

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
БУРЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ ДОРЖИ БАНАЗАРОВА  
ИНСТИТУТ МАТЕМАТИКИ, ФИЗИКИ И КОМПЬЮТЕРНЫХ НАУК  
КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И ИНФОРМАТИКИ

Утверждена на заседании  
Ученого совета ИМФКН  
«\_\_»\_\_\_\_\_ 202\_\_ г.  
Протокол № \_\_

**Рабочая программа дисциплины**  
**Разработка программных приложений**

Направление подготовки / специальность  
**09.04.02 Информационные системы и технологии**

Профиль  
**Проектирование, разработка и эксплуатация информационных систем**

Квалификация (степень) выпускника  
**Магистр**

Форма обучения  
**Очная**

Улан-Удэ  
2025

# Пояснительная записка

## Цели освоения дисциплины

Цель изучения курса "Разработка программных приложений" - сформировать у студентов практические навыки по разработке программного обеспечения для решения экономических и расчетных задач с применением современных методов и технологий программирования

## Место дисциплины в структуре образовательной программы

Данная дисциплина входит в блок Б1.О для направления подготовки 09.04.02

Информационные системы и технологии. Полученные при изучении предмета умения и навыки дают возможность овладеть основами теоретических и практических знаний в области создания программных приложений.

**В результате освоения дисциплины студент должен:**

## Планируемые результаты обучения по дисциплине и индикаторы достижения компетенций.

ОПК-5. Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем

ОПК-5.2. Модернизирует программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач

ОПК-6. Способен использовать методы и средства системной инженерии в области получения, передачи, хранения, переработки и представления информации посредством информационных технологий

ОПК-6.2. Проводит анализ современных методов и средств информатики для решения прикладных задач различных классов

ОПК-8. Способен осуществлять эффективное управление разработкой программных средств и проектов

ОПК-8.2. Осуществляет выбор средств разработки, оценивает сложность проектов, планирует ресурсы, контролирует сроки выполнения и оценивает качество полученного результата

## Знать:

базовые понятия, лежащие в основе процесса разработки программных приложений; основные модели жизненного цикла программных средств.

## Уметь:

выбирать стратегию разработки программных средств и систем; строить диаграммы моделирования в языке UML.

## Владеть:

навыками объектно-ориентированного анализа и проектирования; навыками проектирования программного обеспечения в современных средах разработки.

## Планируемые результаты освоения образовательной программы:

## Объем дисциплины в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины составляет 3 зачетные единицы, 108 часа.

№ Название разделов дисциплины	Лекция	Лабораторная работа	Самостоятельная работа
Семестр 2	12	22	74
1 Разработка программных приложений	6	12	32

№	Название разделов дисциплины	Лекция	Лабораторная работа	Самостоятельная работа
2	Классические методологии разработки программных приложений	6	10	42

## Тематическое планирование курса

### Темы

#### Разработка программных приложений

Семестр 2

##### Стратегии разработки программных приложений

Лекция. 1(0) ч. Жизненный цикл программного обеспечения

Лабораторная работа. 2(0) ч. Выявление требований к программной системе. Работа с заказчиком

Лекция. 1(0) ч. Каскадная стратегия. Инкрементная стратегия. Эволюционная стратегия

Лабораторная работа. 4(0) ч. Обследование системы, общение с заказчиком, планирование разработки, составление технического задания

Самостоятельная работа. 14(0) ч. Сложность реальной предметной области, сложность описания поведения больших дискретных систем, сложность управления коллективом разработчиков

##### Модели жизненного цикла программных средств

Лекция. 2(0) ч. Модели жизненного цикла, реализующие каскадную стратегию разработки программных средств

Лабораторная работа. 2(0) ч. Классическая каскадная модель. Каскадная модель с обратными связями.

