

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
(МИНОБРНАУКИ РОССИИ)

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНБАССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ДонГТУ»)

Факультет	информационных технологий и автоматизации производственных процессов
Кафедра	интеллектуальных систем и информационной безопасности



УТВЕРЖДАЮ
И.о. проректора
по учебной работе

Д.В. Мулов

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Программная инженерия
(наименование дисциплины)

09.04.01 Информатика и вычислительная техника
(код, наименование направления/специальности)

Искусственный интеллект и цифровые двойники предприятий
(код, наименование направления/специальности)

Квалификация магистр
(бакалавр/специалист/магистр)

Форма обучения очная
(очная, очно-заочная, заочная)

Алчевск, 2024

1 Цели и задачи изучения дисциплины

Цели дисциплины. Целью изучения дисциплины «Программная инженерия» является ознакомление студентов с основными принципами создания спецификации требований, разработки, модификации и сопровождения программного обеспечения, что позволит производить высококачественный программный продукт.

Задачи изучения дисциплины:

- изучить особенности управления процессом разработки программного обеспечения;
- изучить модели процесса создания программного обеспечения; основные этапы процесса создания программного обеспечения; основные принципы аттестации и верификации программного обеспечения; основные принципы управления проектом создания программного обеспечения;
- овладеть навыками разработки спецификации требований к программному обеспечению;
- применять и создавать компоненты многократного использования;
- управлять персоналом занятым созданием программного продукта.

Дисциплина направлена на формирование общепрофессиональной (ОПК-8) компетенции выпускника.

2 Место дисциплины в структуре ОПОП ВО

Логико-структурный анализ дисциплины – курс входит в обязательную часть БЛОКА 1 «Дисциплины (модули)» специальности 09.04.01 Информатика и вычислительная техника.

Дисциплина реализуется кафедрой интеллектуальных систем и информационной безопасности.

Основывается на базе дисциплин: «Информатика», «Программирование», «Системное программирование», изученных студентами при подготовке на уровне бакалавриата.

Является основой для изучения следующих дисциплин: «Выполнение и защита выпускной квалификационной работы (магистерская работа)».

Для изучения дисциплины необходимы компетенции, сформированные у студента для решения профессиональных задач деятельности, связанных с разработкой программного обеспечения.

Курс является фундаментом для ориентации студентов в сфере разработки программного обеспечения.

Общая трудоемкость освоения дисциплины составляет 3 зачетных единицы, 108 ак.ч. Программой дисциплины предусмотрены лекционные (18 ак.ч.), лабораторные (18 ак.ч.) занятия и самостоятельная работа студента (72 ак.ч.).

Дисциплина изучается на 1 курсе в 2 семестре. Форма промежуточной аттестации – экзамен.

3 Перечень результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОПОП ВО

Процесс изучения дисциплины «Объектно-ориентированное программирование» направлен на формирование компетенций, представленных в таблице 1.

Таблица 1 –Компетенции, обязательные к освоению

Содержание компетенции	Код компетенции	Код и наименование индикатора достижения компетенции
Способен осуществлять эффективное управление разработкой программных средств и проектов	ОПК-8	ОПК-8.1. Применяет методы и средства разработки программного обеспечения, методы управления проектами разработки программного обеспечения, способы организации проектных данных, нормативно-технические документы (стандарты и регламенты) по разработке программных средств и проектов

4 Объём и виды занятий по дисциплине

Общая трудоёмкость учебной дисциплины составляет 3 зачётных единицы, 108 ак.ч.

Самостоятельная работа студента (СРС) включает проработку материалов лекций, подготовку к практическим занятиям, текущему контролю, выполнение индивидуального задания, самостоятельное изучение материала и подготовку к экзамену.

При организации внеаудиторной самостоятельной работы по данной дисциплине используются формы и распределение бюджета времени на СРС для очной формы обучения в соответствии с таблицей 2.

