Федеральное государственное бюджетное образовательное учреждение высшего образования
«КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Л. Г. Высоцкий

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

Учебно-методическое пособие по изучению дисциплины
для студентов, обучающихся в бакалавриате по направлению подготовки
09.03.01 Информатика и вычислительная техника

Калининград
Изд-во ФГБОУ ВО «КГТУ»
2022

УДК 004.9(075)

Рецензент:
кандидат педагогических наук, доцент кафедры прикладной информатики
ФГБОУ ВО «Калининградский государственный технический
университет» Е. Ю. Заболотнова

Высоцкий, Л. Г.
Программная инженерия: учебно-методическое пособие по изучению дисциплины для студентов, обучающихся в бакалавриате по направлению подготовки 09.03.01 Информатика и вычислительная техника / **Л. Г. Высоцкий.** – Калининград: Изд-во ФГБОУ ВО «КГТУ», 2022. – 28 с.

Учебно-методическое пособие включает тематический план с расчасовкой и перечень лабораторных работ, которые должны быть выполнены студентами при прохождении дисциплины «Программная инженерия». Все работы выполняются по индивидуальным заданиям. В пособие также включен список рекомендуемой литературы по дисциплине.

Учебно-методическое пособие рассмотрено и одобрено в качестве локального электронного методического материала кафедрой прикладной информатики института цифровых технологий ФГБОУ ВО «Калининградский государственный технический университет» 19 сентября 2022 г., протокол № 3.

Учебно-методическое пособие рекомендовано к использованию в качестве локального электронного методического материала в учебном процессе методической комиссией ИЦТ 20 сентября 2022 г., протокол № 6.

УДК 004.9 (075)

Оглавление

Введение

Данное учебно-методическое пособие предназначено для студентов направления подготовки 09.03.01 Информатика и вычислительная техника, изучающих дисциплину «Высокоуровневые технологии программирования».

Цель освоения дисциплины:

Целью освоения дисциплины «Программная инженерия» является формирование знаний и навыков по практическому использованию основных направлений, концепций, технологий, методик и стандартов в области создания программных средств как продуктов промышленного производства.

Для успешного освоения дисциплины в соответствии с учебным планом ей предшествуют дисциплины «Информатика», «Программирование», «Информационные технологии», «Высокоуровневые технологии программирования», «Математическая логика и теория алгоритмов», «Операционные системы».

Далее в пособии представлен тематический план, содержащий перечень изучаемых тем, обязательных лабораторных работ, мероприятий текущей аттестации и отводимое на них аудиторное время (занятия в соответствии с расписанием) и самостоятельную работу.

В разделе Содержание дисциплины приведены подробные сведения об изучаемых вопросах, по которым студент может ориентироваться в случае пропуска каких-то занятий, а также методические рекомендации преподавателя для самостоятельной подготовки, каждая тема имеет ссылки на литературу (или иные информационные ресурсы), а также контрольные вопросы для самопроверки.

Раздел «Текущая аттестация» содержит описание обязательных мероприятий контроля самостоятельной работы и усвоения разделов или отдельных тем дисциплины. Далее изложены требования к завершающей аттестации – экзамену.

В разделе «Балльно-рейтинговая система» приведен порядок применения балльно-рейтинговой системы контроля успеваемости.

Помимо данного пособия, студентам следует использовать материалы, размещенные в соответствующем данной дисциплине разделу ЭИОС, в которые более оперативно вносятся изменения для адаптации дисциплины под конкретную группу.

Техническое обеспечение дисциплины составляют:

1. Операционная система Windows 7 (получаемая по программе Microsoft "Open Value Subscription");
2. Офисное приложение MS Office Standard 2010 (получаемое по программе Microsoft "Open Value Subscription");
3. Kaspersky Endpoint Security;
4. Google Chrome;
5. Python.

Тематический план

	Раздел (модуль) дисциплины	Тема	Объем аудиторной работы, ч	Объем самост-ной работы, ч
		Лекции		
1.1	Базовые понятия программной инженерии	Базовая терминология программной инженерии	2	0,5
1.2		Проблематика создания ПО на современном этапе	2	0,5
2.1	Жизненный цикл (ЖЦ) программного обеспечения	Основные понятия жизненного цикла	2	0,5
2.2		Постановка задачи на проектирование. Планирование разработки	2	0,5
2.3		Формирование и анализ требований к программной системе	4	0,5
2.4		Проектирование программной системы	4	1
2.5		Кодирование программной системы	2	0,5
2.6		Отладка и тестирование программной системы	2	1
2.7		Поставка и сопровождение программной системы	2	0,5
3.1	Технологические процессы разработки ПО	Управление процессом разработки	4	1
3.2		Управление качеством ПО	2	0,5
3.3		Управление рисками в процессе разработки ПО	2	0,5
			30	**7,5**

		Лабораторные занятия		
1.1		ЛР 1. Планирование разработки программной системы	3	3
1.2		ЛР 2. Использование метода экспертных оценок при разработке программных систем	3	3
1.3		ЛР 3. Формирование информационной модели предметной области	4	3
1.4		ЛР 4. Средства формирования поведенческой модели	4	4
1.5		ЛР 5. Алгоритмизация «задачи коммивояжера»	3	3
1.6		ЛР 6. Разработка спецификаций программного обеспечения	3	3
1.7		ЛР 7. Изучение стандартов	3	3
1.8		ЛР 8. Управление рисками для программной системы	3	3
1.9		ЛР 9. Тестирование программного обеспечения	4	5
			30	**30**

		Курсовая работа		
1.1	Формирование структуры графического пользовательского интерфейса	Контрольная точка 1. Раздел проекта 1	1,8	8
1.2	Формирование функционала программного приложения и его привязка к интерфейсу	Контрольная точка 2. Раздел проекта 2	2	12
		Оформление работы. Защита	1	3,5
			4,8	**23,5**

		Рубежный (текущий) и итоговый контроль		
1.1	Рубежный контроль	Лекция № 1	1	
1.2	Рубежный контроль	Лекция № 2	1	
1.3	Рубежный контроль	Лекция № 3	1	
1.4	Рубежный контроль	Лекция № 4	1	
1.5	Рубежный контроль	Лекция № 5	1	
1.6	Рубежный контроль	Лекция № 6	1	
1.7	Рубежный контроль	Лекция № 7	1	
1.8	Рубежный контроль	Лекция № 8	1	
1.9	Рубежный контроль	Лекция № 9	1	
1.10	Рубежный контроль	Лекция № 10	1	
1.11	Рубежный контроль	Лекция № 11	1	
1.12	Рубежный контроль	Лекция № 12	1	
1.13	Рубежный контроль	Лекция № 13	1	
1.14	Рубежный контроль	Лекция № 14	1	
1.15	Рубежный контроль	Лекция № 15	1	
1.16	Итоговый контроль	Экзамен	0	39,2
			15	**39,2**
		Всего	**79,8**	**100,2**

Содержание дисциплины и указания к изучению

3.1 Раздел 1. Базовые понятия программной инженерии

3.1.1 Тема 1.1 Базовая терминология программной инженерии

Перечень изучаемых вопросов

Понятие программы, программного продукт, программного изделия. программирования. Определение программной инженерии, методологии и технологии программирования. Технологическая операция как основа технологии. Понятие стандарта, роль стандартизации в промышленном производстве программного обеспечения. Классификация стандартов, используемых при подготовке программных продуктов.

