Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ**

**УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

Кафедра автоматизированных систем управления (АСУ)

**ОСНОВЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Методические указания по лабораторным занятиям, самостоятельной и индивидуальной работе бакалавров всех форм обучения

**Направление 09.03.01 Информатика и вычислительная техника**

**Профиль «Программное обеспечение средств вычислительной техники и автоматизированных систем»**

**Томск 2016**

**Калайда В.Т.**

Основы разработки программного обеспечения. Методические указания по лабораторным занятиям, самостоятельной и индивидуальной работе бакалавров всех форм обучения. Направление 09.03.01 Информатика и вычислительная техника. Профиль «Программное обеспечение средств вычислительной техники и автоматизированных систем»./ В.Т. Калайда. – Томск: ТУСУР, 2016. – 16 с.

Методические указания утверждены на заседании кафедры автоматизированных систем управления 12 февраля 2016 г., протокол № 5.

© Калайда В.Т., 2016

**ОГЛАВЛЕНИЕ**

[1 ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ 4](#_Toc465772192)

[1.1 Цели дисциплины: 4](#_Toc465772193)

[1.2 Задачи дисциплины 4](#_Toc465772194)

[2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП 4](#_Toc465772195)

[3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ 4](#_Toc465772196)

[4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ 5](#_Toc465772197)

[4.1 Содержание разделов дисциплины (по лекциям) 5](#_Toc465772198)

[4.2 Лабораторные работы. Цель работы. Задания. 7](#_Toc465772199)

[4.3 Индивидуальная работа студента (Тематика семинаров) 10](#_Toc465772200)

[4.4 Методические рекомендации по подготовке рефератов 12](#_Toc465772201)

[4.5 Требования, предъявляемые к оформлению отчета по реферату 12](#_Toc465772202)

[5. САМОСТОЯТЕЛЬНАЯ РАБОТА 14](#_Toc465772203)

[6 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ 15](#_Toc465772204)

[6.1 Основная литература 15](#_Toc465772205)

[6.2 Дополнительная литература 15](#_Toc465772206)

[6.3 Перечень пособий, методических указаний и материалов, используемых в учебном процессе 16](#_Toc465772207)

[6.4 Базы данных, информационно-справочные и поисковые системы 16](#_Toc465772208)

[7 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ 16](#_Toc465772209)

# 1 ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

## 1.1 Цели дисциплины:

**Целью дисциплины** − рассмотрение вопросов проектирования, разработки и испытаний больших программных систем с точки зрения требований разработчика. Этот курс обобщает знания, полученные студентами, и обеспечивает изучение современных технологий разработки программного обеспечения.

## 1.2 Задачи дисциплины

**Задачей дисциплины** является изучение методов разработки программного обеспечения, способов создания функциональных спецификаций, методов проектирования программных комплексов, создания абстрактных типов данных, доказательства правильности программ, организации тестов и сопровождения программных комплексов.

# 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Дисциплина «Основы разработки программного обеспечения» относится к числу дисциплин Профессионального цикла вариативной части.

Успешное овладение данной дисциплиной предполагает предварительные знания по языкам программирования, полученные в дисциплинах: «Современные проблемы информатики и вычислительной техники», «Современные средства программирования».

Зная технологию разработки программного обеспечения, студенты смогут использовать эти знания при дальнейшем проектировании программных систем, при изучении дисциплины «Научно-исследовательская работа магистра», при подготовке магистерской диссертации.

# 3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Процесс изучения дисциплины «Технология разработки программного обеспечения» направлен на формирование следующих компетенций:

1. Способность разрабатывать модели компонентов информационных систем, включая модели баз данных и модели интерфейсов «человек – электронно-вычислительная машина» (**ПК-1**);

2. Способность обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности (**ПК-3**).