Таблица 2 – Распределение бюджета времени на СРС

Вид учебной работы	Всего ак.ч.	Ак.ч. по семестрам
		2
Аудиторная работа, в том числе:	36	36
Лекции (Л)	18	18
Практические занятия (ПЗ)	-	-
Лабораторные работы (ЛР)	18	18
Курсовая работа/курсовой проект	-	-
Самостоятельная работа студентов (СРС), в том числе:	72	72
Подготовка к лекциям	4	4
Подготовка к лабораторным работам	9	9
Подготовка к практическим занятиям / семинарам	-	-
Выполнение курсовой работы / проекта	—	—
Расчетно-графическая работа (РГР)	-	-
Реферат (индивидуальное задание)	-	-
Домашнее задание	-	-
Подготовка к контрольным работам	-	-
Подготовка к коллоквиуму	-	-
Аналитический информационный поиск	10	10
Работа в библиотеке	10	10
Подготовка к экзамену	36	36
Промежуточная аттестация – экзамен (Э)	Э	Э
Общая трудоемкость дисциплины		
ак.ч.	108	108
з.е.	3	3

5 Содержание дисциплины

С целью освоения компетенции, приведенной в п.3 дисциплина разбита на 14 тем:

Тема 1 (Введение. Процесс создания программного обеспечения);

Тема 2 (Смешанные модели процесса создания программного обеспечения. Этапы создания программного обеспечения);

Тема 3 (Основные процессы управления проектом разработки программного обеспечения. Управление рисками);

Тема 4 (Требования к программному обеспечению. Управление требованиями к программному обеспечению);

Тема 5 (Архитектурное проектирование. Основные типы архитектуры программного обеспечения);

Тема 6 (Объектно-ориентированное проектирование. Использование UML диаграмм для объектно-ориентированного проектирования);

Тема 7 (Проектирование с повторным использованием компонентов);

Тема 8 (Проектирование интерфейса пользователя);

Тема 9 (Верификация и аттестация);

Тема 10 (Процесс тестирования программного обеспечения);

Тема 11 (Управление персоналом. Оценка стоимости программного продукта);

Тема 12 (Обеспечение качества программного продукта. Совершенствование производства программного обеспечения);

Тема 13 (Наследуемые системы. Модернизация программного обеспечения);

Тема 14 (Реинжиниринг программного обеспечения).

Виды занятий по дисциплине и распределение аудиторных часов для очной формы приведены в таблице 3.

Таблица 3 – Виды занятий по дисциплине и распределение аудиторных часов (очная форма обучения)

№ п/п	Наименование темы (раздела) дисциплины	Содержание лекционных занятий	Трудоем- кость в ак.ч.	Темы практи- ческих занятий	Трудоем- кость в ак.ч.	Тема лабораторных занятий	Трудоем- кость в ак.ч.
1	2	3	4	5	6	7	8
1	Введение. Процесс создания программного обеспечения	Цель дисциплины «инженерия программного обеспечения». Методы инженерии программного обеспечения. Компоненты методов инженерии программного обеспечения. Основные фундаментальные процессы при разработке программного обеспечения. Основные проблемы, стоящие перед специалистами по программному обеспечению. Жизненный цикл программного обеспечения. Каскадная модель процесса разработки программного обеспечения. Эволюционная модель процесса разработки программного обеспечения. Разработка программного обеспечения на основе ранее созданных компонентов.	1	-	-	Планирование проекта разработки программного обеспечения с использованием Microsoft Project	6

Продолжение таблицы 3

1	2	3	4	5	6	7	8
2	Смешанные модели процесса создания программного обеспечения. Этапы создания программного обеспечения	Итерационные модели разработки программного обеспечения. Модель пошаговой разработки, спиральная модель разработки. Спецификация программного обеспечения. Проектирование программного обеспечения. Программирование и настройка программного обеспечения. Аттестация программных систем. Эволюция программных систем.	1	—	—		
3	Основные процессы управления проектом разработки программного обеспечения. Управление рисками.	Основные сложности в управлении программными проектами. Процессы управления программным проектом. Составление плана проекта. График работ. Временные и сетевые диаграммы графика работ. Распределение этапов работы над проектом по исполнителям. Основные типы рисков программных проектов. Стадии процесса управления рисками. Категории рисков. Анализ рисков. Стратегии управления рисками.	1	—	—		