Методические указания к изучению

Программная инженерия является молодой наукой, её терминология еще окончательно не сложилась и продолжает бурно эволюционировать. Поэтому надо учитывать возможную неоднозначность в интерпретации некоторых определений разными авторами в разных литературных источниках по программной инженерии, особенно если эти источники разнесены во времени на несколько лет.

Необходимо сразу уяснить важнейшую роль стандартизации процесса создания ПО, освоить базовые стандарты и использовать их студентами уже в ходе учебного процесса. Прежде всего, для формирования отчетов по лабораторным работам и пояснительной записки по курсовой работе. Более подробно и целенаправленно роль стандартизации на разных стадиях разработки ПО рассматривается в лабораторной работе «Изучение стандартов».

Литература

1. 1.1 Введение в программную инженерию, с. 5 – 9.
6. 1.1. Введение в программную инженерию, с. 5 - 20.

Контрольные вопросы

1. Сколько лет программирование носит массовый характер? (С какого времени программирование стало массовым?)
2. Что первично: алгоритм или программа?
3. Алгоритм является компьютерно-ориентированным?
4. Что стоит слева при описании технологической операции?
5. Какое требование предъявляется к оформлению входной и выходной информации технологической операции?
6. Почему документация проектирования программного обеспечения должна оформляться на основе стандартов?
7. Что такое «**метасреда»** для программного изделия?
8. Как в настоящее время называется процесс создания больших программных систем коллективами разработчиков?
9. Назовите основные элементы методологии.
10. Что такое «**стандарт**»?

3.1.2 Тема 1.2 Проблематика создания ПО на современном этапе

Перечень изучаемых вопросов

Кризис программирования. Признаки кризиса: отставание от графика, превышение сметы расходов, низкое качество. Причины кризиса. Качественные изменения процесса разработки ПО. Уникальность программных проектов. Сложность программных проектов. Влияние рыночной экономики. Требования к профессиональному разработчику ПО.

Методические указания к изучению

Необходимо обратиться к истории развития программной инженерии и понять причины кризиса программирования в середине 60-х годов прошлого столетия. Резко возросшая сложность программных продуктов при использовании методов индивидуальной разработки привели к значимому ухудшению их основных параметров. Это потребовало разработки научных методов конструирования больших программных продуктов. Современный специалист в программной инженерии должен уметь работать в коллективе, использовать системный подход к разработке, знать научные методики и технологии создания ПО.

Литература

1. 1.1 Введение в программную инженерию, с. 10 – 19.
2. Тема 9. Проектирование программных систем. с. 14 – 19.

Контрольные вопросы

1. Назовите основные причины кризиса программирования в 60-х годах 20-го столетия.
2. В чем выразился кризис программирования в 60-х годах 20-го столетия?
3. Почему терминология программной инженерии до сих пор не устоялась?
4. Какое событие привело к созданию программной инженерии?
5. В чем состоит принципиальное отличие программных проектов от технических проектов?
6. В чем трудность контроля и управления ходом разработки программных проектов?
7. Чему посвящена книга Брукса?
8. Какое свойство является основным у программных систем?
9. Почему перестает расти производительность группы разработчиков программной системы при переходе через некоторый порог?
10. Какие виды «зрения» должны быть у профессионального программиста?

3.2 Раздел 2. Жизненный цикл (ЖЦ) программного обеспечения

3.2.1. Тема 2.1 Основные понятия жизненного цикла

Перечень изучаемых вопросов

Жизненный цикл программной системы (ЖЦ ПС). Процесс анализа ПС. Процесс синтеза ПС. Процесс сопровождения ПС. Стадия. Этап создания ПС. Фаза ЖЦ ПС. Модель ЖЦ ПС. Каскадная модель. Итерационная модель. Спиральная модель. Распределение расходов по стадиям ЖЦ ПС. Модель CDM.

Методические указания к изучению

Понятие жизненного цикла было введено как научный ответ на кризис программирования. Оно является базовым для разработки программных систем. Жизненный цикл делится на временные промежутки, которые называются стадиями. Их отличительная особенность – предоставление в конце некоторого законченного результата. С другой стороны, стадии объединяются в более крупные структуры – фазы – анализа, синтеза, сопровождения. Необходимо внимательно проанализировать начальный состав стадий и сравнить его с текущим, что дает информацию о развитии программной инженерии за последние полвека. Последовательность стадий в жизненно цикле удобно отображать графически в виде модели. Исторически первой была последовательная модель, заимствованная из области проектирования технических систем. В ходе лекций необходимо критически разобрать все достоинства и недостатки данной модели и понять. почему она эффективна в плановой экономике. Такому же анализу надо подвергнуть спиральную модель и понять причины ее эффективности в рыночной экономике. В настоящее время многие крупные компьютерные фирмы следуют своим моделям ЖЦ, которые учитывают их специфику. Для примера можно рассмотреть модель SDM, используемую в фирме Oracle. Имеет смысл проанализировать распределение затрат на программный продукт по стадиям с учетом специфики последнего.

Литература

1. 2. Жизненный цикл программного продукта с. 20 – 46.
5. 2.3. Модели процесса разработки программного обеспечения, с. 35 – 62.

Контрольные вопросы

1. В какой модели жизненного цикла программной системы сложнее документировать этапы проектирования? Почему?
2. В какой модели жизненного цикла программной системы раньше можно определить экономическую целесообразность разработкой? Почему?
3. В какой модели жизненного цикла программной системы более широко можно использовать повторяющиеся проектные решения? Почему это выгодно?
4. Какая из рассмотренных моделей жизненного цикла программной системы предполагает более узкую специализацию участников разработки?
5. В какой модели жизненного цикла программной системы трудней реализовать контроль и управление разработкой? Почему?
6. Какая модель жизненного цикла программной системы наиболее полно соответствует рыночной экономике? Почему?
7. Какая модель жизненного цикла программной системы может учитывать интересы пользователя?
8. Что такое системы реального времени? В чем заключается специфика ее проектирования?
9. Какая стадия жизненного цикла программной системы является самой затратной?
10. Сколько стадий включает жизненный цикл программной системы по современным стандартам?