В результате освоения дисциплины «Основы разработки программного обеспечения» обучающийся должен:

**Знать:**

* принципы разработки и методы проектирования программных систем, методы управления проектированием программных систем и организации коллективов разработчиков, государственные стандарты, регламентирующие процесс разработки программных систем и их описания;
* теоретические основы методов разработки программного обеспечения, способы создания функциональных спецификаций, методы проектирования программных комплексов, создания абстрактных типов данных, доказательства правильности программ, организации тестов и сопровождения программных комплексов;

**Уметь:**

* самостоятельно выполнять цикл проектирования программного обеспечения, разрабатывать спецификации и абстрактные типы данных на основе анализа требований, предъявляемых к программному обеспечению, доказывать правильность программ, проектировать и кодировать необходимые тесты, пользоваться стандартными терминами и определениями, читать научные статьи и пользоваться литературой для самостоятельного решения научно - исследовательских задач, связанных с разработкой программных систем;

**Владеть:**

* перспективными направлениями работ и методическими подходамив области проектирования и разработки больших программных комплексов.

# 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

## 4.1 Содержание разделов дисциплины (по лекциям)

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование разделов | Содержание разделов | Формируемые компетенции  (ОК, ПК) |
| 1 | 2 | 3 | 5 |
| 1. | **ВВЕДЕНИЕ В ДИСЦИПЛИНУ** | Краткая характеристика дисциплины, её цели и задачи, порядок изучения материала, связи с другими дисциплинами учебного плана. Перечень дисциплин, усвоение которых необходимо студентам для изучения данной дисциплины. Основы методики и форм контроля самостоятельной работы, краткая характеристика учебной литературы. | ПК-1;  ПК-3 |
| 2. | **ОРГАНИЗАЦИЯ УПРАВЛЕНИЯ ПРОЕКТИРОВАНИЕМ ПРОГРАММНОГО ИЗДЕЛИЯ.** | Понятие программного изделия, как средства общения. Нисходящий анализ процесса управления проектированием программного изделия. Организация взаимодействия. Установление целей, средства их достижения. Подбор и обучение кадров. | ПК-1;  ПК-3 |
| 3. | **ОРГАНИЗАЦИЯ ПЛАНИРОВАНИЯ РАЗРАБОТОК ПРОГРАММНОГО ИЗДЕЛИЯ.** | Организационная структура группы планирования. Планы, связанные с созданием программного изделия. Опытный образец изделия. Организация планирования в фазах конструирования и кодирования. Обязанности группы планирования при разработке и утверждении планов разработки программного изделия. | ПК-1;  ПК-3 |
| 4. | **ОРГАНИЗАЦИЯ РАЗРАБОТКИ ПРОГРАММНОГО ИЗДЕЛИЯ.** | Организация разработки программного изделия в фазе исследований. Организация разработки программного изделия в фазе анализа осуществимости. Организация разработки программного изделия в фазе конструирования (проектирования). Организация разработки программного изделия в фазе программирования. Организация разработки программного изделия в фазе оценки. Окончание проекта. Участие группы разработки в фазовых обзорах. | ПК-1;  ПК-3 |
| 5. | **ОРГАНИЗАЦИЯ ОБСЛУЖИВАНИЯ РАЗРАБОТКИ ПРОГРАММНОГО ИЗДЕЛИЯ.** | Организация обслуживания разработки программного изделия в фазе исследования. Организация обслуживания разработки программного изделия в фазах анализа осуществимости и конструирования. Организация обслуживания разработки программного изделия в фазах программирования и оценки. Организация обслуживания разработки программного изделия в фазе использования. Участие группы обслуживания в фазовых обзорах. | ПК-1;  ПК-3 |
| 6. | **ОРГАНИЗАЦИЯ ВЫПУСКА ДОКУМЕНТАЦИИ.** | Организация выпуска документации в фазах исследования и анализа осуществимости. Организация выпуска документации в фазах конструирования и программирования. | ПК-1;  ПК-3 |
| 7. | **ОРГАНИЗАЦИЯ ИСПЫТАНИЙ ПРОГРАММНЫХ ИЗДЕЛИЙ.** | Организационная структура группы испытаний. Организация испытаний в фазах исследований и анализа осуществимости. Организация испытаний в фазах конструирования и программирования. Организация испытаний в фазе оценки. Организация испытаний в фазе использования. Участие группы испытаний в фазовых обзорах. | ПК-1;  ПК-3 |

## 4.2 Лабораторные работы. Цель работы. Задания.

**Лабораторная работа №1. «Назначение и содержание соглашения о требованиях**» .

**Цель работы**: научиться подготавливать программную документацию по организации коллективной разработки ПО.

**Краткие теоретические сведения:**

В соглашении о требованиях должно содержаться письменное изложение того, что будет сделано и что не будет делаться при выпуске программного обеспечения.

