

## Тема 1.1. Разработка спецификаций системного программного обеспечения

### Лабораторная работа № 1. Разработка спецификаций структурных единиц

#### 1. Цель работы

В соответствии с рабочей программой по дисциплине «Системное программирование» в результате выполнения заданий по лабораторным работам студент должен:

уметь:

- осуществлять разработку кода программного модуля на современных языках программирования;
- оформлять документацию на программные средства;

знать:

- основные принципы технологии структурного и объектно-ориентированного программирования;
- методы и средства разработки технической документации;
- основные принципы отладки и тестирования программных продуктов.

Таким образом, студент во время проведения занятия и самостоятельной работы по теме занятия должен:

- приобрести навыки разработки спецификаций системного программного обеспечения;
- закрепить теоретические знания о формировании требований к программному обеспечению.

#### 2. Краткие сведения из теории

Основными вопросами, которые должны рассматривать составитель (-ли) спецификаций, являются следующие:

а) *Функциональные возможности.* Каковы предполагаемые функции программного обеспечения?

б) *Внешние интерфейсы.* Как программное обеспечение взаимодействует с пользователями, аппаратными средствами системы, другими аппаратными средствами и другим программным обеспечением?

в) *Рабочие характеристики.* Каково быстродействие, доступность, время отклика, время восстановления различных функций программного обеспечения и т.д.?

г) *Атрибуты.* Каковы мобильность, правильность, удобство сопровождения, защищенность программного обеспечения и другие критерии?

д) *Проектные ограничения, налагаемые на реализацию изделия.* Предъявляются ли требования к соответствию каким-либо действующим стандартам, к языку реализации, к политикам целостности баз данных, к задействуемым ресурсам, к операционной среде(-ам) и т.д.?

Составителю(-ям) спецификаций следует избегать размещения в спецификациях архитектурных решений или требований к организации и выполнению проекта.

Спецификация является проверяемой, если и только, если каждое требование, изложенное в ней, может быть проверено. Требования являются проверяемым, если и только, если существует некий конечный экономически-эффективный (рентабельный) процесс, используя который пользователь или машина могут убедиться, что программное изделие удовлетворяет этому требованию. В целом, любое неоднозначное требование не проверяемо.

Непроверяемые требования включают формулировки типа «работает хорошо», «хороший пользовательский интерфейс» и «обычно должно происходить». Эти требования не могут быть проверены, так как невозможно определить термины «хороший», «хорошо» или «обычно». Утверждение о том, что «программа никогда не должна заикливаться» является непроверяемым, так как тестирование этого свойства теоретически невозможно.

Примером проверяемого утверждения является следующее:

*Программа должна выдавать результат в течение 20 секунд после заданного события в 60% случаев; и должна выдавать результат в течение 30 секунд после заданного события в 100% случаев.*

Это утверждение может быть проверено, так как в нём используются конкретные термины и измеримые величины.

Если нельзя изобрести метод, чтобы определить, отвечает ли программное обеспечение определённому требованию, то это требование следует исключить или пересмотреть.

Спецификация должна указывать, какие функции должны выполняться с какими данными для получения каких результатов в каком месте и для кого. Спецификация должна фокусироваться на предоставляемых услугах. Спецификация обычно не должна указывать элементы архитектуры и внутренней структуры программы, такие как:

- а) Разбиение программного обеспечения на модули;
- б) Присваивание функций модулям;
- в) Описание потока данных или управления между модулями;
- г) Выбор структур данных.

#### **Рекомендуемое содержание спецификации:**

- 1. Введение
  - 1.1 Назначение
  - 1.2 Область действия
  - 1.3 Определения, акронимы и сокращения
  - 1.4 Публикации
  - 1.5 Краткий обзор
- 2. Полное описание
  - 2.1 Контекст изделия
  - 2.2 Функции изделия
  - 2.3 Характеристики пользователя
  - 2.4 Ограничения
  - 2.5 Допущения и зависимости
- 3. Специфические требования
- Приложения
- Алфавитный указатель

#### **Назначение (Подраздел 1.1 Спецификации)**

Этот подраздел должен:

- а) Обрисовать назначение спецификации;
- б) Указать аудиторию, для которой предназначена спецификация.