Продолжение таблицы 3

1	2	3	4	5	6	7	8
4	Требования к программному обеспечению. Управление требованиями к программному обеспечению.	Пользовательские требования, системные требования к программному обеспечению. Функциональные и нефункциональные требования. Требования предметной области. Документирование требований к программному обеспечению. Процесс разработки требований к программного обеспечения. Формирование и анализ требований. Утверждение требований к программному проекту. Планирование управления требованиями. Управление изменениями требований.	2	–	–		
5	Архитектурное проектирование. Основные типы архитектуры программного обеспечения	Понятие «архитектурное проектирование». Основные подходы к процессу архитектурного проектирования. Структурирование системы. Модели управления. Модульная декомпозиция. Модель потоков данных. Проблемно-зависимые архитектурные модели	1	–	–	Работа с системой отслеживания ошибок Microsoft Visual Studio Работа с системой контроля версий Microsoft Visual Studio	6

Продолжение таблицы 3

1	2	3	4	5	6	7	8
6	Объектно-ориентированное проектирование. Использование UML диаграмм для объектно-ориентированного проектирования.	Объекты и классы объектов. Объектно-ориентированный анализ. Параллельные объекты. Основные этапы процесса объектно-ориентированного проектирования. Окружение системы и модели ее использования. Основные виды UML диаграмм. Спецификация интерфейсов объектов. Модификация системной архитектуры	1	–	–		
7	Проектирование с повторным использованием компонентов	Повторно используемые программы. Повторно используемые компоненты. Повторно используемые функции. Преимущества повторного использования программного обеспечения. Проблемы повторного использования программного обеспечения. Компонентная разработка программного обеспечения. Интерфейсы компонента. Уровни абстракции компонентов. Повторное использование коммерческих программных продуктов. Разработка повторно используемых компонентов. Проектные паттерны.	1	–	–		

Продолжение таблицы 3

1	2	3	4	5	6	7	8
8	Проектирование интерфейса пользователя	Элементы графических интерфейсов. Преимущества графических интерфейсов. Процесс проектирования пользовательского интерфейса. Принципы проектирования пользовательского интерфейса. Основные типы взаимодействия с пользователем. Преимущества и недостатки различных стилей взаимодействия с пользователем. Правильное представление данных. Средства поддержки пользователя. Справочная система и документация программного обеспечения.	2	–	–		

Продолжение таблицы 3

1	2	3	4	5	6	7	8
9	Верификация и аттестация.	Понятие «верификация». Понятие «аттестация». Основные различия верификации и аттестации программного обеспечения. Инспекция программного обеспечения. Планирование верификации и аттестации. Структура плана испытаний программного обеспечения. Основные различия инспектирования и тестирования программного обеспечения. Основные роли в процессе инспектирования программы. Основные этапы автоматического статического анализа программы. Разработка программного обеспечения по методу «чистая комната». Ключевые моменты метода «чистая комната».	1	—	—		

Продолжение таблицы 3

1	2	3	4	5	6	7	8
10	Процесс тестирования программного обеспечения.	Основные этапы тестирования программного обеспечения. Основные различия в тестировании объектно-ориентированных и функционально-ориентированных систем. Процесс тестирования дефектов. Тестирование методом «черного ящика» (функциональное тестирование). Области эквивалентности входных и выходных данных. Выбор тестовых данных. Правила выбора последовательностей данных для тестирования программного обеспечения. Метод структурного тестирования. Тестирование ветвей. Тестирование сборки. Нисходящие и восходящие процессы тестирования. Сравнение нисходящего и восходящего процесса тестирования. Тестирование интерфейсов. Классы ошибок в интерфейсах. Общие правила тестирования интерфейсов. Тестирование с нагрузкой. Тестирование объектно-ориентированных систем. Основные подходы при тестировании сборок объектно-ориентированных систем. Инструментальные средства тестирования.	1	–	–	Тестирование программного обеспечения	6