3.2.2. Тема 2.2 Постановка задачи на проектирование. Планирование разработки

Перечень изучаемых вопросов

Авторы идеи будущего программного продукта. Vision statement (видение будущего продукта). Задачи экспертного опроса. Вопросы экспертам. Менеджер по продуктам. Условия перехода к планированию. План. Планирование. Цель планирования. Задачи, решаемые в ходе

планирования. Ограничения на проект. Виды планов. Версии проекта. Цикличность планирования. Структура затрат на проект. Оценка затрат на проект. Метрики проекта. Процедура предварительной стоимости проекта.

Методические указания к изучению

Существуют два источника идей на разработку новых программных продуктов: производственная необходимость в некоторой фирме или предложение продукта, который, как предполагается, будет востребован на открытом рынке. В первом случае решение о создании программного приложения принимает топ-менеджмент фирмы-заказчика. Второй вариант обычно проходит этап экспертной оценки по уже устоявшемуся перечню вопросов. Положительное решение запускает процесс планирования будущей разработки, в ходе которого решается несколько десятков задач, связанных с оценкой потребных ресурсов, их распределением, разработкой календарных планов, назначением исполнителей на работы и т. д. В крупных программных проектах, кроме основного технологического плана, может создаваться ряд дополнительных планов: по управлению рисками, конфигурацией, качеством и т. д. Оценка различных параметров программного проекта требует использования соответствующих метрик. Исторически первыми были размерно-ориентированные метрики. Автоматизация программирования привела к появлению функционально ориентированных метрик.

В конечном итоге необходимо подчеркнуть, что планирование не носит разовый характер. Исходный план должен в ходе разработки с периодичностью в три-четыре недели анализироваться и, при необходимости, корректироваться.

Теоретический материал этой темы поддерживается лабораторными работами «Планирование разработки программной системы» и «Использование метода экспертных оценок при разработке программных систем».

В курсовой работе в русле рассмотренного материала студенту требуется сначала сформировать по своей теме vision statement и согласовать его с руководителем работы как руководящий документ всей дальнейшей разработки.

Литература

3. Планирование проекта. с. 287 – 290.
4. ПРОЕКТНОЕ ПЛАНИРОВАНИЕ, с. 108 – 134.

Контрольные вопросы

1. Перечислите содержимое плана сопровождения программной системы.
2. Какая работа проводится с планами на более поздних этапах разработки?
3. Дайте определение плана.
4. Какие задачи ставятся перед экспертами при оценке идеи программного проекта?
5. Как определяется потребность в заказном программном продукте?
6. На каких этапах разработки планирование проводится более интенсивно?
7. На какие вопросы должны ответить эксперты в процессе обсуждения идеи программного продукта?
8. По каким причинам эксперты могут отказать в реализации предлагаемому программному проекту?
9. Укажите содержимое плана по управлению персоналом
10. Какие ресурсы требуются для реализации программных проектов?

3.2.3. Тема 2.3 Формирование и анализ требований к программной системе

Перечень изучаемых вопросов

Цель стадии «Формирование и анализ требований». Понятие проблемы. Анализ проблемы. Понятие предметной области. Системный анализ. Задачи системного анализа. ТЭО. Понятие модели. Модели «AS-IS». Модели «TO-BE». Информационная модель. Поведенческая модель. Функциональная модель. Модель SADT. Модель ДПС. Модели «сущность-связь». Модели UML. Понятие требования. Классификация требований. Пользователи требований. Языки описания требований (спецификаций).

Методические указания к изучению

В значительной степени эффективность проектирования программной системы и её качество на выходе определяются точностью и полнотой исходных требований к системе, что предопределяет важность этой стадии жизненного цикла. Обычно она разбивается на два этапа: системный анализ и собственно формирование требований. Задачей первого этапа является всесторонний и полный анализ автоматизируемого объекта. Для этого строится ряд моделей, описывающих определенные аспекты анализируемого объекта. Выделяют, прежде всего, функциональную, поведенческую и информационную модели. На этапе анализа они строятся в парадигме «AS-IS», т. е. описывают существующую на данный момент ситуацию. Существует целый ряд инструментариев для создания указанных моделей: SADT, ДПС, STD, ERD, UML и др.

На втором этапе формируются требования к проектируемой системе на основе проведенного анализа. Требования (в международной терминологии спецификации) обязательно должны представляться в формализованном виде, т. е. с использованием специализированных языков. В роли последних могут выступать и перечисленные ранее модели, но уже в парадигме «TO-BE», т. е. представлять цель проектирования. Переход от моделей «AS-IS к «TO-BE» и является решением проблемы, которая ставится перед созданием программной системы. Существует ряд и других языков спецификаций, вербальных и графических. Последние являются наиболее информационно-емкими.

Поскольку существует целый спектр пользователей спецификаций, отличающихся своими требованиями к последним, то и формируются в результате данного этапа несколько видов спецификаций, отличающихся своими характеристиками.

Теоретический материал этой темы иллюстрируется лабораторными работами «Формирование информационной модели предметной области» и «Средства формирования поведенческой модели».

Все рассмотренные в данной теме средства формирования требований к проектируемой системе обязательно используются для этой же цели и при выполнении курсовой работы по этой дисциплине.

Литература

2. 5. Определение требований к программному обеспечению, с. 71 – 111.
4. 6.4. Управление требованиями, с. 240 – 252.

Контрольные вопросы

1. Конечная цель создания разных моделей на этапе анализа стадии «Формирование и анализ требований»?
2. В чем состоит цель стадии «формирование и анализ требований»?
3. Что означает вариант построения моделей «AS-IS»?
4. Перечислите задачи стадии «формирование и анализ требований».
5. Какие модели создаются в процессе анализа предметной области разрабатываемой программной системы?
6. Что такое «предметная область» разрабатываемой программной системы?
7. Задачи системного анализа при проектировании программной системы?

8. Какой этап стадии «формирование и анализ требований» может отсутствовать?
9. В каких программных проектах отсутствует этап анализа предметной области?
10. Что входит в функциональные требования к проектируемой ПС.

3.2.4. Тема 2.4 Проектирование программной системы

Перечень изучаемых вопросов

Понятие проекта, проектирования. Основные параметры проекта. Структурный подход к проектированию. Принципы подхода. Метод пошаговой детализации. Этапы структурного подхода. Понятие подсистемы и модуля программной системы (ПС). Архитектура ПС. Проектирование архитектуры. Сущность модульного проектирования ПС. Модели управления в ПС. Специфика проектирования ПС при объектном подходе. Модели UML. Классификация структур данных. Проектирование структур данных. Классификация алгоритмов. Проектирование алгоритмов. Проектирование интерфейса пользователя: принципы проектирования. Средства поддержки пользователя. Документация пользователя.