Документ «Соглашение о требованиях» является основным средством управления разработкой программного обеспечения или генеральным планом его разработки.

Все участники разработки программного обеспечения должны выполнять то, что установлено в документе «Соглашение о требованиях» или запрашивать и получать разрешение на его изменение.

Предполагается, что все утверждения, включенные в соглашение о требованиях, являются требованиями, если они не определены как цели.

Каждый документ «Соглашение о требованиях» должен точно соответствовать установленной форме. Тогда каждый раздел можно будет найти в одном и том же месте аналогичного документа любой разработки программного обеспечения. В документ целесообразно включить заголовки всех предусмотренных разделов, если только специально не оговариваются условия, при которых какой-либо раздел может быть опущен. Тогда при рассмотрении документа будет решаться вопрос, действительно ли такие разделы нужны.

Соглашения о требованиях пишутся на естественном языке в терминах понятных и пользователю и разработчику программного обеспечения. Стороны должны четко представлять каждое требование.

Следует напомнить, что пользователь несет ответственность за проверку требований на полноту и точность, а разработчик - за проверку их на осуществимость и понятность.

**Задание**

Разработать соглашение о требованиях на разрабатываемый программный продукт.

**Лабораторная работа №2 « Методы написания спецификаций».**

**Цель работы**: научиться определять спецификации ПО.

**Краткие теоретические сведения:**

На этапе определения спецификаций осуществляется точное описание функций, реализуемых ЭВМ, а также задаются структуры входных и выходных данных, методы и средства их размещения. Определяются алгоритмы обработки данных.

Центральным вопросом определения спецификаций является проблема организации базы данных. При этом решается комплекс вопросов, имеющих отношение к структуре файлов, организации доступа к ним, модификации и удаления.

В случае, когда новая система создается на основе существующих, составной частью спецификаций является схема (алгоритм) приведения существующей базы данных к новому формату. Такое преобразование может потребовать разработку специальной программы, которая становится ненужной после ее первого и единственного использования.

Все эти вопросы должны быть отражены в функциональных спецификациях, которые представляют собой документ, являющийся основополагающим в процессе разработки системы, так как содержит конкретное описание последней.

Чем подробней составлены спецификации, тем меньше вероятность возникновения ошибок.

В спецификациях должны быть представлены данные для тестирования элементов системы и системы в целом. Это требование является объективным и обязательным, так как на данном этапе на параметры тестирования не будет оказывать влияние конкретная реализация системы.

Так как функциональные спецификации описывают принятые решения в целом, данный документ можно использовать для начальных оценок временных затрат, числа специалистов и других ресурсов, необходимых для проведения работ.

В общем случае спецификации определяют те функции, которые должна выполнять система, не указывая, каким образом это достигается. Составление подробных алгоритмов на этом этапе преждевременно и может вызвать нежелательные осложнения.

**Задание**

На основании технического задания и соглашения о требованиях определить внешнюю и внутреннюю спецификации.

**Лабораторная работа №3 «Доказательство правильности программ».**

**Цель работы**: научиться доказывать правильность элементарных программ относительно спецификации.

**Краткие теоретические сведения:**

Одним из важных методов повышения эффективности проектирования программ является верификация программ или математическое доказательство того, что программа работает правильно. Для доказательства правильности программ используется аксиоматический подход, при котором применяется теория перечисления предикатов. Предполагается, что каждый оператор в программе выполняет заранее определенные действия, зависящие только от синтаксиса языка. Для двух предикатов P и Q и оператора S необходимо определить истинно ли выражение:

**«Если P истинно и если выполняется оператор S , то Q истинно».**

Предикат P является спецификацией правильного выполнения оператора S , а предикат Q будет истинным после выполнения оператора S и является спецификацией следующего за S оператора.