#### **Область действия (Подраздел 1.2 Спецификации)**

Этот подраздел должен:

- а) Задавать название создаваемому программному изделию (-ям) (например, Host DMBS (Главная система управления базой данных), Генератор отчетов и т.д.);
- б) Объяснять, что программное изделие будет и, в случае необходимости, не будет делать;
- в) Описывать применение задаваемого программного обеспечения, включая связанные с ним выгоды, цели и задачи;
- г) Согласовываться с аналогичными формулировками в спецификациях более высокого уровня (например, со спецификацией системных требований), если они существуют.

#### **Определения, акронимы и сокращения (Подраздел 1.3 Спецификации)**

В этом подразделе должны быть представлены определения всех терминов, акронимов и сокращений, необходимые для правильной интерпретации спецификации. Эта информация может быть обеспечена ссылками на одно или более приложений в спецификации или ссылкой на другие документы.

**Публикации (Подраздел 1.4 Спецификации)**

Этот подраздел должен:

а) Представить полный список всех документов, на которые делаются ссылки где-либо внутри спецификации;

б) Идентифицировать каждый документ по заголовку, номеру отчета (если применяется), дате и организации-издателю;

в) Определить источники, из которых могут быть получены ссылки.

Эту информацию можно обеспечить ссылкой на приложение или другой документ.

**Краткий обзор (Подраздел 1.5 Спецификации)**

Этот подраздел должен:

а) Описать, что находится в оставшейся части спецификации;

б) Объяснить, как организована структура спецификации.

**Общее описание (Раздел 2 Спецификации)**

Этот раздел спецификации должен описывать общие факторы, которые влияют на программное изделие (-я) и требования, предъявляемые к нему. Этот раздел не устанавливает конкретные требования. Вместо этого, он обеспечивает предварительные сведения о тех требованиях, которые подробно определяются в разделе 3 спецификации, и делает их более простыми для понимания.

Этот раздел обычно состоит из шести подразделов, а именно:

а) Контекст изделия;

б) Функции изделия;

в) Характеристики пользователей;

г) Ограничения;

д) Допущения и зависимости;

е) Разделение требований.

**Контекст изделия (Подраздел 2.1 Спецификации)**

Этот подраздел спецификации должен описывать изделие в контексте с другими, связанными с ним изделиями. Если изделие является независимым и полностью автономным, то это должно быть явно сформулировано в документе. Если спецификация определяет изделие, которое является компонентом большей системы, как это часто бывает, то данный подраздел должен устанавливать связь требований этой большей системы с функциональными возможностями программного обеспечения, и должен идентифицировать интерфейсы между этой системой и программным обеспечением.

Может оказаться полезным использование блок-схемы, показывающей главные компоненты большей системы, соединений и внешних интерфейсов.

Этот подраздел должен также описывать как программное обеспечение функционирует в рамках различных ограничений. Например, эти ограничения могут включать:

а) Системные интерфейсы;

б) Интерфейсы пользователя;

в) Аппаратные интерфейсы;

г) Интерфейсы программного обеспечения;

д) Интерфейсы связи;

е) Память;

ж) Операции;

з) Требования по адаптации места использования.

**Системные интерфейсы**

Этот подраздел должен перечислять каждый системный интерфейс и идентифицировать функции программного обеспечения, выполняющие системные требования, и описание интерфейса для соответствия системе.

**Интерфейсы пользователя**

Этот подраздел должен указывать следующее:

а) *Логические характеристики каждого интерфейса между программным изделием и его пользователями.* Это включает характеристики конфигурации (например, требуемые форматы экрана, компоновку страницы или окна, содержание любых отчетов или меню, или наличие программируемых функциональных клавиш), необходимые для выполнения требований к программному обеспечению.

