT} = 1,5 \cdot 111,11 = 166,67 (\text{чел} / \text{час}).$$

Расчёт затрат труда на подготовку документации.

Затраты труда на подготовку документации по задаче определяются:

$$t_D = t_{DP} + t_{DO},$$

где

t_{DP} - затраты труда на подготовку материалов в рукописи;

t_{DO} - затраты на редактирование, печать и оформление документации.

$$t_{DP} = Q / (150 \dots 200 \cdot K) = 4000 / (200 \cdot 0,8) = 25 (\text{чел} / \text{час});$$

$$t_{DO} = 0,75 \cdot t_{DP} = 0,75 \cdot 25 = 18,75 (\text{чел} / \text{час});$$

Отсюда

$$t_D = 18,75 + 25 = 43,75 (\text{чел} / \text{час}).$$

Итак, общую трудоёмкость программного продукта можем рассчитать:

$$t = 250 + 71,43 + 71,43 + 71,43 + 43,75 + 166,67 = 674,71(\text{чел}/\text{час}).$$

Расчёт средней зарплаты программиста.

Средняя зарплата программиста в современных рыночных условиях может варьироваться в широком диапазоне. Для примера расчёта возьмём среднюю часовую оплату труда, которая составляет $T_{\text{час}} = 20 \text{ руб}/\text{час}$, что составляет $3520 \text{ руб}/\text{мес.}$ при 8-ми часовом рабочем дне и 5-ти дневной рабочей неделе. Вы должны взять примерную зарплату на данный момент в городе.

Затраты на оплату труда программиста состоят из зарплаты программиста и отчислений на социальные нужды. Отчисления на социальные нужды включают в себя:

- пенсионный фонд (29%),
- медстрах (3,6%),
- соцстрах (5,4%),
- фонд занятости (1,5%),
- сбор на образование (1%).

Итого отчисления на социальные нужды составляют 40,5%. Отсюда затраты на оплату труда программиста составляют:

$$Z_{\text{СПП}}^{\text{ЗП}} = 674,71 \cdot 20 \cdot 1,405 = 18959,35 \text{ руб.}$$

Затраты на оплату машинного времени.

Затраты на оплату машинного времени при отладке программы определяются путём умножения фактического времени отладки программы на цену машино-часа арендного времени:

$$Z_{\text{ЭВМ}}^{\text{ЗП}} = C_{\text{час}} \cdot t_{\text{ЭВМ}},$$

где

$C_{\text{час}}$ - цена машино-часа арендного времени, руб/час;

$t_{\text{ЭВМ}}$ - фактическое время отладки программы на ЭВМ;

Расчёт фактического времени отладки.

Фактическое время отладки вычислим по формуле:

$$t_{\text{ЭВМ}} = t_{\text{П}} + t_{\text{ДО}} + t_{\text{ОТ}};$$

$$t_{\text{ЭВМ}} = 71,43 + 18,75 + 166,67 = 256,85 \text{ часа.}$$

Расчёт цены машино-часа.

Цену машино-часа найдём по формуле:

$$C_{\text{час}} = Z_{\text{ЭВМ}} / T_{\text{ЭВМ}},$$

где

$Z_{\text{ЭВМ}}$ - полные затраты на эксплуатацию ЭВМ в течении года;

$T_{\text{ЭВМ}}$ - действительный годовой фонд времени ЭВМ, час/год;

Расчёт годового фонда времени работы ПЭВМ IBM PC AT.

Общее количество дней в году - 365.

Число праздничных и выходных дней - 119.

Время простоя в профилактических работах определяется как еженедельная профилактика по 4 часа.

Итого годовой фонд рабочего времени ПЭВМ составляет:

$$T_{ЭВМ} = 8 \cdot (365 - 119) - 52 \cdot 4 = 1760 \text{ часа.}$$

Расчёт полных затрат на эксплуатацию ЭВМ.