Методические указания к изучению

Данная стадия является центральной в ЖЦ ПС, так как именно здесь идет формирование всех системных решений проекта на основе требований, представляющих результат предыдущей стадии. Исторически первым был структурный подход к проектированию программной системы, унаследованный от разработки технических проектов. Его суть – пошаговая декомпозиция решения проблемы вплоть до низовых функций, которые можно воплотить в листинге на языке реализации системы. Первым шагом для этого является формирование архитектуры ПС как совокупности взаимосвязанных подсистем и программных модулей. В рамках структурного подхода эффективна технология модульного программирования, позволяющая снизить трудозатраты и время разработки, повысить надежность ПС. В ходе всего проектирования необходимо помнить о специфических особенностях каждого программного проекта, прежде всего о его уникальности.

При объектном подходе проектирование представляет собой формирование набора графических моделей на языке UML, начатое еще на предыдущей стадии в процессе обследования объекта автоматизации, т. е. обе эти стадии можно рассматривать как единое целое.

На этапе стадии проектирования, посвященной алгоритмизации, необходимо акцентировать внимание на том, что одна и та же задача может быть выполнена несколькими разными алгоритмами. Поэтому разработчик должен уметь выбирать по определенным критериям (время выполнения, точность и т. д.) наиболее подходящий алгоритм. А для этого он должен оперировать как можно большим количеством известных алгоритмов.

В определенной степени это относится и к структурам данных, выбираемых для представления информации проекта. При том надо уяснить, что алгоритмы и структуры данных базируются на поведенческих и информационных моделях, формируемых на предыдущей стадии, а архитектура ПС – на функциональной модели.

На стадии проектирования также разрабатывается пользовательский интерфейс. Этот процесс должен базироваться на ряде принципов, нацеленных на создание «дружественного» к пользователю интерфейса. Например, рекомендуется использовать в нем только комфортные сочетания цветов текста.

В лабораторном практикуме работа «Разработка спецификаций программного обеспечения» нацелена на закрепление студентами методов формирования модульной структуры ПС, а работа «Алгоритмизация «задачи коммивояжера» - методов выбора наиболее эффективного алгоритма для решения конкретной задачи.

В ходе курсового проектирования решаются практически все задачи, входящие в стадию проектирования.

Литература

2. Тема 13. Конструирование программного обеспечения, с. 71 – 79.
4. Глава 7. АРХИТЕКТУРА И ПРОЕКТИРОВАНИЕ, с. 299 – 332.
5. 5. АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, с. 134 – 160.

Контрольные вопросы

1. Что возрастает с увеличением числа модулей программной системы?
2. На основании чего происходит выбор алгоритма разработчиком?
3. Дайте определение **структуры данных**.
4. Какие структуры данных называются физическими?
5. На основании какого компромисса, в конечном итоге, определяется количество модулей программной системы?
6. Что такое компонентное программирование?
7. На основе какой информации формируются структуры данных программной системы?
8. Укажите наиболее распространенные модели структурирования программных систем
9. Что такое **связность** модуля программной системы?
10. При какой связности модуль представляет «черный ящик»?

3.2.5. Тема 2.5 Кодирование программной системы

Перечень изучаемых вопросов

Понятие кодирования (программирования, реализации) программной системы. Нисходящее кодирование, восходящее кодирование. Стиль кодирования (программирования). Свойства кода. Принципы кодирования: KISS, EIBTI. Рекомендации по оформлению кода. Рефакторинг. Код систем реального времени. Защитное программирование: требования защитного программирования. Парное программирование: достоинства и недостатки.

Методические указания к изучению

Кодирование реализуется на основе спецификаций отдельных модулей, которые создаются на предыдущей стадии. Этот процесс достаточно индивидуален, он сродни искусству, поэтому производительность отдельных программистов может отличаться на порядок. Опыт их работы трудно обобщить и предать другим специалистам. Поэтому рекомендуется каждому программисту выработать свой стиль – набор приемов и методов для создания корректного, легкочитаемого кода. Этот стиль должен опираться на ряд принципов и рекомендаций, сформированных предыдущими поколениями, например, простота, концептуальная понятность кода, разделение кода на логические блоки с помощью неотображаемых символов и т. д. Следует учитывать, что для каждого класса программных систем, в силу их специфики, может существовать и свой набор рекомендаций. Классическим примером такого подхода являются системы реального времени, для которых рекомендуется использовать циклы только с числовыми параметрами, избегать рекурсий и т. д.

К настоящему времени устоялись правила оформления каждого программного модуля с целью более эффективного его повторного использования.

Существует также понятие защитного программирования, нацеленного на предотвращение ошибок времени исполнения кода. Оно предполагает максимальную

проверку входных данных, запрет деления больших чисел на малые, запрет сравнения вещественных чисел в логических выражениях и т. д.

Некоторые фирмы по разработке ПО используют также парное программирование, т. е. работу сразу двух программистов над одним и тем же кодом. Предполагая большие затраты на зарплату, такой подход обеспечивает более надежный код, т. е. снижение затрат на отладку, некритичность временного выбытия одного из разработчиков и другие достоинства.

Материал этой темы закрепляется студентами во время создания кода лабораторных работ курса, требующих программной реализации. А также во время формирования листинга курсовой работы.

Литература

10. 1.3.3 Конструирование ПО, с.37 – 39.
14. 2.2.3 Конструирование (кодирование), с.68 – 71.

Контрольные вопросы

1. Что такое защитное программирование?
2. Почему защитное программирование предполагает запрет на сравнение вещественных чисел в логических выражениях?
3. Почему защитное программирование предполагает запрет на деление больших чисел на малые?
4. В чем недостатки парного программирования?
5. Почему значительно снижается количество ошибок при парном программировании?
6. За счет чего растет квалификация программистов при парном программировании?
7. В каких программных системах вводятся жесткие ограничения на циклические конструкции?
8. Что такое стиль программирования?
9. Почему в системах реального времени жесткие ограничения на циклические конструкции?
10. Что гласит принцип KISS?

3.2.6. Тема 2.6 Тестирование программной системы

Перечень изучаемых вопросов

Тестирование. Цели тестирования. Зависимость стоимости исправления ошибки от стадии ЖЦ ПС. Классификация ошибок по причинам возникновения. Классификация ошибок по характеру проявления. Тест. Характеристики теста. Отладка. Этапы отладки. Методы отладки. Этапы работы с ошибкой. Функциональный подход к тестированию: достоинства и недостатки. Структурный подход к тестированию: достоинства и недостатки. Основные этапы тестирования всей ПС. Альфа-тестирование. Бета-тестирование. Принципы тестирования.

Методические указания к изучению

Сначала необходимо подчеркнуть двойственный характер тестирования: поиск и устранение ошибок в неисправных программах и проверка соответствия исходным данным в исправных программах. Соответственно разрабатываются для разных целей и разные виды тестов. Для работы с ошибками также, в общем случае, создаются два вида тестов: обнаруживающие и локализирующие ошибки. Сам процесс исправления ошибок строго стандартизирован, но не регламентирован по времени.