Если это утверждение распространить на все операторы программы и если P является спецификацией первого оператора, а Q истинно после окончания программы, то будет доказана правильность всей программы относительно предикатов P и Q . Эту конструкцию можно записать в следующем виде:

{P} S {Q} ,

где P называется предусловием истинности Q после выполнения программы S .

Доказательство правильности программы заключается в определении, является ли выражение {P} S {Q} истинным относительно входных спецификаций P , выходных спецификаций Q и операторов S программы. Если {P} S {Q} истинно, то это означает, что доказана правильность программы S относительно P и Q .

**Задание**

Используя расширенные аксиомы исчисления предикатов доказать правильность (или ошибочность) предложенной преподавателем программы.

**Лабораторная работа №4 «Технология написания тестов» – 4 часа.**

**Цель работы**: Научиться определять классы эквивалентности в спецификациях и подготавливать необходимый набор тестов по ним.

**Краткие теоретические сведения:**

Общие принципы тестирования.

Этап тестирования обычно в финансовых затратах составляет половину расходов на создание системы. Плохо спланированное тестирование приводит к существенному увеличению сроков разработки системы и является основной причиной срывов графиков разработки.

В процессе тестирования используются данные, характерные для системы в рабочем состоянии, т.е. данные для тестирования выбираются случайным образом. План проведения испытаний должен быть составлен заранее, обычно на этапе проектирования.

Тестирование подразумевает три стадии:

* 1. автономное;
  2. комплексное;
  3. системное.

При автономном тестировании модуль проверяется с помощью данных, подготовленных программистом. При этом программная среда модуля имитируется с помощью программ управления тестированием, содержащих фиктивные программы вместо реальных подпрограмм, к которым имеется обращение из данного модуля (заглушки). Подобную процедуру называют программным тестированием, а программу тестирования – UUT (тестирующей программой). Модуль, прошедший автономное тестирование, подвергается комплексному тестированию.

В процессе комплексного тестирования проводится совместная проверка групп программных компонентов. В результате имеем полностью проверенную систему. На данном этапе тестирование обнаруживает ошибки, пропущенные на стадии автономного тестирования. Исправление этих ошибок может составлять до четверти от общих затрат.

Системное (или оценочное) тестирование – это завершающая стадия проверки системы, т.е. проверка системы в целом с помощью независимых тестов. Независимость тестов является главным требованием. Обычно Заказчик на стадии приемки работ настаивает на проведении собственного системного тестирования. Для случая, когда сравниваются характеристики нескольких систем (имеется альтернативная разработка), такая процедура известна как сравнительное тестирование.

**Технология тестирования, классы эквивалентности.**

Одним из способов изучения данного вопроса является исследование стратегии тестирования, называемой стратегией черного ящика, тестированием с управлением по данным или тестированием с управлением по входу-выходу. При использовании этой стратегии программа рассматривается как черный ящик.

Иными словами, такое тестирование имеет целью выяснение обстоятельств, в которых поведение программы не соответствует ее спецификации. При таком подходе обнаружение всех ошибок в программе является критерием исчерпывающего входного тестирования. Последнее может быть достигнуто, если в качестве тестовых наборов использовать все возможные наборы входных данных.

Разработка тестов методом эквивалентного разбиения осуществляется в два этапа:

1) выделение классов эквивалентности;

2) построение тестов.

Классы эквивалентности выделяются путем выбора каждого входного условия (обычно это предложение или фраза в спецификации) и разбиением его на две или более группы. Для проведения этой операции используют таблицу, подобную табл. 1.

Таблица 1 – Форма таблицы для перечисления классов эквивалентности

|  |  |  |
| --- | --- | --- |
| Входные условия | Классы эквивалентности | |
| Правильные | Неправильные |
|  |  |  |

Различают два типа классов эквивалентности: правильные классы эквивалентности, представляющие правильные входные данные программы, и неправильные классы эквивалентности, представляющие все другие возможные состояния условий (т.е. ошибочные входные значения). Таким образом, придерживаются одного из принципов необходимости сосредоточивать внимание на неправильных или неожиданных условиях.