б) *Все аспекты оптимизации интерфейса с пользователем, который должен использовать систему.* Они могут просто включать список разрешений и запрещений различных способов представления системы пользователю. Одним из примеров может быть требование, предъявляемое к опции длинных или коротких сообщений об ошибках. Подобно всем другим, эти требования должны быть проверяемыми, например, «машинистка 4-го класса может выполнить задачу X за Z минут после 1 часа тренировки», а не «машинистка может выполнить задачу X» (Это также может быть определено в Атрибутах Программной Системы в разделе, озаглавленном Простота использования).

#### **Аппаратные интерфейсы**

Этот подраздел должен определять логические характеристики каждого интерфейса между программным изделием и аппаратными компонентами системы. Он включает конфигурационные характеристики (число портов, наборы команд и т.д.). Он также охватывает вопросы, касающиеся того, какие устройства должны поддерживаться, как они должны поддерживаться и протоколы. Например, поддержка терминала может указывать полноэкранную поддержку в противоположность построчной поддержке.

#### **Интерфейсы программного обеспечения**

Этот подраздел должен определять использование других необходимых программных изделий (например, системы управления данными, операционной системы или пакета математических программ) и интерфейсов с другими прикладными системами (например, связь между системой дебиторской задолженности и общей системой финансового учета). Для каждого требуемого программного изделия необходимо обеспечить следующее:

- Имя;
- Мнемонический код;
- Номер спецификации;
- Номер версии;
- Источник

Для каждого интерфейса необходимо обеспечить следующее:

а) Обсуждение назначения взаимодействующего программного обеспечения в отношении к этому программному изделию.

б) Определение интерфейса с точки зрения содержания и формата сообщений. Нет необходимости в детализации любого хорошо задокументированного интерфейса, но при этом требуется выполнить ссылку на документ, определяющий интерфейс.

#### **Интерфейсы связи**

Этот подраздел должен определять различные интерфейсы связи, такие как протоколы локальной сети и т.д.

#### **Ограничения памяти**

Этот подраздел должен определять любые применимые характеристики и ограничения первичной и вторичной памяти.

#### **Операции**

Этот подраздел должен определять стандартные и специальные операции, требуемые пользователем, такие как:

а) Различные режимы операций в организации пользователя (например, операции, запускаемые пользователем);

б) Периоды интерактивных операций и периоды автоматических операций;

в) Функции поддержки обработки данных;

г) Операции по резервированию и восстановлению.

**ПРИМЕЧАНИЕ** - Этот подраздел иногда указывается как часть раздела Интерфейсы пользователя.

#### **Требования к адаптации места использования**

Этот подраздел должен:

- а) Определять требования к любым данным или последовательностям инициализации, которые являются специфическими для данного места, задачи, или режима работы (например, значения сетки, пределы безопасности и т.д.);
- б) Определять характеристики места или характеристики, связанные с задачей, которые следует изменить, чтобы адаптировать программное обеспечение к конкретной установке.

#### **Функции изделия (Подраздел 2.2 Спецификации)**

Этот подраздел спецификации должен представлять сводку основных функций, которые будет выполнять программное обеспечение. Например, спецификация для программы бухгалтерского учета может использовать эту часть, чтобы выделить такие функции, как сопровождение счетов клиента, создание отчетов о движении средств клиента и подготовка счетов без упоминания большого количества деталей, которые требуются каждой из этих функций.

Иногда сводка функций, которая необходима для этой части, может быть взята непосредственно из раздела спецификации более высокого уровня (если он существует), который назначает отдельные функции программному изделию. Заметьте, что в целях обеспечения ясности:

- а) Функции должны быть организованы таким способом, который делает перечень функций понятным заказчику или любому человеку, читающему документ впервые.
- б) Могут быть использованы текстовые или графические методы, чтобы показать различные функции и их связи. Такая схема не предназначается для того, чтобы показать структуру изделия, а просто показывает логические связи между переменными.

#### **Характеристики пользователя (Подраздел 2.3 Спецификации)**

Этот подраздел спецификации должен описывать общие характеристики пользователей изделия, включающие образовательный уровень, опыт и специальные технические знания. Он не должен использоваться для формулировки конкретных требований, а должен представлять основания, по которым некоторые специфические требования далее определяются в разделе 3 спецификации.