Различают два основных подхода к тестированию ошибок: функциональный (тестирование «черного ящика») и структурный (тестирование «белого (прозрачного) ящика»). У каждого из них есть свой объект тестирования и свои методики. Но необходимо

всегда помнить, что полное тестирование даже небольшого программного приложения практически невозможно. Поэтому разрабатываются различные стратегии тестирования, минимизирующие количество остающихся ошибок.

Накопленный к настоящему времени опыт в данной сфере программной инженерии аккумулирован в принципах тестирования, главный из которых гласит «Самая последняя ошибка всегда является предпоследней».

Материал этой темы закрепляется студентами во время выполнения лабораторной работы «Тестирование программного обеспечения», отладки кода других лабораторных работ курса, требующих программной реализации, а также во время формирования листинга курсовой работы.

Литература

2. 17.1. Основы тестирования, с.123 – 131.
4. Технологический процесс тестирования, с.192 – 194.
4. Глава 8. ТЕСТИРОВАНИЕ, с. 333 – 362.

Контрольные вопросы

1. Перечислите основные подходы к тестированию программных систем.
2. С каким «ящиком» ассоциируется функциональное тестирование?
3. С каким «ящиком» ассоциируется структурное тестирование?
4. В каком подходе к тестированию внутренняя структура программной системы не рассматривается, а только анализируются системные характеристики?
5. В каком подходе возможно исчерпывающее тестирование программной системы?
6. Что собой представляет стандарт тестирования С1?
7. Какой подход используется при интеграционном тестировании программной системы? Почему?
8. Какие ошибки обнаруживаются на этапе компиляции программных модулей?
9. Дайте определение понятию тестирование.
10. Укажите классификацию программных ошибок по характеру проявления.

3.2.7. Тема 2.7 Поставка и сопровождение программной системы

Перечень изучаемых вопросов

Варианты поставки. Состав поставки. Правила «хорошего тона» при поставке. Понятие сопровождения. Принцип Биледи. Задачи, решаемые при сопровождении. Виды сопровождения.

Методические указания к изучению

К настоящему времени на практике, в связи с развитием технологии интернета, появилось несколько вариантов поставки программного продукту пользователю, что приводит к разным наборам компонентов ПС в поставке. Наиболее полно комплект поставки представлен в коробочном варианте. В этом случае рекомендуется выполнить ряд правил «хорошего тона», выработанных за предыдущие десятилетия программной инженерии.

Эксплуатация и сопровождение ПС, как указывалось ранее, по статистке является самой затратной стадией ЖЦ ПС. Специалисты по сопровождению должны включиться в коллектив разработчиков с первых стадий ЖЦ, чтобы лучше знать работу ПС изнутри. К настоящему времени выработан перечень задач, которые стандартно решаются в процессе эксплуатации и сопровождения. Главной из них является усовершенствование работающей системы, поскольку в соответствии с принципом Биледи, ПС будет экономически эффективной только в том случае, если она постоянно развивается. В зависимости от степени

вмешательства также различают четыре варианта сопровождения: от незначительного изменения кода до полного реинжиниринга. В настоящее время граница раздела между собственно процессом разработки и сопровождением постоянно стирается, поскольку и в том, и в другом случае предусматривается одна и та же последовательность этапов.

Литература

3. 25. Сопровождение программных систем, с.93 – 116.
4. 6.6. Управление конфигурациями и сопровождение. с.275 – 277.
10. 1.3.5 Сопровождение ПО с. 43 – 48.

Контрольные вопросы

1. Сколько существует правил хорошего тона при поставке программного продукта?
2. Какие варианты доставки программных продуктов покупателю существуют в настоящее время?
3. При каких вариантах доставки программного продукта не предполагается печатная документация?
4. При каких вариантах доставки программного продукта печатная документация обязательна?
5. Какой вариант доставки программного продукта включает рекламные материалы?
6. В чем заключается первое правило хорошего тона при доставке программного продукта?
7. Что гласит принцип Биледи?
8. Сколько классов задач решается в ходе сопровождения программной системы?
9. В чем заключается второе правило хорошего тона при доставке программного продукта?
10. Что предполагает сопровождение программного продукта?

3.3 Раздел 3. Технологические процессы разработки ПО

3.3.1. Тема 3.1 Управление процессом разработки

Перечень изучаемых вопросов

Проект. Управление проектом. Характеристики проекта. Специфика программного проекта. ограничения на проект. Задачи управления. Главный тезис управления. Задачи менеджера проекта. Управление персоналом. Организация групповой работы. организация рабочей среды. Управление готовностью персонала.

Методические указания к изучению

В управлении программными проектами главенствует тезис «Хорошее управление не гарантирует успешного завершения проекта, но плохое управление обязательно приведет к его провалу». Этот тезис подчеркивает всю важность процесса управления в структуре разработки программного продукта, тем более что данный процесс реализуется на протяжении всей длительности ЖЦ ПС. В больших проектах руководителей может быть двое: менеджер и технический лидер, в малых – обе роли может выполнять один человек. Задачи руководства достаточно многочисленны и охватывают как решение различных задач внутри коллектива разработчиков, так и контакт с внешним окружением. В коллективе менеджер должен создавать комфортные условия для коллективной работы специалистов, использовать механизмы стимулирования их труда, уделять внимание постоянному повышению квалификации разработчиков, вне коллектива менеджер обеспечивает контакты с вышестоящим начальством и заказчиками, занимается вопросами поступления финансовых средств.

Литература

1. 7. Управление программным проектом, с. 112 – 117.
4. 9.2. Управление программными проектами. Коллективный проект, с. 367 – 369.
13. Управление программными проектами

Контрольные вопросы

1. Какие обязанности относятся к менеджеру программного проекта по методологии MSF?
2. Что гласит главный тезис специалистов по управлению проектами?
3. Что такое «**гедонизм»**? Как его может использовать менеджер при работе над программным проектом?
4. Дайте определения понятию **управление**.
5. Перечислите задачи менеджеров, руководящих программными проектами?
6. Что такое когнитивная теория? Как ее может использовать менеджер при работе над программным проектом?
7. В чем состоит задача управления программным проектом?
8. Что такое потребность? Виды существуют групп потребностей у человека?
9. Что предполагает показатель **наблюдаемость** программного проекта?
10. В чем отличие программных проектов от других видов проектов?

3.3.2. Тема 3.2 Управление качеством ПО

Перечень изучаемых вопросов

Определение понятия качества. Факторы, влияющие на качество. «Рыбий скелет» К. Исикавы. Внутренне качество ПО. Внешнее качество ПО. Качество ПО при использовании. Метрики качества. Основные характеристики качества. Атрибуты характеристик качества ПО. Надежность ПО. Основные подходы по обеспечению надежности ПО. Методы определения надежности. Основные подходы для обеспечения качества ПО.