Самостоятельная работа. 4(0) ч. V-образная модель

Лекция. 2(0) ч. Модели жизненного цикла, реализующие инкрементную стратегию разработки программных средств. Модель с уточнением требований на начальных этапах разработки. Модели жизненного цикла, реализующие эволюционную стратегию разработки программных средств. Структурная эволюционная модель быстрого прототипирования. Спиральная модель Боэма

Лабораторная работа. 4(0) ч. Модели быстрой разработки приложений

Самостоятельная работа. 14(0) ч. Упрощенные варианты спиральной модели

#### Классические методологии разработки программных приложений

Семестр 2

##### Структурное программирование

Лекция. 2(0) ч. Основные положения структурного программирования. Реализация основ структурного программирования в языках программирования. Графические представления структурного программирования. Методы нисходящего проектирования. Методы восходящего проектирования

Лабораторная работа. 4(0) ч. Метод Дамке. Схемы Насси–Шнейдермана

Самостоятельная работа. 22(0) ч. Методы расширения ядра. Метод JSP Джексона

##### Методология объектно-ориентированного проектирования сложных систем

Лекция. 4(0) ч. Основы объектно-ориентированного анализа и проектирования. Основы языка UML.

Лабораторная работа. 6(0) ч. Диаграмма моделирования в языке UML

Самостоятельная работа. 20(0) ч. Диаграмма вариантов использования

Семестр	Контрольные точки	Баллы
2	<b>Текущий контроль</b> в разделе «Разработка программных приложений»	
	Защита проекта	60
2	<b>Экзамен</b>	
	Устный ответ на вопрос экзаменационного билета	15
	Тест	25
Итого за семестр 2:		100

## Учебно-методическое и информационное обеспечение учебного процесса

### Образовательные технологии (в том числе на занятиях, проводимых в интерактивных формах).

Теоретическая часть курса, общие вопросы разработки программных приложений излагаются в лекционном курсе. Отдельные вопросы могут выноситься на самостоятельное изучение. Для приобретения навыков работы на ПК предназначены лабораторные занятия. При изучении дисциплины используются интерактивные формы занятий (лекция-дискуссия, защита рефератов).

- [Методич рек CPC.doc](#)

Режим доступа: [https://my.bsu.ru/content/file/2/21/213/69325\\_metodich\\_rek\\_srs.doc](https://my.bsu.ru/content/file/2/21/213/69325_metodich_rek_srs.doc)

### Учебно-методические материалы, в том числе методические указания для обучающихся по освоению дисциплины

Теоретическая часть курса, общие вопросы методики и технологий применения компьютерных средств излагаются преподавателем в лекционном курсе. Отдельные вопросы могут выноситься на самостоятельное изучение. Студент должен иметь в виду, что на лекциях преподаватель определяет такие вопросы и рекомендует необходимую для их изучения литературу, источники и др. ресурсы. Для успешного освоения курса необходимо внимательно фиксировать основные положения лекции, своевременно их усваивать, при необходимости самостоятельно прорабатывать, используя основную и дополнительную литературу.

Для приобретения навыков общения с ПК в процессе освоения инструментальных систем и отладки программ предназначены лабораторные занятия. Лабораторные занятия проводятся в специальных классах, оборудованных средствами вычислительной техники. На первом лабораторном занятии студенты получают инструктаж по технике безопасности при работе в классе и знакомятся с особенностями работы на конкретной вычислительной машине. Последующие лабораторные работы заключаются в освоении инструментальных систем и отладке программ решения типовых задач. Индивидуальные задания и методические указания к выполнению каждой последующей лабораторной работы студент получает, как правило, на предыдущем занятии. Подготовка к выполнению лабораторных работ осуществляется в часы самостоятельной работы. Студенты, не подготовившиеся к занятиям, к работе на компьютере не допускаются.

По каждой выполненной лабораторной работе студент оформляет отчет по установленной форме.

Самостоятельные занятия под контролем преподавателя предназначены для самостоятельного изучения студентами тех разделов курса, по которым не предусмотрено чтение лекций, либо проводятся лекции обзорного характера. По усмотрению преподавателя в часы индивидуальных занятий студентам может поручаться выполнение других заданий. Занятия проводятся с академической группой или с половиной группы в часы, установленные расписанием занятий. На занятиях студент должен иметь конспект лекций, учебную и справочную литературу, отдельную тетрадь для записей. Весь теоретический материал, изученный в процессе индивидуальных занятий, должен быть законспектирован.