Продолжение таблицы 3

1	2	3	4	5	6	7	8
11	Управление персоналом. Оценка стоимости программного продукта.	Модели решения проблем и мотивация сотрудников. Групповая работа. Основные факторы, влияющие на групповую работу. Подбор членов команды. Сплоченность команды. Преимущества и недостатки сплоченных групп. Общение в группе. Организация группы. Факторы, влияющие на выбор персонала. Модель оценки уровня развития персонала. Факторы, которые влияют на стоимость программного продукта. Методы оценки себестоимости. Алгоритмическое моделирование стоимости. Модель СОСОМО.	1	–	–		
12	Обеспечение качества программного продукта. Совершенствование производства программного обеспечения.	Основные виды стандартов для обеспечения качества программного обеспечения. Качество процесса создания программного обеспечения. Планирование качества. Контроль качества. Процесс измерения показателей программного обеспечения. Качество продукта и производства. Измерение производственного процесса. Модели оценки уровня развития. Оценка уровня развития.	1	–	–		

Завершение таблицы 3

1	2	3	4	5	6	7	8
13	Наследуемые системы. Модернизация программного обеспечения.	Общие структуры наследуемых систем. Логические составляющие наследуемых систем. Проектирование наследуемых систем. Оценивания наследуемых систем. Основные стратегии модернизации программных систем. Динамика развития программ. Эволюция системной архитектуры.	1	–	–		
14	Реинжиниринг программного обеспечения.	Понятие «реинжиниринг». Преимущества реинжиниринга. Процесс реинжиниринга. Преобразование исходного кода программ. Анализ систем. Совершенствование структуры программ. Создание программных модулей. Создание абстракций данных. Изменение данных.	1	–	–		
Всего аудиторных часов			18	–	–		18

6 Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации по итогам освоения дисциплины

6.1 Критерии оценивания

В соответствии с Положением о кредитно–модульной системе организации образовательного процесса ФГБОУ ВО «ДонГТУ» (https://www.dstu.education/images/structure/license_certificate/polog_kred_modul.pdf) при оценивании сформированности компетенций по дисциплине используется 100-балльная шкала.

Перечень работ по дисциплине и способы оценивания знаний приведены в таблице 4.

Таблица 4 – Перечень работ по дисциплине и способы оценивания знаний

Вид учебной работы	Способ оценивания	Количество баллов
Выполнение лабораторных работ	Предоставление отчетов	50 - 70
Выполнение домашнего задания	Предоставление материалов домашнего задания	30 - 30
Итого	-	60-100

Экзамен проставляется автоматически, если студент набрал в течении семестра не менее 60 баллов и отчитался за каждую контрольную точку. Минимальное количество баллов по каждому из видов текущей работы составляет 60% от максимального.

Экзамен по дисциплине «Программная инженерия» проводится по программированию» проводится по результатам работы в семестре. В случае, если полученная в семестре сумма баллов не устраивает студента, во время сессии студент имеет право повысить итоговую оценку либо в форме устного собеседования по приведенным ниже вопросам (п.п. 6.5), либо в результате тестирования.

Шкала оценивания знаний при проведении промежуточной аттестации приведена в таблице 5.

Таблица 5 –Шкала оценивания знаний

Сумма баллов за все виды учебной деятельности	Оценка по национальной шкале зачёт/экзамен
0-59	Не зачтено/неудовлетворительно
60-73	Зачтено/удовлетворительно
74-89	Зачтено/хорошо
90-100	Зачтено/отлично

6.2 Домашнее задание

Домашнее задание не предусмотрено.

6.3 Темы для рефератов (презентаций) – индивидуальное задание

Реферат (индивидуальное задание) не предусмотрен.

6.4 Оценочные средства для самостоятельной работы и текущего контроля успеваемости

Тема 1.

1. Какая из моделей разработки программного обеспечения подразумевает последовательное выполнение этапов от анализа требований до тестирования и выпуска продукта?