Методические указания к изучению

Одним из признаков «кризиса программирования», возникшего в 60-е годы прошлого столетия, было низкое качество выпускаемой программной продукции, не соответствующее исходным требованиям. Поэтому процесс управления качеством ПС должен брать свое начало с первых шагов ее создания и продолжаться весь ЖЦ. Разработаны десятки международных стандартов группы 9000, а также их отечественные аналоги, имеющие своей целью поддержку и регламентацию данного процесса. Определен перечень характеристик, по которым определяется уровень качества ПО, а также атрибуты этих характеристик. Необходимо стремиться к тому, чтобы все оценки характеристик носили количественный характер, что предполагает наличие соответствующих метрик. При этом метрики должны делиться и по вариантам качества: внутреннем, внешнем, при использовании системы. Важнейшей из характеристик качества является надежность, что обусловило создание соответствующих методов и подходов для ее обеспечения. Для упрощения оценки надежности и других характеристик качества ПС на выходе проектирования необходимо стремиться к как можно более строгому и точному формулированию исходных требований к ПС в начала ее разработки.

Литература

5. 1.4. Затраты, сроки и качество, с. 17 – 23.
4. 3.4. Метрика и качество кода, с. 123 – 142.

Контрольные вопросы

1. Какое количество ошибок допускает особо надежное ПО?
2. Какой стандарт определяет характеристики качества?
3. Что такое **качество** как общепринятое понятие?
4. Что входит в первую группу факторов, влияющих на качество программного продукта?
5. Что собой представляет «рыбий скелет» К. Исикавы?
6. Как распределяются факторы, влияющие на качество программного продукту по фазам жизненного цикла?
7. Какие виды метрик вводятся для оценки качества программного продукта?
8. Какая характеристика качества определяется как отношение полезного эффекта к произведенным затратам?
9. Что такое качество программной системы?
10. Что такое LOC? Для оценки какой характеристики качества ПО используется LOC?

3.3.3. Тема 3.3 Управление рисками в процессе разработки ПО

Перечень изучаемых вопросов

Понятие риска. Управление рисками. Схема управления рисками. Типы рисков. Анализ рисков. Планирование рисков. Мониторинг рисков. Определение рисков. Категории рисков. Источники рисков. Шкала вероятностей рисков. Показатель риска. Вероятность неудовлетворительного результата. Потери при неудовлетворительном результате. Стратегия управления значимым риском. Причины прекращения программного проекта из-за рисков.

Методические указания к изучению

Все те факторы, которые привели к «кризису программирования» в шестидесятые годы прошлого столетия, являются проявлениями тех или иных видов рисков, существующих в программной инженерии. Именно по этой причине специалист в данной области должен ясно представлять и предвидеть те нежелательные явления, которые могут возникнуть в ходе реализации программного проекта. В команде разработчиков должны быть персонально назначенные сотрудники, в функции которых входит управление рисками. Эта работа не должна носить разовый характер, а проводиться на всем протяжении программного проекта. Начальная оценка перечня рисков и их последствий должна периодически (например, через три недели) пересматриваться как по номенклатуре, так и по ущербу. Особое значение должно придаваться разработке эффективных стратегий как предотвращения рисков, так и их последствий.

Теоретический материал этой темы частично закрепляется в лабораторной работе «Информационная безопасность программных систем», в которой рассматривается стратегия минимизации потерь данных проекта путем управления доступа к ним. В курсовой работе также обязательно требуется создание парольного входа для управления доступом к обрабатываемой информации.

Литература

5. 4.4. Планирование управления рисками, с. 121 – 124.
3 22. Методы управления рисками в проекте, с. 65 – 78.
10. 2.2 Управление рисками проекта, с. 78 – 87.

Контрольные вопросы

1. К какому типу риска относится смена руководства компании-заказчика?
2. Каким рискам уделяется наибольшее внимание при разработке ПО?
3. В чем заключается работа с риском, если его нельзя предотвратить?
4. Какие виды ущерба рассматриваются при управлении рисками?
5. Что собой представляют бизнес-риски для программного проекта?
6. Что может представлять собой технологический риск?
7. В чем заключается планирование рисков?
8. Какие риски могут создать сами разработчики программного проекта?
9. Какие риски могут создать заказчики программного продукта?
10. Управление рисками это разовое мероприятие или постоянная деятельность на протяжении всего программного проекта?

ТРЕБОВАНИЯ К АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

Текущая аттестация

В ходе изучения дисциплины студентам предстоит пройти следующие этапы текущей аттестации:

1. Опрос после каждой лекции, представляющий ответ на пять вопросов. Правильный ответ на каждый вопрос добавляет балл к итоговой оценке.
2. Оценка вовремя выполненной курсовой работы по пятибалльной системе.
3. Наличие к началу сессии всех выполненных и защищенных лабораторных работ.

Условия получения положительной оценки

Завершающим этапом изучения дисциплины является промежуточная аттестация, представляющая собой:

Условие получения положительной оценки по БРС является наличие сданной курсовой работы и выполненный полностью к началу сессии лабораторный практикум. К оценкам, полученным в ходе опроса после каждой лекции, добавляется оценка за курсовую работу, после чего находится средняя оценка, которая округляется до ближайшего целого числа. Если студент не соглашается с полученной оценкой, он сдает экзамен по традиционной схеме.

Студенты, не выполнившие план учебной работы в течение семестра, а также при наличии ликвидации всех задолженностей, сдают экзамен по традиционной схеме.

Примерные вопросы к экзамену по дисциплине

1. Терминология программной индустрии.
2. Методология и технология программирования, классификация, определения, требования к технологии.
3. Виды используемых в технологии программирования стандартов, их назначение.
4. Основные проблемы программной инженерии на современном этапе.
5. Основные участники программного проекта.
6. Требования к профессиональному программисту.
7. Понятие жизненного цикла (ЖЦ), основные стадии ЖЦ программной системы (ПС).
8. Основные модели ЖЦ, их достоинства и недостатки, распределение затрат по стадиям ЖЦ.
9. Стадия ЖЦ ПС "Возникновение и исследование идеи".
10. Стадия ЖЦ ПС "Планирование проекта": определения, цели, решаемые задачи, виды ограничений.
11. Виды дополнительных планов при разработке ПС.
12. График разработки ПС: критический путь, резервы, виды оптимизации.
13. Документация программного проекта: классификация, функции.
14. Структура затрат на разработку программного проекта, оцениваемые параметры проекта.
15. Виды метрик, используемые при оценивании программного проекта.
16. Методика предварительного оценивания параметров программного проекта.
17. Стадия ЖЦ ПС "Формирование и анализ требований": задачи системного анализа проблемы.
18. Поведенческие модели, используемые при анализе проблемы, решаемой проектируемой ПС.
19. Функциональные модели, используемые при анализе проблемы, решаемой проектируемой ПС.
20. Модели ERD.
21. Спецификации. Языки описания требований к программному проекту. Пользователи спецификаций.
22. Понятие проекта и проектирования ПС, характеристики проекта, классификация.