### **Оценочные средства**

По данной дисциплине разработаны оценочные средства, критерии их оценивания, а также методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

- [Методич рек СРС.doc](#)
- [ФОС\\_Разработка программных приложений\\_Маги.docx](#)

### **Список литературы**

Перечень основной и дополнительной литературы, необходимой для освоения дисциплины.

#### **Основная**

1. [Программная инженерия. Визуальное моделирование программных систем:](#) Учебник для вузов/Черткова Е. А.. —Москва: Юрайт, 2022. —147 с.  
Режим доступа: <https://urait.ru/bcode/491629>
2. [Программная инженерия. Парадигмы, технологии и CASE-средства:](#) Учебник для вузов/Лаврищева Е. М.. —Москва: Юрайт, 2022. —280 с.  
Режим доступа: <https://urait.ru/bcode/491048>
3. [Принципы разработки программных пакетов](#)/Нобак М.. —Москва: ДМК Пресс, 2020. —274 с.  
Режим доступа: <https://e.lanbook.com/book/179459>
4. [Программная инженерия и технологии программирования сложных систем:](#) Учебник для вузов/Лаврищева Е. М.. —Москва: Юрайт, 2022. —432 с.  
Режим доступа: <https://urait.ru/bcode/491029>

#### **Дополнительная**

1. [Программирование. Объектно-ориентированный подход:](#) Учебник и практикум для вузов/Зыков С. В.. —Москва: Юрайт, 2021. —155 с.  
Режим доступа: <https://urait.ru/bcode/470281>

### **Перечень ресурсов информационно-коммуникационной сети «Интернет», необходимых для освоения дисциплины**

<http://www.intuit.ru/department/informatics/intinfo/> - Введение в информатику. Курс Интернет-университета информационных технологий

<http://www.junior.ru/wwwexam/> - Информатика и информационные технологии - web-

конспект и тесты

<http://www.intuit.ru/department/se/pinform/> - Практическая информатика. Курс Интернет-университета информационных технологий

<http://www.ict.edu.ru> - Информационно-коммуникационные технологии в образовании: федеральный образовательный портал

<http://www.computer-museum.ru> - Виртуальный компьютерный музей

<http://www.problems.ru/inf> - Задачи по информатике

<http://iit.metodist.ru> Информатика - и информационные технологии: сайт лаборатории информатики МИОО

<http://www.intuit.ru> - Интернет-университет информационных технологий (ИНТУИТ.ру)

<http://www.edu-it.ru> - ИТ-образование в России: сайт открытого е-консорциума

<http://edu.of.ru> - Конструктор образовательных сайтов (проект Российского общеобразовательного портала)

<http://labinfo.ioso.ru> - Лаборатория обучения информатике Института содержания и методов обучения РАО

<http://test.specialist.ru> - Онлайн-тестирование и сертификация по информационным технологиям

<http://www.firststeps.ru> - Первые шаги: уроки программирования

<http://algolist.manual.ru> - Проект AlgoList: алгоритмы и, методы

<http://alglib.sources.ru> - Проект Alglib.ru: библиотека алгоритмов

**Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)**

CodeBlocks

Office Standard 2007

Windows 7 Корпоративная

Личный кабинет преподавателя или студента БГУ <http://my.bsu.ru>

**Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине**

Аудитория 4204

Корпус:4

Назначение аудитории:

учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации

Число посадочных мест:20

Площадь (кв. м):48.5

Оборудование:

персональный компьютер - 12 шт.

доска аудиторная - 1 шт.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ  
ФГБОУ ВО «БУРЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ ДОРЖИ БАНЗАРОВА»

**Фонд оценочных средств по дисциплине**  
**Разработка программных приложений**

Направление подготовки/ специальность  
**09.04.02 Информационные системы и технологии**

Профиль подготовки /специализация  
Проектирование, разработка и эксплуатация информационных систем

Квалификация (степень) выпускника

магистр

Форма обучения

очная

### Паспорт фонда оценочных средств

ОПК-5.2 Умеет модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач

ОПК-6.2 Умеет проводить анализ современных методов и средств информатики для решения прикладных задач различных классов

ОПК-8.2 Осуществляет выбор средств разработки, оценивает сложность проектов, планирует ресурсы, контролирует сроки выполнения и оценивает качество полученного результата.