Полные затраты на эксплуатацию ЭВМ можно определить по формуле:

$$Z_{ЭВМ} = (Z_{ЗП} + Z_{АМ} + Z_{ЭП} + Z_{ЕМ} + Z_{ТР} + Z_{ПР}),$$

где

$Z_{ЗП}$ - годовые издержки на заработную плату обслуживающего персонала, руб/год;

$Z_{АМ}$ - годовые издержки на амортизацию, руб/год;

$Z_{ЭП}$ - годовые издержки на электроэнергию, потребляемую ЭВМ, руб/год;

$Z_{ЕМ}$ - годовые издержки на вспомогательные материалы, руб/год;

$Z_{ТР}$ - затраты на текущий ремонт компьютера, руб/год;

$Z_{ПР}$ - годовые издержки на прочие и накладные расходы, руб/год;

Амортизационные отчисления.

Сумма годовых амортизационных отчислений определяется по формуле:

$$Z_{АМ} = C_{БАЛ} \cdot H_{АМ},$$

где

$C_{БАЛ}$ - балансовая стоимость компьютера, руб/шт.;

$H_{АМ}$ - норма амортизации, %.

Согласно постановления совета министров СССР от 22 октября 1990 года № 1072 «О единых нормах амортизационных отчислений на полное восстановление основных фондов народного хозяйства СССР» $H_{АМ} = 12,5\%$.

Балансовая стоимость ПЭВМ включает отпускную цену, расходы на транспортировку, монтаж оборудования и его наладку:

$$C_{БАЛ} = C_{РЫН} + Z_{УСТ},$$

где

$C_{РЫН}$ - рыночная стоимость компьютера, руб/шт.;

$Z_{УСТ}$ - затраты на доставку и установку компьютера, руб/шт.

Компьютер, на котором велась работа, был приобретен по цене $C_{РЫН} = 9000 \text{ руб.}$, (вставить свои данные) затраты на установку и наладку составили примерно 10% от стоимости компьютера:

$$Z_{УСТ} = 10\% \cdot C_{РЫН} = 0,1 \cdot 9000 = 900 \text{ руб.}$$

Отсюда

$$C_{БАЛ} = 9000 + 900 = 9900 \text{ руб / шт.}$$

а

$$Z_{АМ} = 9900 \cdot 0,125 = 1237,5 \text{ руб / год.}$$

Расчёт затрат на электроэнергию.

Стоимость электроэнергии, потребляемой за год, определяется по формуле:

$$Z_{ЭЛ} = P_{ЭЛ} \cdot T_{ЭВМ} \cdot C_{ЭЛ} \cdot A,$$

где

$T_{ЭВМ}$ - суммарная мощность ЭВМ;

$C_{ЭЛ}$ - стоимость $1 \text{ кВт} \cdot \text{ч}$ электроэнергии;

A - коэффициент интенсивного использования мощности машины.

Согласно техническому паспорту ЭВМ $P_{ЭВМ} = 0,22 \text{ кВт}$, стоимость $1 \text{ кВт} \cdot \text{ч}$ электроэнергии для предприятий $C_{ЭЛ} = 0,548 \text{ руб.}$, интенсивность использования машины $A = 0,98$.

Тогда расчётное значение затрат на электроэнергию:

$$Z_{ЭЛ} = 0,22 \cdot 1760 \cdot 0,548 \cdot 0,98 = 207,94 \text{ руб.}$$

Расчёт затрат на текущий ремонт.

Затраты на текущий и профилактический ремонт принимаются равными 5% от стоимости ЭВМ:

$$Z_{ТР} = 0,05 \cdot C_{ЭЛ} = 0,05 \cdot 9900 = 495 \text{ руб.}$$

Расчёт затрат на вспомогательные материалы.

Затраты на материалы, необходимые для обеспечения нормальной работы ПЭВМ составляют около 1% от стоимости ЭВМ:

$$Z_{ЕМ} = 0,01 \cdot 9900 = 99 \text{ руб.}$$

Прочие затраты по эксплуатации ПЭВМ.

Прочие косвенные затраты, связанные с эксплуатацией ПЭВМ, состоят из амортизационных отчислений на здания, стоимости услуг сторонних организаций и составляют 5% от стоимости ЭВМ:

$$Z_{ПР} = 0,05 \cdot 9900 = 495 \text{ руб.}$$

Годовые издержки на заработную плату обслуживающего персонала.