23. Сущность и структура проектирования ПС при структурном подходе.
24. Архитектурное проектирование ПС, основные этапы, понятие подсистемы и модуля.
25. Модели структуры, модели управления в программной системе.
26. Модульная декомпозиция, понятие модуля, формирование модульной структуры ПС
27. Структуры данных ПС: определение, классификация
28. Алгоритмы: определение, классификация, рекомендации по проектированию.
29. Модели UML на этапе проектирования ПС при объектном подходе.
30. Модели UML на этапе реализации ПС при объектном подходе.
31. Графический интерфейс пользователя: структура, основные элементы, достоинства.
32. Структура проектирования интерфейса пользователя, принципы проектирования.
33. Ввод информации: основные формы ввода, рекомендации по организации компьютерного ввода/вывода.
34. Средства поддержки пользователя. Принципы, закладываемые в сообщения системы и справочную систему.
35. Справочная система ПС.
36. Документация пользователя, структура, пользователи.
37. Стадия ЖЦ ПС "Кодирование": основные рекомендации по формированию листинга.
38. Защитное и парное программирование.
39. Стадия ЖЦ ПС "Тестирование": задачи, виды ошибок, понятие теста, отладки.
40. Структура отладки программной системы и работы с ошибкой.
41. Функциональное тестирование: задачи, достоинства, недостатки.
42. Структурное тестирование: задачи, стратегии, достоинства, недостатки.
43. Этапы тестирования ПС.
44. Принципы тестирования программной системы.
45. Ввод программной системы в действие.
46. Стадия ЖЦ ПС "Эксплуатация и сопровождение": виды сопровождения, решаемые задачи.
47. Технологические процессы ЖЦ ПС: определение, структура в соответствии с международными стандартами.
48. Методология MSF: основные принципы.
49. Методология MSF: модель команды.
50. Методология MSF: модель процесса.
51. Управление проектом: определение, основные задачи и проблемы.
52. Руководство проектом: участники, задачи, виды менеджмента.
53. Руководство проектом: подбор персонала разработки ПС, его рабочая среда.
54. Оценивания уровня развития персонала разработки ПС.
55. Управление готовностью: определение, решаемые задачи, основные этапы.
56. Риски: определение, классификации, структура управления.
57. Риски: определение рисков.
58. Риски: анализ рисков.
59. Риски: планирование рисков.
60. Риски: мониторинг рисков.
61. Качество разрабатываемого ПС: определение, влияющие факторы.
62. Основные характеристики качества ПС.
63. Надежность программных продуктов: определение, подходы к обеспечению, методы определения надежности.
64. Формирование требований к качеству ПС.
65. Основные подходы к достижению качества ПС.
66. Сертификация программных продуктов.
67. Рекомендации по эффективному управлению программными проектами.
68. Оценка зрелости персонала.
69. Стандартизация: основные понятия.

70. Классификация стандартов.
71. Стандартизация в области программного обеспечения.

Заключение

УМП по изучению дисциплины **«Программная инженерия»** включает описание всех компонентов, требуемых для эффективного освоения данного предмета студентами направления подготовки 09.03.01 Информатика и вычислительная техника.

Литература

Основная литература:

1. Программная инженерия: учебное пособие / сост. Т. В. Киселева; Северо-Кавказский федеральный университет. – Ставрополь: Северо-Кавказский Федеральный университет (СКФУ), 2017. – Часть 1. – 137 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=467203 (дата обра-щения: 16.05.2022). – Библиогр. в кн. – Текст: электронный.
2. Программная инженерия: учеб. пособие / сост. Т. В. Киселева; Северо-Кавказский федеральный университет. – Ставрополь: Северо-Кавказский Федеральный университет (СКФУ), 2017. – Часть 2. – 100 с.: схем. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=494790 (дата обращения: 16.05.2022). – Текст: электронный.
3. Программная инженерия: [16+] / сост. Т. В. Киселева. – Ставрополь: Северо-Кавказский Федеральный университет (СКФУ), 2018. – Часть 3. – 130 с.: ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=563341 (дата обращения: 16.05.2022). – Библиогр.: с. 128. – Текст : электронный.
4. Романов, Е. Л. Программная инженерия: учебное пособие: [16+] / Е. Л. Романов; Новосибирский государственный технический университет. – Новосибирск: Новосибирский государственный технический университет, 2017. – 395 с. : табл., схем., ил. – (Учебники НГТУ). – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=573945 (дата обращения: 16.05.2022). – Библиогр. в кн. – ISBN 978-5-7782-3455-0. – Текст: электронный.
5. Абдулаев, В. И. Программная инженерия: учебное пособие: [16+] / В. И. Абдулаев. – Йошкар-Ола : Поволжский государственный технологический университет, 2016. – Часть 1. Проектирование систем. – 168 с.: схем., табл., ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=459449 (дата обращения: 16.05.2022). – Библиогр. в кн. – ISBN 978-5-8158- 1767-8 (ч. 1); ISBN 978-5-8158- 1766-1. – Текст: электронный.
6. Антамошкин, О. А. Программная инженерия. Теория и практика: учебник/ О. А. Антамошкин; Сибирский федеральный университет. – Красноярск: Сибирский федеральный университет (СФУ), 2012. – 247 с. : ил., табл., схем. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=363975 (дата обращения: 16.05.2022). – Библиогр.: с. 240. – ISBN 978-5-7638-2511-4. – Текст : электронный.
7. Перл, И. А. Введение в методологию программной инженерии: учебное пособие: [16+] / И. А. Перл, О. В. Калёнова. – Санкт-Петербург: Университет ИТМО, 2019. – 53 с.: ил., схем. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=566776 (дата обращения: 16.05.2022). – Бибиогр. в кн. – Текст : электронный.

8. Мейер, Б. Объектно-ориентированное программирование и программная инженерия: учеб. пособие: [16+] / Б. Мейер. – 2-е изд., испр. – Москва: Национальный Открытый Университет «ИНТУИТ», 2016. – 286 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=429034 (дата обращения: 16.05.2022). – Текст: электронный.

9. Соловьев, Н. А. Введение в программную инженерию: учебное пособие / Н. А. Соловьев, Л. А. Юркевская; Оренбургский государственный университет. – Оренбург: Оренбургский государственный университет, 2017. – 112 с.: схем., табл., ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=481815 (дата обращения: 16.05.2022). – Библиогр.: с. 83. – ISBN 978-5-7410-1685-5. – Текст: электронный.