№	Контролируемые разделы, темы, модули	Формируемые компетенции	Этапы формирования	Оценочные средства	
				Вид	Количество
1.	Стратегии разработки программных приложений	ОПК-5.2 ОПК-6.2 ОПК-8.2	3 семестр	Тест	1
2.	Классические методологии разработки программных приложений	ОПК-5.2 ОПК-6.2 ОПК-8.2	3 семестр	Контрольная работа	1
Всего:				2	2

### Описание показателей и критериев оценивания уровня приобретенных компетенций на различных этапах их формирования

Результаты обучения	Показатели оценивания	Критерии сформированности	Шкала оценивания
<b>Знает:</b> базовые понятия, лежащие в основе процесса разработки программных приложений; основные модели жизненного цикла программных средств. <b>Умеет:</b> выбирать стратегию разработки программных средств и систем; строить диаграммы моделирования в языке UML <b>Владеет:</b> знаниями положений структурного	<b>Пороговый уровень</b> (как обязательный для всех студентов)	<b>Знает:</b> базовые понятия, лежащие в основе процесса разработки программных приложений; <b>Умеет:</b> выбирать стратегию разработки программных средств и систем; <b>Владеет:</b> знаниями положений структурного программирования	60-69 баллов
	<b>Базовый уровень</b>	<i>помимо удовлетворения критериям порогового уровня:</i> <b>Знает:</b> основные модели жизненного цикла программных средств; <b>Умеет:</b> строить диаграммы моделирования в языке UML; <b>Владеет:</b> основами объектно-ориентированного анализа и проектирования	70-84 баллов
	<b>Высокий уровень</b>	<i>помимо удовлетворения критериям порогового и базового уровня:</i> <b>Знает:</b> классическую каскадную модель, Каскадную модель с обратными связями;	85-100 баллов



программирования; навыками объектно-ориентированного анализа и проектирования		<b>Умеет:</b> строить диаграммы вариантов использования; <b>Владеет:</b> методом расширения ядра и методом JSP Джексона	
--	--	--	--

**Балльно-рейтинговая система**  
по дисциплине «Разработка программных приложений»

Общая максимальная сумма баллов, которую студент может набрать по дисциплине в течение семестра – 100 баллов: 60 баллов текущий контроль и рубежный контроль + 40 баллов зачет/экзамен (итоговый контроль);

– общая максимальная сумма баллов, которую студент может набрать в течение семестра за выполнение всех видов работ во время аудиторных и внеаудиторных занятий, активность и посещаемость, должна быть равна 60 баллам;

– минимальная сумма баллов, при которой студент допускается к зачету/экзамену (итоговому контролю), равна 36 баллам (60% от 60 баллов);

– минимальная сумма баллов, при которой студент получает положительную итоговую оценку по дисциплине равна 60 баллам (60% от 100 баллов).

**Связь между четырехбалльной и стобалльной системами  
оценки качества обучения студентов**

Оценка	Буквенный эквивалент оценки	Рейтинговые баллы
Отлично	A+	95-100
	A	90-94
	A-	85-89
Хорошо	B+	80-84
	B	75-79
	B-	70-74
Удовлетворительно	C+	67-69
	C	64-66
	C-	60-63
Неудовлетворительно	D	40-59
–	F	<40
Зачтено	S	60-100
Не зачтено	U	<60

**Оценочные средства и критерии их оценки**

**Примерные вопросы к экзамену:**

1. Стратегии разработки ПО и модели жизненного цикла. Каскадная модель, спиральная модель и инкрементная стратегии разработки ПО. Перечень, последовательность и содержимое этапов разработки по методологиям. Привести примеры итераций, задач, макетов и моделей.

2. Стратегии разработки ПО и модели жизненного цикла. Макетирование и модель быстрой разработки RAD. Перечень, последовательность и содержимое этапов разработки по методологиям. Примеры итераций, задач, макетов и моделей.

3. Тяжеловесные и облегченные процессы разработки ПО. Экстремальное программирование (XP-процесс). Перечень, последовательность и содержимое этапов разработки по методологиям. Примеры итераций, задач, макетов и моделей.