Издержки на заработную плату обслуживающего персонала складываются из основной заработной платы, дополнительной и отчислений на заработную плату:

$$Z_{ЗП} = Z_{ЗП}^{ОСН} + Z_{ЗП}^{ДОП} + Z_{ЗП}^{ОТЧ}$$

Сумма основной заработной платы определяется исходя из общей численности работающих в штате:

$$Z_{ЗП}^{ОСН} = 12 \cdot \sum Z_{ОКЛ}^i,$$

где

$Z_{ОКЛ}^i$ - тарифная ставка i - го работника в месяц, руб.;

В штат обслуживающего персонала должны входить инженер-электронщик с месячным окладом 1000 руб. и электрослесарь с окладом 500 руб.

Тогда, учитывая, что данный персонал обслуживает 10 машин, имеем издержки на основную заработную плату обслуживающего персонала составят:

$$Z_{ЗП}^{ОСН} = 12 \cdot (1000 + 500) / 10 = 1800 \text{ руб.}$$

Сумма дополнительной заработной платы составляет 60% от основной заработной платы:

$$Z_{ЗП}^{ДОП} = 0,6 \cdot 1800 = 1080 \text{ руб.}$$

Сумма отчислений на социальные нужды составляет 40.5% от суммы дополнительной и основной заработных плат:

$$Z_{ЗП}^{ОГЧ} = 0,405 \cdot (1800 + 1080) = 1166,4 \text{ руб.}$$

Тогда годовые издержки на заработную плату обслуживающего персонала составят:

$$Z_{ЗП} = 1800 + 1080 + 1166,4 = 4046,4 \text{ руб.}$$

Полные затраты на эксплуатацию ЭВМ в течение года составят:

$$Z_{ЭВМ} = 4046,4 + 1237,5 + 207,94 + 99 + 495 + 495 = 6580,84 \text{ руб.}$$

Тогда цена машино-часа арендуемого времени составит:

$$C_{\text{час}} = 6580,84 / 1760 = 3,74 \text{ руб.}$$

А затраты на оплату машинного времени составят:

$$Z_{\text{СПИ}}^{\text{МЕ}} = 3,74 \cdot 256,85 = 960,62 \text{ руб.}$$

Расчёт общих расходов.

Общие расходы это расходы на освещение, отопление, коммунальные услуги и т.п. Они принимаются равными одной трети основной зарплате разработчика программы, т.е. 2249,03 руб.

Тогда затраты на создание программного продукта составят:

$$Z_{\text{СПИ}} = 18959,35 + 960,62 + 2249,03 = 22169 \text{ руб.}$$

Можно сделать следующие выводы.

Например: В результате расчета затраты на создание данного программного продукта составили 22169 руб. Данную цифру сложно оценить, так как имеющиеся на рынке подобные продукты слишком специфичны и количество их очень мало. Для снижения затрат можно предложить следующие решения. Как видно, три четверти расходов представляют из себя затраты на оплату труда программиста. Снизить эти затраты можно путем повышения эффективности труда программиста за счет использования более современных ЭВМ для работы, повышения удобства рабочего места и прочих факторов. Также при более бережном и аккуратном отношении к компьютерной технике возможно снизить затраты на ремонт, а так же уменьшить привлечение дополнительного персонала на обслуживание компьютеров.

Заключение.

В этой части студент формулирует основные выводы по теме и рекомендации относительно возможностей практического применения материалов работы. Следует также указать на имеющиеся проблемы и возможные перспективы их решения.

Список использованной литературы.

Перечисляется литература, использованная при выполнении курсовой работы, в том числе и ГОСТы.

Приложения.

В приложение нужно поместить все экранные формы программы, а так же возможно и листинга программы.

3. ПРАВИЛА ОФОРМЛЕНИЯ КУРСОВОЙ РАБОТЫ

По объему курсовые проекты должны быть не менее 30 страниц печатного текста.

Курсовая работа должна быть сдана в машинописном виде (на компьютере). Текст должен быть напечатан через 1,5 интервала шрифтом Times New Roman стандартного размера (14 шрифт) с соблюдением установленных размеров отступа от края листа:

- левое поле - 30 мм;
- правое поле - 10 мм;
- верхнее и нижнее поля - по 20 мм.

Титульный лист и содержание не нумеруются, но включены в общую нумерацию.

Содержание проекта, введение, название параграфов, заключение, список литературы, приложение начинается с новой страницы. Каждый самостоятельный раздел нумеруется и озаглавляется аналогично тому, как это сделано в "Содержании". Названия параграфов печатаются симметрично тексту (по центру), 16 шрифтом, жирным. Названия пунктов печатаются с красной строки строчными буквами (кроме первой), 14 шрифтом, жирным.

