1. **Курсовое проектирование** 
   1. **Общие требования**

Курсовой проект должен включать оттестированное программное средство и пояснительную записку.

Пояснительная записка проекта должна иметь следующую структуру:

- титульный лист установленного образца (Приложение А);

- техническое задание (Приложение Б);

- содержание курсового проекта (Приложение В).

В графической части курсового проекта могут быть представлены следующие результаты:

- функциональная структура ПС, показывающая функциональное назначение всего ПС и его отдельных частей;

- модульная (иерархическая) структура ПС, фиксирующая результаты проектирования ПС;

- диаграммы наследования, зависимостей, классов и структур классов ПС, фиксирующие результаты объектно-ориентированного проектирования ПС;

- схемы алгоритмов, иллюстрирующих основные методы и алгоритмы, реализованные в ПС;

- результаты работы ПО, показывающие наиболее типичные результаты в форме графиков, таблиц, примеров выходной документации и т.п.

**2.2 Общие требования к разработке ПС**

Разработка ПС является определяющим элементом курсового проектирования и может вестись с использованием какого-либо подхода проектирования например, водопадной модели разработки ПС (см. пункт 1.2).

Можно придерживаться следующих этапов жизненного цикла ПС см. рисунок 2.1.

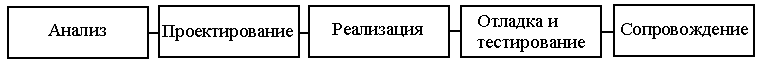


Рисунок 2.1 - Этапы жизненного цикла ПС

Целью этапа анализа является описание задачи, которое должно быть полным, последовательным, доступным для чтения и обзора различными заинтересованными сторонами, позволяющим производить сравнение с реальными условиями.

В ходе этого этапа решаются задачи:

- уточнение требований, приведенных в задании на проектирование;

- разработка спецификаций на ПС.

Итогом выполнения этого этапа являются эксплуатационные и функциональные спецификации, содержащие конкретное описание ПС.

Эксплуатационные спецификации должны содержать сведения о быстродействии ПО, затратах памяти, требуемых технических средствах, надежности и т.д. Функциональные спецификации определяют функции, которые должно выполнять ПС. Спецификации должны быть полными, точными и ясными.

Цель этапа проектирования - иерархическое разбиение сложной задачи создания ПО на подзадачи меньшей сложности.

На этапе проектирования решаются следующие задачи:

- формирование структуры ПС и разработка алгоритмов, задаваемых спецификациями;

- определение состава модулей с разделением их на иерархические уровни;

- выбор структуры информации в базе данных;

- фиксация межмодульных интерфейсов.

Результатом работы на этом этапе являются спецификации на отдельные модули, дальнейшая декомпозиция которых нецелесообразна.

Этап реализации или программирования включает в себя непосредственное кодирование текстов программ на выбранном алгоритмическом языке программирования. Цель этого этапа - получение текстов программ.

Цель этапа тестирования и отладки - выявление в ПС ошибок, проверка работоспособности ПС, его соответствие спецификациям.

В ходе этого этапа решаются следующие задачи:

- подготовка данных для отладки;

- планирование отладки;

- испытание ПО.

Результатом работы должно являться оттестированное и отлаженное ПС.

На этапе сопровождения возможно расширение функциональных возможностей ПС, уточнение существующих, а также устранение ошибок. В курсовом проекте, как правило, выполняются четыре этапа проектирования.

Примерные временные соотношения между отдельными видами работ представлены в таблице 2.1.

Разработка ПС должна начинаться с тщательного изучения задания на курсовое проектирование.

Этапы анализа и проектирования должны быть формализованы с помощью одного из рекомендуемых средств:

- аппарат формальных спецификаций;

- методы структурного анализа;

- методы объектно-ориентированного анализа;

- методы объектно-ориентированного проектирования.