**Задание**

Идентифицировать входные условия и по ним определить классы эквивалентности, построить соответствующие тесты. В заключении, дать рекомендации по устранению ошибок, выявленных в результате тестирования.

## 4.3 Индивидуальная работа студента (Тематика семинаров)

1. Этапы разработки программного обеспечения. Анализ требований, предъявляемых к системе. Жизненный цикл программного обеспечения. Функциональные спецификации. Определение спецификаций. Проектирование. Кодирование.
2. Тестирование: программное, системное, оценочное и сравнительное тестирование. Сбой системы, выброс, ошибка. Испытания. Верификация системы. Правильность и надежность программ. Эксплуатация и сопровождение. Периоды обновления.
3. Организация интерфейса между модулями, написанными разными программистами. Выполнение проекта. Бригада главного программиста. Методика оценки затрат. Методика инженерно-технической оценки затрат. Методика экспертных оценок. Метод алгоритмического анализа. Пошаговый анализ. Закон Паркинсона. Затраты на завершения разработки.
4. Оценка длительности разработки на основе распределения Рэлея. Контрольные точки. Средства обработки. Надежность. Концептуальная целостность. "Уровни правильности" программ. Методы программирования.
5. Определение спецификаций. Язык определения задач и анализатор определения задач (PSL/PSA).
6. Система структурного проектирования SADT. Структурное проектирование. Методика Джексона. Стратегия объединения различных методов проектирования.
7. Язык проектирования программ PDL. Операторы выбора. Операторы цикла. Операторы описания данных. Операторы ввода вывода и вызова процедур. Оператор leave. Предложения на естественном языке.
8. Нисходящее проектирование и нисходящая разработка. Пошаговое совершенствование. Восходящее проектирование. Структурное проектирование. Простая программа. Элементарная программа. Управляющие структуры, способы их описания.
9. Скалярные и агрегативные типы данных. Массивы. Структуры. Списки. Очереди. Стеки. Множества. Графы. Деревья.
10. Абстрактные конструкции. Фиксированные данные абстрактного типа. Размещение указателей. Защита данных от несанкционированного доступа. Правильность программ.
11. Аксиомы: правила следствия; аксиома присвоения; аксиома следования; аксиома цикла; аксиома выбора. Правила целочисленной арифметики - коммутативность, ассоциативность, дистрибутивность, вычитания, обработка констант.
12. Стратегия тестирования. Имена переменных. Константы. Входные данные. Списки параметров. Проверка спецификаций. Данные для тестирования. Формализация тестирования программ.
13. Стандартные методы проектирования. Разбиение задачи на независимые подзадачи. Разбиение задачи на одинаковые по сложности части. Стратегия распределения памяти. Сопрограммы.
14. Понятие изделия, как средства общения. Нисходящий анализ процесса управления созданием программного изделия. Установление целей и средства их достижения. Подбор и обучение кадров.
15. Организация планирования разработки программного изделия. Виды планов. Декомпозиция планов. Организационная структура группы планирования. Виды планов, связанных с созданием программного изделия. Организация планирования разработки программного изделия. Вопросы, рассматриваемые в фазовых обзорах группой планирования,
16. Управление проектом. Организация работы группы разработки в фазах создания программного изделия. Организация работы группы обслуживания в фазах создания программного изделия. Организация работы группы выпуска документации в фазах создания программного изделия. Организация испытаний программного изделия.
17. Психология и экономика тестирования программ. Принципы тестирования. Инспекции, сквозные просмотры и обзоры программы. Список вопросов для выявления ошибок при инспекции. Тестирование путем покрытия логики программы. Эквивалентное разбиение.
18. Анализ граничных значений. Применение функциональных диаграмм. Предположение об ошибке. Стратегия. Понятие изделия, как средства общения. Нисходящий анализ процесса управления созданием программного изделия. Установление целей и средства их достижения.
19. Подбор и обучение кадров. Организация планирования разработки программного изделия. Виды планов. Декомпозиция планов. Организационная структура группы планирования. Виды планов, связанных с созданием программного изделия. Организация планирования разработки программного изделия. Вопросы, рассматриваемые в фазовых обзорах группой планирования.
20. Управление проектом. Организация работы группы разработки в фазах создания программного изделия. Организация работы группы обслуживания в фазах создания программного изделия. Организация работы группы выпуска документации в фазах создания программного изделия. Организация испытаний программного изделия. Рекурсия. Динамическое программирование. Моделирование. Алгоритм выбора из конечного числа состояний.