#### **Ограничения (Подраздел 2.4 Спецификации)**

Этот подраздел спецификации должен обеспечить общее описание любых других позиций, которые будут ограничивать опции разработчика. Они включают:

- а) Регулирующие политики;
- б) Аппаратные ограничения (например, требования к синхронизации сигналов);
- в) Интерфейсы с другими приложениями;
- г) Параллельные операции;
- д) Функции контроля;
- е) Функции управления;
- ж) Требования к языку более высокого порядка;
- з) Протоколы квитирования сигналов (например, XON-XOFF, ACK-NACK);
- и) Требования к надежности;
- к) Критичность применения изделия;
- л) Критерии безопасности и защиты.

#### **Допущения и зависимости (Подраздел 2.5 Спецификации)**

В этом подразделе спецификации должны быть перечислены все факторы, которые влияют на требования, сформулированные в спецификации. Эти факторы не являются архитектурными ограничениями программного обеспечения, это, скорее, любые их изменения, которые могут воздействовать на требования в спецификации.

Например, может быть сделано допущение, что определенная операционная система будет присутствовать на аппаратных средствах, предназначенных для программного изделия. Если в действительности операционная система отсутствует, тогда спецификация должна быть изменена соответствующим образом.

#### **Распределение требований (Подраздел 2.6 Спецификации)**

Этот подраздел спецификации должен идентифицировать требования, которые могут быть отложены до появления будущих версий системы.

#### **Специфические требования (Раздел 3 Спецификации)**

Этот раздел спецификации должен содержать все требования к программному обеспечению на уровне детализации, достаточной, чтобы дать проектировщикам возможность разработать систему, удовлетворяющую этим требованиям, а испытателям - убедиться, что система удовлетворяет этим требованиям. В этом разделе каждое сформулированное требование должно восприниматься пользователями, операторами или другими внешними системами. Эти требования должны включать как минимум описание каждого входного воздействия (стимула) на систему, каждого выходного сигнала (отклика) системы и всех функций, выполняемых системой в ответ на входное воздействие или для поддержания выходного сигнала.

Поскольку часто эта часть является самой большой и наиболее важной частью спецификации, применяются следующие принципы:

а) Специфические требования должны быть сформулированы в соответствии со всеми характеристиками, описанными ранее.

б) Специфические требования должны иметь перекрёстные ссылки к более ранним документам, к которым они имеют отношение.

в) Все требования должны быть однозначно идентифицируемы.

г) Необходимо уделить особое внимание оформлению требований, чтобы максимизировать их удобочитаемость.

Перед исследованием специфических способов организации требований полезно осмыслить различные элементы, в которые входят требования, как описано ранее.

#### **Внешние интерфейсы**

Этот раздел должен представлять детализированное описание всех входных и выходных данных системы программного обеспечения. Он должен дополнять описание интерфейса, и не должен повторять эту информацию.

Он должен включать содержание и формат следующим образом:

- а) Наименование позиции;
- б) Описание назначения;
- в) Источник входных данных или адресат выходных данных;
- г) Допустимый диапазон, точность и/или допуск;
- д) Единицы измерения;
- е) Синхронизация;
- ж) Связи с другими входами/выходами;
- з) Форматы/организация экрана;
- и) Форматы/организация окна;
- к) Форматы данных;
- л) Форматы команд;
- м) Сообщения о конце.

#### **Функции**

Функциональные требования должны определять фундаментальные операции, которые должны выполняться программным обеспечением при принятии и обработке входных данных и обработке и генерации выходных данных. Они обычно перечисляются как формулировки типа «должно», начинающиеся с «Система должна .....,».

Они включают:

- а) Проверку достоверности по входам;

- б) Точную последовательность операций;
- в) Отклики на ненормальные ситуации, включая:
  - 1) Переполнение
  - 2) Средства связи
  - 3) Обработка и устранение ошибок;
- г) Влияние параметров;
- д) Связь выходов с входами, включая:
  - 1) Последовательности ввода/вывода
  - 2) Формулы для преобразования ввода-вывода.