Таблица 2.1 - Распределение времени по этапам разработки ПС

(в % к общему времени разработки)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Этапы разработки ПС | | | | | Всего |
| Виды работ | Анализ | Проектирование | Программирование | Отладка и тестирование |
| Анализ требований и разработка спецификаций | 13 |  |  |  | 13 |
| Подготовка данных для отладки |  | 2 | 2 | 4 | 8 |
| Планирование отладки | 2 |  | 2 | 4 | 8 |
| Проектирование |  | 13 |  |  | 13 |
| Тестирование | 5 | 5 | 4 | 11 | 25 |
| Программирование |  |  | 8 |  | 8 |
| Испытание ПС |  |  |  | 17 | 17 |
| Документирование |  |  | 4 | 4 | 8 |
| Всего | 20 | 20 | 20 | 40 | 100 |

**2.3 Организация графического интерфейса**

Разрабатываемое в курсовом проектировании ПС должно быть обязательно оснащено графическим пользовательским интерфейсом, что соответствует современным тенденциям и требованиям рынка на ПО.

Под графическим пользовательским интерфейсом (GUI - Graphical User Interface) понимается некоторая система (среда), служащая для организации диалога ПС с пользователем на основе графического многооконного представления данных. В среде GUI организацию всего взаимодействия с пользователем берет на себя именно сама среда, оставляя ПС делать только свою работу.

К общим принципам, лежащим в основе графического пользовательского интерфейса, относятся:

- графический режим работы;

- представление ряда объектов пиктограммами;

- многооконность;

- использование указывающего устройства - мыши;

- адекватность изображения на экране изображаемому объекту (принцип WYSIWIG - What You See Is What You Get);

- наглядность;

- стандартизация всех основных действий и элементов (все программы для данной графической среды выглядят и ведут себя совершенно одинаково, используют одинаковые принципы функционирования);

- наличие большого числа стандартных элементов (кнопок, полей редактирования, переключателей и т.д.), которые могут использоваться при конструировании ПС, делая их похожими в обращении и облегчая процесс их написания.

В основе современного графического пользовательского интерфейса лежат две основные концепции.

Первой из них является понятие программы, управляемой данными.

Как правило, эта концепция практически реализуется через механизм сообщений. Внешние устройства (клавиатура, мышь, таймер) посылают сообщения модулям программы о наступлении тех или иных событий (например, при нажатии клавиши или передвижении мыши). Поступающие сообщения попадают в очередь сообщений, откуда извлекаются прикладной программой.

Таким образом, программа не должна все время опрашивать мышь, клавиатуру и другие устройства в ожидании, не произошло ли чего-нибудь, заслуживающего внимания. Когда событие произойдет, программа получит извещение об этом с тем, чтобы надлежащим образом его обработать. Поэтому программы для таких сред обычно представляют собой цикл обработки сообщений: извлечь очередное сообщение, обработать его, если оно интересно, либо передать стандартному обработчику сообщений, обычно входящему в систему и представляющему собой стандартные действия системы в ответ на то или иное событие.

Сообщения могут посылаться не только устройствами, но и отдельными частями программы (в частности, возможна посылка сообщения себе). Так один модуль может послать сообщение другому модулю, или меню посылает сообщение о выборе определенного пункта. При этом существует также способ прямой посылки сообщения, минуя очередь, когда непосредственно вызывается обработчик сообщений адресата.

Второй основополагающей концепцией является понятие окна как объекта. Окно - это не просто прямоугольная область на экране, это и программа (процедура, функция), способная выполнить различные действия, присущие окну. Одним из основных таких действий является реагирование на поступающие сообщения и посылка сообщений другим объектам.

Одной из основных функций окна является перерисовка содержания окна. Любое окно должно уметь при получении соответствующего запроса перерисовать себя (или свою часть) на экране. Перерисовка может реализовываться или как реакция на специальное сообщение, или как виртуальная функция (при использовании объектно-ориентированных языков). В состав любой GUI обязательно входит достаточно мощный графический модуль, обеспечивающий выполнение всех основных графических операций и поддерживающий отсечение изображения по заданной (в том числе и довольно сложной) области отсечения. За счет этого реализуется возможность перерисовки фрагмента окна - устанавливается область отсечения, совпадающая с требуемым фрагментом, а затем выполняется запрос на перерисовку. При отработке запроса на перерисовку окна можно определить размер текущей области и не пытаться рисовать то, что заведомо будет отсечено.