4. Стратегии разработки ПО и модели жизненного цикла. Agile (Scrum) процесс и компонентно-ориентированная модель. Перечень, последовательность и содержимое этапов разработки по методологиям. Примеры итераций, задач, макетов и моделей.

5. Руководство программным проектом. Планирование проекта. Структурная декомпозиция работ. Ресурсы и риски. Распределение задач и ресурсов. Вехи. Привести определения и примеры.

6. Руководство программным проектом. Планирование проекта. Анализ и управление рисками. Типы рисков. Календари ресурсов. Привести определения и примеры.

7. Оценка бюджета проекта: экспертная, по аналогии, алгоритмическая, на основе графика выполнения. Точность оценки. Оценка затрат, длительности и стоимости разработки. Метрический базис. Привести определения и примеры.

8. Оценка затрат, длительности и стоимости разработки. Оценка размера проекта на основе LOC, FP или OP. Привести определения и примеры.

9. Оценка затрат, длительности и стоимости разработки. Размерно-ориентированные и функционально-ориентированные метрики, их соотношение. Привести определения и примеры.

10. Вычисление затрат, длительности и стоимости разработки на основе моделей СОСОМО. Модели композиции приложения, раннего этапа проектирования и этапа пост-архитектуры. Привести определения и примеры.

11. Оценка затрат, длительности и стоимости разработки. Анализ чувствительности программного проекта с применением модели СОСОМО. Цель и методы выполнения. Привести определения и примеры.

12. Проектирование программных систем. Декомпозиция подсистем на модули.

13. Модуль и модульность. Виды связности модуля. Виды сцепление модулей. Привести определения и примеры.

14. Оценка качества программных систем. Функциональные метрики программных систем. Сложность модуля. Сложность программной системы.

15. Проектирование программных систем. Модели архитектуры: хранилище данных, клиент-сервер, трехуровневая модель, многоуровневая модель, каналы и фильтры, компонентная и каркасы. Привести определения и примеры.

16. Проектирование программных систем. Структурирование системы.

17. Определение моделей архитектуры, управления и интерфейсов взаимодействия. Привести определения и примеры.

18. Структурные шаблоны. MVC: подсистемы, их назначение и схема взаимодействия. Требования и особенности. Реализация обратного вызова. Привести примеры иллюстрирующих диаграмм классов и взаимодействия.

19. Структурные шаблоны. PCMEF: подсистемы, их назначение и схема взаимодействия. Требования и особенности. Делегирование. Привести примеры иллюстрирующих диаграмм классов и взаимодействия.

20. Структурные шаблоны. PCMEF+: подсистемы, их назначение и схема взаимодействия. Требования и особенности. Привести примеры иллюстрирующих диаграмм классов и взаимодействия.

21. Паттерны проектирования. Назначение и принцип описания. Типы и виды паттернов (бизнес-логика, БД, Gof). Параметры паттернов. Привести определения и примеры.

22. Паттерны для реализации бизнес-логики. Сценарий транзакций и модель предметной области. Назначение, особенности реализации и область применения. Примеры диаграмм классов и взаимодействия для указанных паттернов.

#### **Критерии оценки теоретической части:**

- оценка «отлично» (14-15 баллов) выставляется студенту, если он

- Четко знает стратегии разработки программных средств и систем, умеет создавать модели жизненного цикла, овладел умением строить диаграммы моделирования в языке UML.

- Дает четкий и правильный ответ, выявляющий понимание учебного материала и характеризующий прочные знания, излагает материал в логической последовательности с использованием принятой терминологии;

- Ошибок не делает, но допускает оговорки по невнимательности при работе с программными продуктами, которые легко исправляет по требованию преподавателя;

- Ответ логичен, последователен, технически грамотен.