Может оказаться удобным разбить разделы функциональных требований на подфункции или подпроцессы. Это не означает, что структура программного обеспечения будет организована таким же образом.

#### **Требования к рабочим характеристикам (производительности)**

В этом подразделе должны быть определены требования к статическим и динамическим числовым характеристикам, предъявляемым к программному обеспечению или ко взаимодействию пользователя с программным обеспечением в целом. Статические числовые требования могут включать:

- а) Число поддерживаемых терминалов;
- б) Число одновременно поддерживаемых пользователей;
- в) Количество и тип обрабатываемой информации.

Требования к статическим числовым характеристикам иногда указываются в отдельном разделе, озаглавленном «Диапазон значений».

Требования к динамическим числовым характеристикам могут включать, например, количество транзакций, задач и количество данных, которые будут обрабатываться за определенное время для нормальных и пиковых условий рабочей нагрузки.

Все эти требования должны быть установлены в измеряемых терминах.

Например,

*95 % транзакций должны обрабатываться не более чем за 1 с.*

а не

*Оператор не должен ждать завершения транзакции.*

**ПРИМЕЧАНИЕ** - Числовые ограничения, применяемые к одной определенной функции, обычно указываются как часть описания подпункта по обработке этой функции.

#### **Требования к логическому устройству базы данных**

В этом подразделе должны быть определены требования к логике любой информации, которая должна размещаться в базе данных. Она может включать следующее:

- а) Типы информации, используемой различными функциями;
- б) Частота использования;
- в) Возможности доступа;
- г) Информационные объекты и их связи;
- д) Ограничения целостности;
- е) Требования к сохранности данных.

#### **Архитектурные ограничения**

Здесь должны указываться архитектурные ограничения, которые могут налагаться другими стандартами, аппаратными ограничениями и т.д.

#### **Соответствие стандартам**

Этот подраздел должен определять требования, происходящие из существующих стандартов или инструкций. Они могут включать следующее:

- а) Формат отчета;
- б) Правила именования данных;
- в) Процедуры учета;
- г) Журналирование операций.

Например, может быть указано требование для программного обеспечения отслеживать (журналировать) все действия. Такие журналы необходимы для некоторых программных приложений, чтобы обеспечить соответствие минимальным регулирующим или финансовым стандартам. Требование к журналированию операций может, например, утверждать, что все изменения базы данных платежных ведомостей должны быть записаны в файле журнала, включая предыдущие и последующие значения.

#### **Атрибуты (качества) программной системы**

Существует ряд атрибутов программного обеспечения, которые могут служить в качестве требований.

Важно, чтобы необходимые атрибуты были определены таким образом, чтобы их выполнение можно было объективно проверить.

##### **1. Надёжность**

Этот подраздел должен определять факторы, необходимые для установления требуемой надёжности системы программного обеспечения при поставке.

##### **2. Доступность**

Этот подраздел должен определять факторы, необходимые для обеспечения заданного уровня доступности для всей системы, такие как контрольная точка, восстановление и перезапуск.

##### **3. Защита**

Этот подраздел должен определять факторы, которые защищают программное обеспечение от случайного или злонамеренного доступа, использования, изменения, разрушения или раскрытия. Специфические требования в этой области могут включать потребность в:

- а) Использовании определённых методов криптографии;
- б) Ведении специального журнала или наборов исторических данных;
- в) Назначении некоторых функций различным модулям;
- г) Ограничении связи между некоторыми областями программы;
- д) Проверке целостности данных для критических переменных.

##### **4. Удобство сопровождения**

Этот подраздел должен определять атрибуты программного обеспечения, которые относятся к простоте сопровождения самого программного обеспечения. Может иметься некоторое требование к определённой модульности системы, интерфейсам, сложности и т.д. Требования не должны указываться в этом подразделе только потому, что они считаются хорошими практиками проектирования.

##### **5. Мобильность**

Этот подраздел должен определять атрибуты программного обеспечения, которые относятся к простоте перенесения программного обеспечения на другие машины и/или операционные системы. Они могут включать следующее:

- а) Процентное соотношение компонентов с кодом, зависящим от платформы машины;
- б) Процентное соотношение кода, зависящего от платформы машины;
- в) Применение промышленного переносимого (многоплатформенного) языка;
- г) Применение определённого компилятора или подмножества языка;
- д) Применение определённой операционной системы.