Среди окон вводятся отношения принадлежности и следования, т.е. любое окно может иметь окно-родителя, которому оно принадлежит, и, следовательно, задается во внутренних координатах родительского окна, отсекается в размерах родительским окном и уничтожается при уничтожении родительского окна. Любое окно может иметь и принадлежащие ему окна (подокна), причем последние некоторым образом упорядочиваются. Тем самым окна могут образовывать древовидные структуры подчинения.

Родительское окно и принадлежащие ему подокна могут обмениваться сообщениями друг с другом. Эти сообщения обычно разделяются на два класса - запрос на выполнение окном некоторого действия и сообщение, оповещающее окно о том, что в другом окне (обычно подокне) произошли некоторые изменения.

Любая подобная система должна предоставлять для работы некоторый стандартный набор типов окон, из которых пользователь может строить свои программы.

В состав окна могут входить другие окна и действовать при этом как единое целое. Например, в состав окна-списка может входить скроллер.

Среди окон обычно выделяются окна, предназначенные для ведения диалога с пользователем, ввода данных и т.п. Обычно в их основе лежит стандартное окно с большим набором подокон, играющих роль управляющих элементов. Как правило, диалоговое окно (или процедура, ведущая диалог) снабжается специальной функцией для координации работы управляющих элементов. Например, диалог для выбора файла.

Кроме стандартных окон пользователь может создавать свои собственные типы окон, либо добавляя какие-то новые свойства, либо переопределяя часть старых и наследующих все остальное.

При работе с клавиатурой важную роль играет понятие фокуса ввода. Фокус ввода - это то окно, которому поступают все сообщения от клавиатуры. Существует несколько способов перемещения фокуса ввода:

- при нажатии кнопки мыши фокус передается тому окну, над которым это произошло;

- окна диалога обычно переключают фокус между управляющими элементами диалога при нажатии определенных клавиш (стандартно это Tab и Shift-Tab);

- посредством явного вызова функции установки фокуса ввода.

Окну, теряющему фокус ввода, обычно посылается уведомление об этом, и оно может предотвратить переход фокуса от себя. Окну, получающему фокус, передается сообщение о том, что оно получило фокус ввода.

**2.4 Требования к программной документации**

При разработке документации необходимо придерживаться требований СТП101-00 и стандартов ЕСПД. Стандарт предприятия разработан с учетом всех действующих ГОСТов на документацию. При работе над курсовым проектом необходимо изучить следующие разделы:

**-** оформление текста;

**-** титульный лист;

**-** аннотация;

**-** содержание;

**-** введение;

**-** основная часть;

**-** изложение текста;

**-** заключение;

**-** оформление иллюстраций;

**-** построение таблиц;

**-** список использованных источников;

**-** приложения;

**-** графическая часть;

**-** схемы;

**-** кодирование документов;

**-** общие требования к оформлению курсовых проектов (работ).

В стандарте ЕСПД необходимо обратить внимание на следующие разделы:

##### − виды программных документов ГОСТ 19.101-77;

*−*  стадии разработки ГОСТ 19.102-77;

*−*  техническое задание. Требования к содержанию и оформлению ГОСТ 19.201-78;

#### − схемы алгоритмов, программ данных и систем ГОСТ 19.701-90;

*−*  текст программы ГОСТ 19.401-78;

*−*  описание программы ГОСТ 19.402 -78;

*−*  программа и методика испытаний ГОСТ 19.301-79

*−*  пояснительная записка ГОСТ 19.404-79;

*−*  описание применения ГОСТ 19.502-78;

*−*  руководство системному программисту ГОСТ 19.503-79;

*−*  руководство программиста ГОСТ 19.504-79;

− руководство оператору ГОСТ 19.505-79.

* 1. **Содержание курсового проекта**

Курсовой проект должен иметь следующую структуру и состоять из разделов.