оценка «хорошо» (11-13 баллов) выставляется студенту, если он

- Овладел программным материалом, ориентируется в базовых концепциях разработки программных средств и систем, умеет создавать модели жизненного цикла с небольшим затруднением, но умеет строить диаграммы моделирования в языке UML;
- Дает правильный ответ в определенной логической последовательности;  
*оценка «удовлетворительно» (9-10 баллов) выставляется студенту, если он*
- Основной программный материал знает нетвердо, но большинство изученных понятий и обозначений усвоил;
- Ответ дает неполный, построенный несвязно, но выявивший общее понимание вопросов;  
*оценка «неудовлетворительно» (0-8 баллов) выставляется студенту, если он*
- Обнаруживает незнание или непонимание большей или наиболее важной части учебного материала;
- Ответы строит несвязно, допускает существенные ошибки, которые не может исправить даже с помощью преподавателя.

### Примерные тестовые задания к экзамену:

#### Тест

1 вопрос:

Укажите последовательность этапов ЖЦ ПС

Последовательность понятий:

- [3] Проектирование
- [2] Анализ требований и определение спецификаций
- [4] Реализация
- [5] Сопровождение
- [1] Постановка задачи

2 вопрос:

Первой моделью ЖЦ ПС является

Варианты ответа:

- ☒ каскадная
- ☐ спиральная
- ☐ с промежуточным контролем
- ☐ сопровождение

3 вопрос:

Какая модель ЖЦ ПО предъявляет высокие требования к наглядности описания разрабатываемого программного средства

Варианты ответа:

- ☐ RAD-технология
- ☐ COM-технология
- ☐ GPRS
- ☒ CASE-технология

4 вопрос:

Какая модель ЖЦ предполагает быстрое получение версий разрабатываемого программного обеспечения

Варианты ответа:

- ☐ CASE-технология
- ☒ RAD-технология
- ☐ COM-технология
- ☐ CMM

5 вопрос:

Автономно компилируемая программная единица

Ответ:

модуль

6 вопрос:

При каком методе программирования используются три вида вычислительного процесса: линейный, разветвленный, циклический

Варианты ответа:

☒ структурное

☐ объектно-ориентированное

☐ процедурное

☐ модульное

7 вопрос:

Формализованное текстовое описание алгоритма например, Выполнять <действие> До <условие>)

Ответ:

псевдокод

8 вопрос:

Программы и системы, ориентированные на решение конкретных пользовательских задач

Варианты ответа:

☐ системные

☒ прикладные

☐ оболочки

☐ общего назначения

9 вопрос:

Свойство программного продукта, обеспечивающее правильность работы при любых допустимых данных и защиту от неправильных данных

Варианты ответа:

☐ адаптируемость

☐ правильность

☒ универсальность

☐ точность результатов

10 вопрос:

Адресованный компьютеру набор инструкций, точно описывающий последовательность действий, которые необходимо выполнить

Ответ:

программа

11 вопрос:

Программное обеспечение, представляющее собой организованную совокупность подсистем (программ), позволяющих решать широкий класс задач из некоторой прикладной области

Варианты ответа:

☐ пакет программ

☐ программный комплекс

☒ программная система

☐ программа

12 вопрос:

Вид пользовательского интерфейса, реализующий работу принципом иерархической структуры

Варианты ответа:

☐ прямое манипулирование

☒ меню

☐ свободная навигация

☐ примитивный

13 вопрос:

Языки программирования, являющиеся частью профессиональных сред, характеризующиеся узкой направленностью для создания конкретных типов программного обеспечения

Варианты ответа:

☐ языки низкого уровня

☒ специализированные языки разработчика

☐ специализированные языки пользователя

14 вопрос:

Стандартный язык описания разработки программных продуктов с использованием объектного подхода

Варианты ответа:

☒ UML

☐ XML

☐ CASE-средства

☐ CMM

15 вопрос:

Порция информации, участвующая в диалоговом обмене

Ответ:

сообщение

16 вопрос:

Процесс выполнения программы, целью которого является выявление ошибок

Варианты ответа:

☐ отладка

☒ тестирование

☐ компиляция

☐ интерпретация

17 вопрос:

Процесс локализации и исправления ошибок, обнаруженных при тестировании

Ответ:

отладка

18 вопрос:

Система точно сформулированных правил, определяющая процесс преобразования исходных данных в результат за конечное число шагов

Варианты ответа:

☐ внешнее описание

☒ алгоритм

☐ информационный поток

☐ программный комплекс

19 вопрос:

Поддержка работоспособности программы (переход на новые версии, внесение изменений и т.д.)