### **3. Порядок выполнения работы**

#### **3.1 Разработка технического задания на программный продукт**

1. В соответствии с вариантом задания выполнить разработку технического задания, которое должно включать (пример в Приложении 2):

- введение;
- основание для разработки;
- назначение;
- требования к программе и программному продукту;



- требования к программной документации.

### **3.2 Разработка спецификаций на программный продукт**

1. В соответствии с техническим заданием выполнить разработку спецификаций на программный продукт, которые должны включать:

- спецификации;
- словарь терминов.

**3.3 Варианты заданий (номер варианта задания 0,1,2,3,4,5 определяется как остаток от деления по модулю 6 последней цифры зачетной книжки (студенческого билета) студента или суммы последних цифр зачетных книжек (студенческих билетов) бригады студентов из двух-трех человек)**

0. Реализация ftp-сервера. Команды, которые должен поддерживать сервер, должны быть следующие: положить файл, взять файл, удалить файл, создать директорию, удалить директорию, показать содержимое директории, переместиться в другую директорию, поддержка пассивного режима и др. Должна быть совместимость со стандартным ftp-протоколом.

1. Реализация ftp-клиента. Команды, которые должен поддерживать клиент, должны быть следующие: положить файл, взять файл, удалить файл, создать директорию, удалить директорию, показать содержимое директории, переместиться в другую директорию, поддержка пассивного режима и др. Должна быть совместимость со стандартным ftp-протоколом.

2. Реализация pop3-сервера. Команды, которые должен поддерживать сервер, должны быть следующие: авторизация (пользователь/пароль), листинг писем, просмотреть письмо, удалить письмо и др. Должна быть совместимость со стандартным pop3-протоколом.

3. Реализация простейшего smtp-сервера. Команды, которые должен поддерживать сервер, должны быть следующие: предназначение письма, содержание письма и др. Должна быть совместимость со стандартным smtp-протоколом.

4. Реализация chat-сервера над TCP/IP. Реализовать подобие ICQ-сервера над очередями сообщений IPC (кольцевой буфер). Должны быть реализованы следующие возможности: регистрация пользователя, авторизация пользователя, посылка сообщения от пользователя к пользователю, список пользователей на сервере.

5. Реализация chat-клиента над TCP/IP. Реализовать подобие ICQ-клиента. Должны быть реализованы следующие возможности: регистрация пользователя на сервере, авторизация пользователя, получение списка клиентов от сервера, посылка сообщения от пользователя к пользователю как через сервер, так и клиенту напрямую.

## **4. Содержание отчёта**

1. Название, цель работы
2. Разработанное техническое задание.
3. Разработанные спецификации.
4. Ответы на контрольные вопросы.

Отчет должен быть предоставлен в электронном виде

## **5. Вопросы к защите**

1. Спецификация программного обеспечения.
2. Виды спецификаций.
3. Состав функциональной спецификации.
4. Перечислите этапы разработки программных продуктов.
5. Для чего необходимо техническое задание?
6. Кто занимается разработкой технического задания?
7. Какие пункты включает техническое задание?
8. Для чего разрабатываются спецификации на программный продукт?
9. Что должны включать спецификации на программный продукт?

10. Что должна содержать спецификация процессов?
11. Что такое словарь терминов и для чего он используется?
12. Что такое диаграмма переходов состояний и для чего ее используют?
13. Что такое диаграмма потоков и для чего ее используют?

---

## Приложение 1

### Самостоятельная работа по лабораторной работе № 1 «Разработка спецификаций структурных единиц»

Самостоятельная работа по теме занятия включает в себя:

- изучение теоретического материала лекционных занятий, учебной литературы, Интернет-ресурсов, раздела «Краткие сведения из теории» настоящего описания ЛР;
- выполнение практических заданий и решение задач

Задачи и практические задания

1. Разработать диаграммы переходов состояний.
2. Разработать диаграммы потоков с детализацией.