Аннотация

Введение

1 Общие сведения о программном средстве

1.1 Основное функциональное назначение программного средства

1.2 Полное наименование программного средства

1.3 Условное обозначение программного средства

1.4 Разработчики программного средства

2 Техническое задание

2.1 Основание для разработки

2.2 Назначение разработки

2.3 Требования к программному средству

2.4 Требования к программной документации

2.5 Требования к эргономике и технической эстетике

2.6 Стадии и этапы разработки

2.7 Порядок контроля и приемки

3 Пояснительная записка

3.1 Декомпозиция поставленной задачи

3.2 Общая архитектура программного средства

3.3 Реализация функционального назначения программного средства

3.4 Разработка алгоритма решения задачи

3.4.1 Детальная разработка алгоритмов отдельных подзадач

3.5 Структурная организация данных

3.6 Разработка интерфейса ПС

3.7 Описание структуры выходной информации

4 Руководство системного программиста

4.1 Общие сведения о программном средстве

4.2 Структура программного средства

4.3 Установка программного средства

4.4 Проверка программного средства

4.5 Сообщения системному программисту

5 Руководство программиста

5.1 Назначение и условия применения программного средства

5.2 Характеристика программного средства

5.3 Работа с программным средством

5.4 Входные и выходные данные

5.5 Сообщения программисту

6 Руководство пользователя

6.1 Назначение программного средства

6.2 Условия выполнения программного средства

6.3 Эксплуатация программного средства

6.4 Сообщения пользователю

7 Заключение

8 Список использованной литературы

9 Приложения

В пояснительной записке должна содержаться следующая информация.

*Аннотация*. В аннотации указывается тема работы, ее основное назначение, функции, выполняемые программным средством (ПС), объем пояснительной записки, перечень графической части работы, перечень приложений и др. (Приложение Д).

*Введение.* Во введении кратко описывается состояние вопроса, формулируются цель и задачи проекта, а также актуальность и обоснованность его решения.

*Общие сведения о ПС*. В этом разделе описывается назначение ПС, т.е. какую задачу оно решает и где может найти применение. Указывается полное наименование ПС с расшифровкой слов, которые представлены аббревиатурой. Указывается условное обозначение ПС, которое дали ему авторы разработки, например пакет прикладных программ (ППП) «OMEGA» или программный комплекс (ПК) «НАДЕЖДА» и др. Представляются авторы разработки: Фамилия, имя, отчество, основное место учебы (учеба и работа), а также та часть работы, которая выполнялась в данном курсом проекте.

Далее необходимо сформулировать техническое задание, на данную работу состоящее из нескольких разделов.

*Основание для разработки.* В этом разделе должны быть указаны документы, на основании которых ведется разработка, организация, утвердившая этот документ, дата утверждения, наименование или условное обозначение темы разработки.

*Назначение разработки*. Здесь должно быть указано функциональное и эксплутационное назначение ПС.

*Требования к ПС.* В этом разделе должна содержаться следующая информация:

* требования к функциональным характеристикам (состав выполняемых функций, организация входных и выходных данных и т.п.);
* требования к структуре ПС (возможность модернизации, увеличению функциональных возможностей);
* требования к надежности (надежное функционирование, контроль входной и выходной информации и т.п.);
* условия эксплуатации (температура, влажность, количество и квалификация персонала);
* требования к составу и параметрам технических средств (состав технических средств с указанием их технических характеристик);
* требования к информационной и программной совместимости (требования к информационном структурам, языкам программирования);
* требования к транспортировке и хранению;
* специальные требования.

*Требования к программной документации.* Указываются стандарты, на основании которых должна выполняться документация на разрабатываемое ПС и состав документации.

*Требования к эргономике и технической эстетике.* Формулируются требования к организации пользовательского интерфейса. Общий дизайн ПС. Представление выходной информации (иллюстрации, графическое представление информации, анимация и т.д.).

*Стадии и этапы разработки.* Указываются сроки разработки, т. е. временной период, в течение которого данная работа должна быть выполнена. А также, на какие этапы данная работа разбита (Приложение Е).