## 4.4 Методические рекомендации по подготовке рефератов

Тема выбирается магистрантом из числа предложенных выше или может быть определена самостоятельно по рекомендации научного руководителя. Реферат должен включать в себя оглавление, введение, основную часть, заключение, биографические справки об упоминаемых в тексте ученых и подробный библиографический список, составленный в соответствии со стандартными требованиями к оформлению литературы, в том числе к ссылкам на электронные ресурсы. Работа должна носить самостоятельный характер, в случае обнаружения откровенного плагиата (дословного цитирования без ссылок) реферат не засчитывается. Сдающий реферат магистрант должен продемонстрировать умение работать с литературой, отбирать и систематизировать материал, увязывать его с существующими математическими теориями и фактами общей истории.

**Во введении** обосновывается актуальность выбранной темы, определяются цели и задачи реферата, приводятся характеристика проработанности темы в историко-математической литературе и краткий обзор использованных источников.

**В основной части**, разбитой на разделы или параграфы, излагаются основные факты, проводится их анализ, формулируются выводы (по разделам). Необходимо охарактеризовать современную ситуацию, связанную с рассматриваемой тематикой.

**Заключение** содержит итоговые выводы и, возможно, предположения о перспективах проведения дальнейших исследований по данной теме.

**Биографические данные** можно оформлять сносками или в качестве приложения к работе.

**Список литературы** может быть составлен в алфавитном порядке или в порядке цитирования, в полном соответствии со стандартными требованиями к библиографическому описанию.

## 4.5 Требования, предъявляемые к оформлению отчета по реферату

Оформление текста отчета по реферату должно отвечать требованием ОС ТУСУР.

Текст написан на компьютере через полтора межстрочных интервала. Размер шрифта – 12 ÷14 пт.

Текст работы следует писать или печатать, соблюдая следующие размеры полей:

- левое – 25 мм.;

- правое – 15 мм.;

- верхнее – 20 мм.;

- нижнее – 20 мм.

Абзацы в тексте начинают отступом, равным пяти ударам клавиатуры персонального компьютера или 1 см.

Текст основной части работы делится на главы, разделы, подразделы, пункты.

Заголовки структурных частей работы («Содержание», «Введение», «Основная часть», «Заключение», «Список использованных источников», «Приложение») следует выполнять по центру с прописной буквы без точки в конце, не подчеркивая.

Расстояние между заголовками и текстом должно быть равно одному межстрочному интервалу. Каждую структурную часть (главу) работы следует начинать с нового листа.

Табличный и графический материал, приводимый для иллюстрации отдельных положений, следует снабжать ясными заголовками и нумеровать. При оформлении таблиц:

- название таблицы (заголовок) должно соответствовать ее содержанию и отвечать на три вопроса одновременно: что, где, когда;

- наименование «Таблица» располагается перед заголовком в левой части страницы, знак «№» не ставится, например: Таблица 2.1, если таблица имеет название, то его помещают через пробел и тире с заглавной буквы, точка в конце наименования не ставится;

- единица измерения, если она едина для всех показателей, указывается после заголовка таблицы через запятую, например «млн. руб.», «%»; при разной размерности единицы измерения показателей таблицы указываются в заголовках соответствующих граф таблицы через запятую.

Все таблицы должны быть удобно обозреваемыми, органически связанными с текстом, полностью соответствовать требованиям статистики и, как правило, не занимать более одной страницы.

Большие таблицы, содержащие более десяти строк или восьми колонок - граф, следует выносить в приложения. Перенос на другую страницу небольших и средних таблиц не рекомендуется.