Варианты ответа:

☐ отладка

☒ сопровождение

☐ компиляция

☐ программирование

20 вопрос:

Используется для обеспечения процесса разработки программ, включает специализированное программное обеспечение, поддерживающее все технологические этапы процесса проектирования, программирования, отладки и тестирования создаваемых программ

Варианты ответа:

☐ системное программное обеспечение

☒ инструментарий технологии программирования

☐ прикладное программное обеспечение

21 вопрос:

Транслирует программу без её выполнения

Варианты ответа:

☒ компилятор

☐ транслятор

☐ отладчик

☐ интерпретатор

22 вопрос:

В технологии программирования акцент делается на

Варианты ответа:

☒ изучение технологических процессов разработки программного средства и порядке их прохождения

- ☐ разработку детального внешнего описания разрабатываемого программного средства
- ☐ создание качественного программного средства
- ☐ изучение инструментальных средств разработки программных средств

23 вопрос:

К линейным структурам данных с последовательным распределением относятся

Варианты ответа:

- ☐ вектор
- ☐ массив
- ☐ запись
- ☐ двусвязный список

24 вопрос:

Единственно верным сцеплением программных модулей является

Варианты ответа:

- ☐ сцепление по содержимому
- ☐ сцепление по образцу
- ☒ сцепление по данным
- ☐ сцепление по внешним ссылкам

25 вопрос:

Характеристикой программного модуля, отражающей независимость от предыстории обращений к нему, является

Варианты ответа:

- ☒ рутинность
- ☐ прочность
- ☐ сцепление
- ☐ размер

26 вопрос:

Метод разработки программ при модульном программировании, когда реализация начинается с головного модуля, затем программируются модули уровнем ниже и т.д.

Варианты ответа:

- ☐ восходящая разработка
- ☒ нисходящая разработка
- ☐ архитектурный подход
- ☐ структурный подход

27 вопрос:

Словарь терминов внешнего описания предназначен для

Варианты ответа:

- ☐ формализованного описания алгоритма
- ☒ повышения степени понимания предметной области
- ☐ хранения основных показателей качества программного средства
- ☐ упрощения процесса тестирования программного средства

28 вопрос:

Какая модель отражает функциональную диаграмму

Варианты ответа:

- ☐ DFD
- ☐ ERD
- ☒ SADT
- ☐ STD

29 вопрос:

Для обозначения множества объектов, имеющих одинаковую структуру, поведение и отношения с другими объектами, используется

Варианты ответа:

- ☒ класс
- ☐ метод
- ☐ событие
- ☐ DFD-диаграмма

30 вопрос:

Какая модель ЖЦ соответствует CASE-технологии?

Варианты ответа:

- ☐ каскадная
- ☒ спиральная
- ☐ RAD
- ☐ модель с промежуточным контролем

31 вопрос:

При каком тестировании проверяется минимально возможный компонент (класс или функция)?

Варианты ответа:

- ☐ интеграционное
- ☐ альфа-тестирование
- ☒ модульное
- ☐ "черного ящика"

32 вопрос:

При каком тестировании проверяет, есть ли проблемы в интерфейсах или во взаимодействии между компонентами системы?

Варианты ответа:

- ☒ интеграционное
- ☐ системное
- ☐ модульное
- ☐ отладка

33 вопрос:

При каком тестировании разработчик теста имеет доступ к исходному коду и может вносить изменения в код?

Варианты ответа:

- ☐ модульное
- ☒ "белого ящика"
- ☐ "черного ящика"
- ☐ интеграционное

34 вопрос:

Какие ошибки "ловятся" на стадии компиляции?

Варианты ответа:

- ☐ многопользовательского доступа
- ☒ синтаксиса
- ☐ инструментария
- ☐ логики взаимосвязи модулей

35 вопрос:

Какой раздел не входит в руководство пользователя?

Варианты ответа:

- ☐ инструкция по работе
- ☐ общие сведения о программе
- ☐ описание установки
- ☒ методика испытаний

36 вопрос:

Какие языки позволяют осуществлять программирование машинными командами?