---

## Приложение 2

### Пример технического задания

#### Техническое задание

##### 1 Введение

При проектировании сложных систем, таких как микропроцессорная автоматизированная системы или её узлы, возникает ряд комбинаторно-оптимизационных задач структурного синтеза, на всех этапах, начиная с эскизного проектирования и заканчивая разработкой конструкторской документации. К таким задачам относятся, например, схемная компоновка, размещение компонентов в монтажном пространстве, коммутации соединений и т. д. Автоматизация указанных процессов является актуальной проблемой. Поэтому целью настоящего курсового проекта является автоматизация решения комбинаторно-оптимизационных задач.

Для достижения указанных задач необходимо решить ряд инженерных задач, а именно:

- анализ области применения комбинаторно-оптимизационных задач;
- математические методы выбранных задач;
- разработка архитектуры ПС;
- выбор метода проектирования, программных средств реализации ПС;
- разработка структуры данных для хранения информации;
- разработка алгоритмов решения задач;
- тестирование ПС;
- разработка руководства программиста;
- разработка руководства пользователя;
- оценка экономической эффективности внедрения ПС;
- безопасность труда пользователя.

Решение данных задач возможно на основании методов дискретной математики, программирования на языках высокого уровня, теории баз данных, статистических методов и моделей.

##### 2 Основание для разработки

Система разрабатывается на основании учебного плана подготовки специалистов 230105.65 - ПОВТАС и рабочей программе по дисциплине специализации «Системы автоматизации проектирования программного обеспечения»

### **3 Назначение**

Целью данной разработки является создание программного средства, позволяющего автоматизировать работу по решению комбинаторно-оптимизационных задач коммутации.

Первая версия системы предназначена для решения небольшого круга комбинаторно-оптимизационных задач на графах (поиск кратчайшего пути, минимального покрывающего дерева и покрывающего цикла минимальной длины). В следующих версиях предполагается увеличение количества решаемых задач.

Пользователями могут выступать научные работники и инженеры, занимающиеся проектированием компьютерной техники, и студенты соответствующих специальностей. Пользователями могут также быть и специалисты других предметных областей, которым приходится решать подобные задачи.

### **4 Требования к программе или программному изделию**

#### **4.1 Требования к функциональным характеристикам**

Система должна представлять совокупность методических и программных средств решения следующих задач:

- задачи построения минимального покрывающего дерева;
- поиск покрывающего цикла минимальной длины (задача коммивояжера);
- задачи поиска кратчайшего пути.

Для этих задач должны быть реализованы:

- алгоритмы, обеспечивающие получение точного решения;
- в том случае, если точное решение получается алгоритмом, имеющим неполиномиальную вычислительную сложность, то необходимо дополнительно разработать алгоритм, обеспечивающий получение приближенных решений с полиномиальной вычислительной сложностью.

Методическое обеспечение должно быть реализовано в пользовательском интерфейсе системы, который должен предполагать выбор задачи, метода и алгоритма ее решения; ввод данных; решение проектной задачи и сохранение исходных данных, промежуточных и окончательных результатов во встроенной базе данных для последующего анализа.

#### **4.2 Требования к надежности**

Предусмотреть контроль вводимой информации и блокировку некорректных действий пользователя при работе с системой.

Обеспечить корректное завершение вычислений с соответствующей диагностикой при превышении имеющихся вычислительных ресурсов.

Обеспечить целостность информации, хранящейся в базе данных.

#### **4.3 Требования к составу и параметрам технических средств**

Система должна работать на IBM совместимых персональных компьютерах.

Минимальная конфигурация:

Тип процессора.....Pentium-100;

Объем оперативного запоминающего устройства .....16 Мб;

Тип монитора.....SVGA (15').

#### **4.4 Требования к информационной и программной совместимости**

Система должна работать под управлением операционной системы Windows'95 и выше.

### **5 Требования к программной документации**

Разрабатываемая система должна включать справочную информацию о работе системы и подсказки пользователю.

В состав сопровождающей документации должны входить:

- пояснительная записка;
- руководство пользователя.