При ссылке в тексте на использованные источники следует приводить порядковые номера по списку использованных источников, заключенные в квадратные скобки, например: «... как указано в монографии [10]», «... в работах [11,12,15–17]».

Библиографическое описание литературных источников производится в соответствии с ГОСТ. Стандарт приведен в разделе 6.2 дополнительной литературы [6] и располагается в следующей последовательности:

- фамилия и инициалы автора (после фамилии);

- точное название работы (по титульному листу);

- место издания (приводится полностью в именительном падеже, за исключением названий городов Москва – М., Санкт-Петербург – СПб.);

- название издательства (или издающей организации);

- год издания (только цифра без буквы «г»);

- страницы.

При описании журнальных и газетных статей место издания и название издательства не указываются. В многотомных изданиях номер тома (или части) ставится после года издания, например: «…1994. – Т.2. - …»; «…1994. – 4.1. - …»; «…1994. – Вып.3…». Список действительно использованной литературы приводится в конце работы.

Нумерация работы, начиная с титульного листа, сплошная, и выполняется арабскими цифрами в верхнем правом углу страницы (без точек и черточек). При этом титульный лист считается первым, но не нумеруется.

Состав работы включает:

- титульный лист;

- задание на работу;

- оглавление (или содержание);

- текст работы, подразделяющийся на главы и параграфы;

- заключение (выводы и предположения);

- приложения и библиография.

**Титульный лист** имеет единую форму и реквизиты.

**Оглавление** (или содержание). В нем последовательно указывается наименование частей работы (введение, название глав и входящих в них параграфов, заключение, приложения, список использованной литературы). Против каждого наименования в правой стороне листа указывается номер страницы, с которой начинается данная часть работы.

**Введение**. Формируются цель и основные задачи (цель работы всегда одна, а задач столько, сколько требуется для достижения этой цели), даются пояснения к избранному плану и содержанию работы; чем обусловлена принятая структура, какие методы обработки использованы и так далее.

**Основная часть**. Посвящается описанию выполненной работы. Приводится описание используемых классов, алгоритмов. При необходимости этот раздел может быть разбит на главы и параграфы. Перед названием главы и параграфов пишутся их номера арабскими цифрами. Причем знак параграфа не ставится, вместо него указывается через точку номер главы и параграфа, в первой главе – 1.1; 1.2; во второй – 2.1; 2.2 и т.д. В тексте работы название глав и параграфов следует выделять соответствующими интервалами, исполнять заглавия разделов более крупными буквами. Каждый раздел работы, кроме параграфов, следует начинать с новой страницы. Выравнивание текста – по центру.

**Заключение** – это резюме всей работы.

# 5. САМОСТОЯТЕЛЬНАЯ РАБОТА

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | № раздела дисциплины из табл. 5.1 | Тематика самостоятельной работы  *(детализация)* | ОК, ПК | Контроль выполнения работы  (Опрос, тест, дом. задание, и т.д.) |
| 1. | 1÷7 | Проработка лекционного материала | ПК-1;  ПК-3 | Опрос на занятиях (устно) |
| 2. | 2, 3, 4, 6 | Подготовка к лабораторным работам | ПК-1;  ПК-3 | Отчёт,  защита лабораторных. работ |
| 4. | 3, 5 | Самостоятельное изучение тем теоретической части | ПК-1;  ПК-3 | Дом. задание, тест |

***Темы для самостоятельного изучения***

* + 1. Организация обучения.
    2. Виды планов. Декомпозиция планов.
    3. Организация поддержки программных изделий.
    4. Организация сопровождения программных изделий.
    5. Организация планирования в фазах оценки и использования.
    6. Организационная структура группы выпуска документации. Стандарты и практические руководства.
    7. Организационная структура группы обслуживания.
    8. Современное состояние методов обеспечения качества программного изделия.
    9. Организация планирования в фазе исследований.
    10. Организация планирования в фазе осуществимости.
    11. Организация выпуска документации в фазах оценки и использования.
    12. Участие группы выпуска документации в фазовых обзорах.