Варианты ответа:

- ☒ низкого уровня
- ☐ специализированные языки пользователя
- ☐ высокого уровня
- ☐ специализированные языки разработчика

37 вопрос:

Метод защиты информации шифрованием от несанкционированного прочтения

Варианты ответа:

- ☐ правовой
- ☐ лицензирование
- ☒ криптография
- ☐ обфускация

38 вопрос:

Как называются тесты, составляющиеся исходя из текста программы?

Варианты ответа:

- ☐ функциональные
- ☒ структурные
- ☐ эффективные
- ☐ специализированные

39 вопрос:

Что не является характеристикой программного модуля?

Варианты ответа:

- ☐ один вход и один выход
- ☐ функциональная завершенность
- ☐ логическая независимость
- ☒ высокая квалификация разработчика

40 вопрос:

Многопользовательскую архитектуру реализуют системы, построенные по принципу

Варианты ответа:

- ☐ пакет программ
- ☐ локальная сеть
- ☒ клиент-сервер
- ☐ программная система

### Критерии оценки:

- оценка «отлично» (23-25 баллов) выставляется студенту, если количество правильных ответов >90%;
- оценка «хорошо» (20-22 баллов): 76%-90%;
- оценка «удовлетворительно» (15-19 балла): -50% - 75%;
- оценка «неудовлетворительно» (0-14 баллов): <50%.

### Примерные темы проектов

1. Разработка приложения для торгового предприятия
2. Разработка приложения для кредитного отдела банка
3. Разработка приложения для гостиницы
4. Разработка приложения для авторемонтной мастерской
5. Разработка приложения для автосалона
6. Разработка приложения для агентства недвижимости
7. Разработка приложения для склада
8. Разработка приложения для учета расходов семьи
9. Разработка приложения для организации делопроизводства
10. Разработка приложения для рекламного агентства
11. Разработка приложения для службы поддержки
12. Разработка приложения для кадровой службы организации
13. Разработка приложения для туристического агентства
14. Разработка приложения компьютерного магазина
15. Разработка приложения для страховой компании
16. Разработка приложения для кафедры вуза
17. Разработка приложения для мебельного магазина
18. Разработка приложения для книжного магазина
19. Разработка приложения для учета договоров в организации
20. Разработка приложения для строительной компании
21. Разработка приложения по взаимодействию с клиентами организации
22. Разработка приложения для взаимодействия с заказчиками в организации
23. Разработка приложения для салона красоты



24. Разработка приложения сервисного центра по ремонту компьютерной техники
25. Разработка приложения для транспортной компании
26. Разработка приложения для менеджера по продажам
27. Разработка приложения для учета компьютерной техники и программного обеспечения в организации
28. Разработка приложения для организации документооборота
29. Разработка приложения для управляющей компании ЖКХ
30. Разработка приложения для салона сотовой связи

#### **Критерии оценки:**

*оценка «отлично» (56-60 баллов)* выставляется студенту, если

- правильно построена модель жизненного цикла, обоснована стратегия разработки программного приложения, создана UML диаграмма, разработанное программное приложение функционирует без сбоев;
- продемонстрировано усвоение изученного учебного материала, сформированность и устойчивость используемых умений и навыков;

*оценка «хорошо» (45-55 баллов)* выставляется студенту, если

- правильно построена модель жизненного цикла, обоснована стратегия разработки программного приложения, создана UML диаграмма, разработанное программное приложение в некоторых случаях работает некорректно;
- продемонстрировано усвоение изученного учебного материала;

*оценка «удовлетворительно» (31-44 баллов)* выставляется студенту, если он

- модель жизненного цикла или UML диаграмма построена с недочетами, разработанное программное приложение в некоторых случаях работает некорректно;
- продемонстрировал недостаточную сформированность основных умений и навыков в применении правил на практике;

*оценка «неудовлетворительно» (менее 30 баллов)* выставляется студенту, если он

- модель жизненного цикла построена неправильно, разработанное программное приложение не отвечает поставленным требованиям либо не функционирует;
- обнаружено незнание или непонимание большей или наиболее важной части учебного материала;
- допущены грубые ошибки при использовании изученных правил и приемов.