*Порядок контроля и приемки*. Поясняются общие требования к аттестации, данной работы.

В техническом задании допускается включать приложения. Допускается уточнять содержание разделов, вводить новые разделы или объединять отдельные из них.

В пояснительной записке должна быть представлена следующая информация.

*Декомпозиция поставленной задачи.* Поставленная задача разбивается на ряд подзадач, которые необходимо решить при решении общей целевой задачи.

*Общая архитектура ПС.* После разбиения задачи на подзадачи, которые могут быть реализованы в виде отдельных модулей, процедур, функций, необходимо разработать архитектуру ПС с описанием взаимодействия отдельных элементов ПС. А также показать, как обрабатывается и преобразуется информация при переходе от одного модуля к другому. Описание должно иллюстрироваться схемами.

*Реализация функционального назначения ПС.* Поскольку разные функции ПС реализованы отдельными структурными единицами, необходимо описать, какая информация необходима для выполнения конкретной функции и какие результаты получаются в результате ее выполнения. В описании необходимо привести функциональную схему работы ПС.

*Разработка алгоритма решения задачи.* На основе анализа всех функций, которые должно выполнять проектируемое ПС, необходимо разработать и описать алгоритм решения задачи. В зависимости от выполнения или невыполнения тех или иных условий показать порядок и последовательность решения задачи. Логическую структуру ПС показать на укрупненной схеме алгоритма.

*Детальная разработка алгоритмов отдельных подзадач.* В этом разделе должна быть представлена логическая структура модулей и процедур, составляющих данный ПК. Для каждой программной единицы необходимо представить входные данные, функции, которые выполняются и результаты работы модуля. Для модулей, которые имеют сложную логическую структуру, описание может быть иллюстрировано схемой алгоритма.

*Структурная организация данных.* В этом разделе нужно представить данные используемые в ПС (файлы, массивы, и т.д.) их структуру, типы и т.д. Если данные имеют сложную структуру, то описание необходимо пояснять графическими схемами.

*Разработка интерфейса ПС.* В этом разделе необходимо описать структуру разработанного интерфейса. Обосновать его эргономичность.

*Описание структуры выходной информации.* Здесь необходимо описать, какую информацию можно получить в результате эксплуатации ПС.

Разделы пояснительной записки могут сопровождаться схемами, для повышения наглядности процесса проектирования ПС (см. Приложение Ж).

В разделе «Руководство системного программиста» нужно представить информацию, необходимую для того, чтобы системный программист мог быстро и правильно установить ПС, проверить его и убедиться в том, что ПС находится в рабочем состоянии.

*Общие сведения о программе.* Здесь дается краткая информация о ПС, его назначении, основные функции, которое оно выполняет, применение ПС.

*Структура программы.* Указывается из скольких программных единиц и файлов состоит ПС, т.е. описывается полная комплектность с ссылкой на пункт 3.2 раздела 2.5.

*Установка программы.* Описываются все действия, необходимые для установки ПС на ПЭВМ. Также объем, занимаемый ПС на жестком магнитном диске, минимальный объем оперативной памяти, необходимый для эксплуатации ПС. Технические характеристики оборудования, необходимого для работы ПС.

*Проверка программы.* После, того как ПС установлено, необходимо убедиться, что оно установлено верно, т. е. установлены все программные компоненты путем инсталляции или простым копированием, ПС находится в рабочем состоянии и может выполнять свое функциональное назначение.

*Сообщения системному программисту.* В процессе инсталляции или работы ПС возможны сообщения системному программисту для пояснения или подтверждения правильности его действий.

В разделе «Руководство программиста» нужно представить информацию необходимую для того, чтобы программист мог разобраться в структуре и логике ПС, с организацией входных и выходных данных для внесения изменений, расширения функциональных возможностей и исправления ошибок.

*Назначение и условия применения ПС.* Поясняется основное функциональное назначение ПС и возможность его применения. Минимальный состав технических средств с указанием их технических характеристик для нормальной эксплуатации ПС.