# 6 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

## 6.1 Основная литература

1. Технология разработки программного обеспечения: Учебное пособие / Калайда В. Т., Романенко В. В. – 2012. 220 с. [Электронный ресурс] - Научно-образовательный портал ТУСУР – 2012. – Режим доступа: <http://edu.tusur.ru/training/publications/2076>

## 6.2 Дополнительная литература

1. Калайда В. Т. Технология разработки программного обеспечения : Учебное пособие / В. Т. Калайда, В. В. Романенко; Федеральное агентство по образованию, Томский государственный университет систем управления и радиоэлектроники. - Томск: ТУСУР, 2007. - 237[1] с. (90 экз.)**.**
2. [Брауде Э. Д.](file:///C:\publish\authors\21144\194723663\) Технология разработки программного обеспечения. ‑ СПб.: Питер, 2004. - 654 с. (22 экз.)**.**
3. Вендров А. М. Проектирование программного обеспечения экономических информационных систем. – М.: Финансы и статистика, 2002. – 176 с. (34 экз.)
4. [Орлов, С.А.](http://lib.tusur.ru/cgi-bin/irbis64r_01/cgiirbis_64.exe?Z21ID=&I21DBN=LIB&P21DBN=LIB&S21STN=1&S21REF=1&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%9E%D1%80%D0%BB%D0%BE%D0%B2,%20%D0%A1%D0%B5%D1%80%D0%B3%D0%B5%D0%B9%20%D0%90%D0%BB%D0%B5%D0%BA%D1%81%D0%B0%D0%BD%D0%B4%D1%80%D0%BE%D0%B2%D0%B8%D1%87) Технологии разработки программного обеспечения. Разработка сложных программных систем : Учебное пособие для вузов / Сергей Александрович Орлов. - СПб. : Питер, 2002. - 464 с. ( 26 экз.)+
5. [Елизаров, А.И.](http://lib.tusur.ru/cgi-bin/irbis64r_01/cgiirbis_64.exe?Z21ID=&I21DBN=LIB&P21DBN=LIB&S21STN=1&S21REF=10&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%95%D0%BB%D0%B8%D0%B7%D0%B0%D1%80%D0%BE%D0%B2,%20%D0%90%D0%BB%D0%B5%D0%BA%D1%81%D0%B5%D0%B9%20%D0%98%D0%B3%D0%BE%D1%80%D0%B5%D0%B2%D0%B8%D1%87)  Технология разработки программного обеспечения : учебное методическое пособие для студентов специальности 230105 / А. И. Елизаров, В. В. Романенко ; Федеральное агентство по образованию, Томский государственный университет систем управления и радиоэлектроники, Кафедра автоматизированных систем управления. - Томск : ТМЦДО, 2007. - 119 с. ( 8 экз.) +

## 6.3 Перечень пособий, методических указаний и материалов, используемых в учебном процессе

* 1. [Елизаров А.И. Методические рекомендации к лабораторным занятиям и самостоятельной работе по дисциплине «Технология разработки программного обеспечения» для студентов специальности 080801 «Прикладная информатика (в экономике)» / Томск: ТУСУР, 2012. - 9 с.](http://asu.tusur.ru/learning/spec080801/d60/s080801_d60_labs.pdf)

<http://asu.tusur.ru/learning/spec080801/d60/>

## 6.4 Базы данных, информационно-справочные и поисковые системы

1. [www.compress.ru](http://www.compress.ru) – Журнал «КомпьютерПресс»

2. [www.osp.ru](http://www.osp.ru) – Издательство «Открытые системы»

3. [www.cnews.ru](http://www.cnews.ru) – Издание о высоких технологиях

4. [www.it-daily.ru](http://www.it-daily.ru) – Новости российского ИТ-рынка

5. [www.isn.ru](http://www.isn.ru) – Российская сеть информационного общества

# 7 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для проведения теоретического материала (лекций) и лабораторных занятий по дисциплине используются персональный ПК с процессором Pentium 4, операционная система MS Windows ХР, пакет Microsoft Office 2007. Лекции и практические занятия осуществляются в специализированной аудитории с проектором, экраном, на который слайды демонстрации проецируются.