Расстояние между заголовком параграфа и основным текстом должно быть равно трем (одинарным) межстрочным интервалам.

Рисунки (схемы, графики) должны быть выполнены разборчиво и иметь поясняющие надписи, которые расположены под рисунком (примеры смотрите выше). В текст должна быть ссылка на каждый рисунок (Например, данные... см. на рис.1).

При оформлении таблицы слово «Таблица» пишется в левом верхнем углу (без кавычек) и ставится е порядковый номер. Нумерация может быть сквозной через всю работу или по параграфам. Таблица должна содержать название (табл. 3 - Таблица входных данных и их файлов).

При упоминании о таблице в тексте делается ссылка (табл. 1.1).

Список литературы должен содержать перечень всех источников (не менее 5), использованных при написании курсовой работы. Источники располагаются в алфавитном порядке (по первой букве первого слова. В авторских источниках первым словом считается фамилия автора).

- все источники в перечне нумеруются. Для каждого источника указываются: фамилия и инициалы автора (авторов); полное название книги, статьи; название журнала или сборника статей (для статей);

- название города (Москва и Санкт-Петербург - сокращенно, соответственно М. и СПб., остальные полностью);

- название издательства (если имеется в выходных данных) - для книг;

- год издания (для статей - N журнала и год).

Например:

1. Стуков, С.А. Экономический анализ и рынок / С.А. Стуков. // Бухгалтерский учет. -1991.- №3.- С. 106-111.

2. Ткач, В.И. Международная система учета и отчетности / В.И. Ткач. - М.: Финансы и статистика, 1992. – 528с.

3. Управление по результатам. Пер. с финск. / Под ред. А.Я. Лейтмана. - М.: Прогресс, 1993. – 157с.

Примеры описаний электронных ресурсов:

- *Библиографические записи Интернет-ресурсов.*

Исследовано в России [Электронный ресурс]: многопредмет. науч. журн. / Моск. физ.- техн.ин-т. – Электрон. журн. – Долгопрудный: МФТИ, 1998. Режим доступа к журн.:<http://zhurnal.mipt.rssi.ru>.

Российская государственная библиотека [Электронный ресурс] / Центр информ. технологий РГБ; ред. Власенко Т.В.; Web-мастер Козлова Н.В. – Электрон. дан. – М.: Рос. гос. б-ка, 1997. –Режим доступа: <http://www.rsl.ru>, свободный. – Загл. с экрана. – Яз. рус., англ.

- *Библиографические записи сборников электронных ресурсов без общего заглавия:*

Английский для бизнесменов [Электронный ресурс]; Английский технический;

Английский для чтения газет и журналов: [к сб. в целом]: курс изучения иностр. яз. Intell. –Электрон. дан. и прогр. – М.: сор. Квант, 1994–1997. – 1 электрон., опт. диск (CD-ROM): зв., цв. –(25 кадр).

- *Библиографические записи электронных ресурсов, состоящие из нескольких отдельных частей (выпусков):*

Большая автомобильная энциклопедия [Электронный ресурс]. – Электрон. дан. – М.: Xelana Media Group, 1998.

Помимо основного текста курсовой проект может содержать приложения. В качестве приложений целесообразно оформлять код программного продукта и основные экраны интерфейса.

В этом случае в основном тексте в необходимых местах должны содержаться ссылки: «см. приложение». Приложение располагается непосредственно за списком литературы, на первом листе приложенных документов в верхнем правом углу пишется слово "ПРИЛОЖЕНИЕ" (а затем его порядковый номер) то же пишется в последней строке содержания.

Каждое приложение должно иметь заголовок, который выравнивается по центру страницы. Приложения располагаются в порядке ссылки на них в основном тексте.

ЛИТЕРАТУРА

1. Вендров А.М. Практикум по проектированию программного обеспечения экономических информационных систем. – М.: Финансы и статистика, 2004.
2. Иванова Г.С. Технология программирования. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2003.
3. Орлов С.А., Цилькер Б.Я. Технологии разработки программного обеспечения. – СПб.: Питер, 2012.

www.rugost.com

<http://khpi-iip.mipk.kharkiv.edu/library/case/leon/index.html>

(Леоненков.

Самоучитель UML)