*Характеристика ПС*. Дается краткая характеристика ПС: режимы работы, необходимый объем памяти для эксплуатации, средства контроля и др.

*Работа с ПС.* Здесь поясняется обращение к программе, способы передачи управления, вызов программы и др.

*Входные и выходные данные.* В этом разделе описывается организация входных и выходных данных.

*Сообщения программисту.* Если в ПС при проектировании предусмотрена возможность расширения или изменения некоторых функций они должны быть описаны для программиста, который будет заниматься сопровождением ПС.

В разделе «Руководство пользователя» нужно представить информацию необходимою для эксплуатации ПС. Должна быть описана последовательность выполнения работы, средства защиты, разработанные в данном ПС, реакцию ПС на неверные действия пользователя.

*Назначение ПС.* В этом разделе дается краткое описание основного назначения ПС.

*Условия выполнения программы*. Описываются условия, при которых данное ПС может нормально функционировать (минимальный или максимальный состав аппаратуры и ПС).

*Выполнение ПС*. Описываются все действия пользователя для выполнения ПС своего функционального назначения, т.е. как работать с ПС.

*Сообщения пользователю*. При эксплуатации программного средства, могут быть предусмотрены различные сообщения, которые поясняют действия пользователя, предотвращают ошибки и дают возможность исправить допущенные ошибки.

*Заключение.* В заключении приводятся основные выводы и перспективы дальнейшего развития представленного ПС.

В разделе «Список использованных источников» дается перечень всей литературы, которая была использована при разработке ПС и оформлении документации на него. Список использованных источников формируется в том порядке, в котором были ссылки на использованную литературу, с указанием издательства, года издания и количества листов в книге согласно СТП101-00 (Приложение З).

В «Приложении» должен быть текст ПС, контрольные и тестовые примеры, результаты работы ПС, также могут быть документы, на основании которых ведется разработка.

**2.6 Задания для курсового проектирования**

2.6.1Задачи вычислительного типа

Задание: разработать пакет прикладных программ (ППП) по заданной теме, реализовать указанное преподавателем количество методов, с возможностью подключения еще нескольких методов. Оценить точность каждого метода. Провести полное тестирование. Привести контрольные примеры. Оформить удобный пользовательский интерфейс (окна, возможность удаления, корректировки, добавления информации, проверка на допустимость входных данных, выходные формы, помощь и др.).

# Вариант № 6

Разработать ППП «Численное дифференцирование»

различными методами:

1. на основе интерполяционной формулы Лагранжа;
2. на основе интерполяционной формулы Ньютона;
3. по безразностным формулам численного дифференцирования.

2.6.2 Задачи по обработке и анализу информации

# Вариант №6

Тема: «Автоматизация работ культурно-массовых

мероприятий»

Разработать ПС, позволяющее автоматизировать работу фирмы «Лира», занимающейся культурно-массовыми мероприятиями. ПС должно обрабатывать следующую информацию: вид культурного заведения (театр, музей, цирк, кинотеатр, дворец культуры и т.д.); для учреждения каждого типа: вид и название проводимого мероприятия (спектакль, кинофильм, выставка, экскурсия и др.), дата и время проведения мероприятия, количество имеющихся и проданных билетов, цена билетов и др. А также анкетные данные для посещающих: возраст, образование, профессия и т.д.

Обеспечивать составление программы культурного отдыха по запросу клиента (на определенный день, неделю, месяц); представлять справочную информацию об имеющихся в продаже билетах на определенный спектакль, коллектив, кинофильм, и т.д., позволять бронировать и покупать билеты.

Осуществлять анализ посещаемости культурных заведений:

* определять самые посещаемые заведения, мероприятия и коллективы;
* исследовать динамику роста или спада посещаемости заведений различными слоями населения по месяцам, годам;
* выявлять корреляционную зависимость между ценой билета и количеством проданных билетов; образованием и количеством посещений культурных заведений в месяц, год и т.д.

Представлять графическую интерпретацию полученных результатов.