10. Ехлаков, Ю. П. Введение в программную инженерию: учебное пособие / Ю. П. Ехлаков; Томский Государственный университет систем управления и радиоэлектроники (ТУСУР). – Томск: Томский государственный университет систем управления и радиоэлектроники, 2011. – 148 с. : табл., схем. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=209001 (дата обращения: 16.05.2022). – ISBN 978-5-4332-0018-0. – Текст: электронный.

Дополнительная литература:

11. Ехлаков, Ю. П. Экономика программной инженерии: учеб. пособие / Ю. П. Ехлаков; Томский Государственный университет систем управления и радиоэлектроники (ТУСУР). – Томск: Эль Контент, 2013. – 132 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=480604 (дата обращения: 16.05.2022). – Библиогр.: с. 124-125. – ISBN 978-5-4332-0126-2. – Текст: электронный.

12. Программная инженерия: лабораторный практикум: практикум: [16+] / Д. Г. Лагерев, Д. А. Коростелев, А. А. Азарченков, Е. В. Коптенок. – Москва; Берлин: Директ-Медиа, 2021. – 157 с.: табл., ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=602232 (дата обращения: 16.05.2022). – Библиогр. в кн. – ISBN 978-5-4499-2105-5. – Текст: электронный.

13. Ехлаков, Ю. П. Управление программными проектами: учебник / Ю. П. Ехлаков; Томский Государственный университет систем управления и радиоэлектроники (ТУСУР). – Томск: Томский государственный университет систем управления и радиоэлектроники, 2015. – 217 с. : схем., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=480634 (дата обращения: 16.05.2022). – Библиогр. в кн. – ISBN 978-5-86889-723-8. – Текст: электронный.

14. Мякишев, Д. В. Принципы и методы создания надежного программного обеспечения АСУТП: учеб. пособие: [16+] / Д. В. Мякишев. – 2-е изд. – Москва; Вологда: Инфра-Инженерия, 2021. – 116 с. : ил., табл., схем. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=617225 (дата

обращения: 16.05.2022). – Библиогр. в кн. – ISBN 978-5-9729-0674-1. – Текст: электронный.
15. Ехлаков, Ю. П. Планирование и организация вывода программного продукта на рынок: учеб. пособие: [16+] / Ю. П. Ехлаков. – Томск: ТУСУР, 2017. – 121 с.: ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=481009 (дата обращения: 16.05.2022). – Библиогр.: с. 115-117. – ISBN 978-5-4332-0258-0. – Текст: электронный.
16. Петрухин, В. А. Методы и средства инженерии программного обеспечения: курс: учеб. пособие / В. А. Петрухин, Е. М. Лаврищева; Национальный Открытый Университет "ИНТУИТ". – Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2008. – 424 с. : табл., схем. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=234553 (дата обращения: 16.05.2022). – Текст: электронный.
17. Ехлаков, Ю. П. Управление программными проектами: учеб. пособие/ Ю. П. Ехлаков; Томский Государственный университет систем управления и радиоэлектроники (ТУСУР). – Томск: Эль Контент, 2014. – 140 с.: схем., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=480462 (дата обращения: 16.05.2022). – Библиогр.: с. 128-130. – ISBN 978-5-4332-0163-7. – Текст: электронный.
18. Москвитин, А. А. Решение задач на компьютерах: учеб. пособие:/ А. А. Москвитин. – Москва; Берлин : Директ-Медиа, 2015. – Часть 2. Разработка программных средств. – 429 с.: ил., схем., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=273667 (дата обращения: 16.05.2022). – Библиогр. в кн. – ISBN 978-5-4475-3646-6. – DOI 10.23681/273667. – Текст: электронный.
19. Соловьев, Н. Системы автоматизации разработки программного обеспечения: учеб. пособие / Н. Соловьев, Е. Чернопрудова; Оренбургский государственный университет. – Оренбург: Оренбургский государственный университет, 2012. – 191 с.: ил., схем., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=270302 (дата обращения: 16.05.2022). – Библиогр.: с. 182-183. – Текст: электронный.
20. Кайгородцев, Г. И. Введение в курс метрической теории и метрологии программ: учебник/ Г. И. Кайгородцев. – Новосибирск: Новосибирский государственный технический университет, 2011. – 190 с. : табл., схем., ил. – (Учебники НГТУ). – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=435984 (дата обращения: 16.05.2022). – Библиогр. в кн. – ISBN 978-5-7782-1648-8. – Текст: электронный.
21. Дроздов, С. Н. Структуры и алгоритмы обработки данных: учебное пособие: [16+] / С. Н. Дроздов. – Таганрог: Южный федеральный университет, 2016. – 228 с.: схем., ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=493032 (дата обращения: 16.05.2022). – Библиогр. в кн. – ISBN 978-5-9275-2242-2. – Текст: электронный.

22. Нехорошкова, Л. Г. Информационное моделирование и анализ требований : учеб. пособие: [16+] / Л. Г. Нехорошкова ; Поволжский государственный технологический университет. – Йошкар-Ола: Поволжский государственный технологический университет, 2020. – 146 с. : ил., табл., схем. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=615678 (дата обращения: 16.05.2022). – Библиогр.: с. 113-114. – ISBN 978-5-8158-2209-2. – Текст: электронный.
23. Зыкина, А. В. Методы принятия оптимальных решений: учеб. пособие: [16+] / А. В. Зыкина, О. Н. Канева, Т. Ю. Финк; Омский государственный технический университет. – Омск : Омский государственный технический университет (ОмГТУ), 2020. – 178 с.: ил., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=683053 (дата обращения: 16.05.2022). – Библиогр. в кн. – ISBN 978-5-8149-3175-7. – Текст: электронный.
24. Программирование и основы алгоритмизации: учеб. пособие / В. К. Зольников, П. Р. Машевич, В. И. Анциферова, Н. Н. Литвинов; Федеральное агентство по образованию, Воронежская государственная лесотехническая академия. – Воронеж: Воронежская государственная лесотехническая академия, 2011. – 341 с.: ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=142309 (дата обращения: 29.05.2022). – Текст: электронный.
25. Князьков, В. С. Введение в теорию автоматов / В. С. Князьков, Т. В. Волченская. – Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2008. – 78 с. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=234134 (дата обращения: 29.05.2022). – Текст: электронный

Локальный электронный методический материал

Леонид Григорьевич Высоцкий

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

Редактор Г. А. Смирнова

Уч.-изд. л. 1,75. Печ. л. 1,75.

Издательство федерального государственного бюджетного образовательного учреждения высшего образования «Калининградский государственный технический университет».
236022, Калининград, Советский проспект, 1