- а) Эволюционная модель
- б) Каскадная модель
- в) Модель с итерациями
- г) Модель спирали

2. Какой из компонентов методов инженерии программного обеспечения отвечает за гарантирование качества разработки на всех этапах?

- а) Управление проектом
- б) Тестирование
- в) Риск-менеджмент
- г) Документация

3. Что является основным недостатком каскадной модели разработки программного обеспечения?

- а) Отсутствие обратной связи с пользователем
- б) Невозможность внесения изменений после начала работы
- в) Долгое время разработки
- г) Сложность тестирования

4. Опишите основные этапы жизненного цикла программного обеспечения и их значение в процессе разработки.

5. Какие проблемы могут возникать при использовании каскадной модели разработки программного обеспечения, и как эти проблемы решаются в эволюционной модели?

Тема 2.

1. Какая из моделей разработки программного обеспечения предполагает итерационный подход, где каждый цикл разработки включает планирование, анализ, проектирование и тестирование?

- а) Модель пошаговой разработки
- б) Спиральная модель
- в) Каскадная модель
- г) Модель с быстрыми прототипами

2. Что из перечисленного является важнейшей частью проектирования программного обеспечения?

- а) Тестирование
- б) Спецификация
- в) Программирование
- г) Настройка программного обеспечения

3. Какой процесс является завершающим этапом в создании программного обеспечения, когда система проверяется на соответствие требованиям и стандартам?

- а) Программирование
- б) Аттестация программных систем
- в) Проектирование
- г) Эволюция программных систем

4. Опишите, что такое спиральная модель разработки программного обеспечения и какие преимущества она имеет по сравнению с каскадной моделью.

5. Какие этапы включает в себя процесс проектирования программного обеспечения, и почему они так важны для успешной разработки?

Тема 3.

1. Какой из перечисленных элементов является важной частью составления плана проекта разработки программного обеспечения?

- а) График работ
- б) Анализ рисков
- в) Тестирование системы
- г) Спецификация требований

2. Какая диаграмма используется для визуализации взаимосвязей между задачами и определения времени их выполнения в рамках проекта

разработки программного обеспечения?

- а) Временная диаграмма
- б) Сетевое планирование
- в) Сетевые диаграммы графика работ
- г) Диаграмма Ганта

3. Какой тип рисков связан с возможностью нарушения сроков и потери ресурсов в процессе разработки программного обеспечения?

- а) Технические риски
- б) Организационные риски
- в) Финансовые риски
- г) Внешние риски

4. Опишите основные этапы процесса управления рисками в проекте разработки программного обеспечения и как они помогают минимизировать влияние рисков на проект.

5. Какие сложности могут возникнуть при управлении программными проектами, и какие стратегии управления рисками могут помочь в их преодолении?

Тема 4.

1. Какие из следующих требований относятся к нефункциональным требованиям программного обеспечения?

- а) Производительность системы
- б) Пользовательский интерфейс
- в) Функции обработки данных
- г) Тестирование системы

2. Что из следующего является основным этапом в процессе разработки требований к программному обеспечению?

- а) Документирование требований
- б) Программирование
- в) Тестирование
- г) Эксплуатация системы

3. Какая из следующих категорий требований относится к характеристикам системы, не связанным напрямую с ее функциональностью, например, безопасность и надежность?

- а) Пользовательские требования
- б) Системные требования
- в) Нефункциональные требования
- г) Требования предметной области

4. Опишите процесс управления требованиями к программному

обеспечению и как он влияет на успех проекта.

5. Какова роль утверждения требований к программному проекту и что может произойти, если требования не были должным образом утверждены?

Тема 5.

1. Что является основной целью архитектурного проектирования программного обеспечения?

- а) Создание графического интерфейса пользователя
- б) Определение общей структуры системы и взаимодействия ее компонентов

в) Написание исходного кода программы

г) Оптимизация алгоритмов

2. Какой из подходов используется при структурировании системы?

а) Процедурное программирование

б) Модульная декомпозиция

в) Инкапсуляция данных

и) Нейросетевые алгоритмы

3. Какая архитектурная модель наиболее часто применяется для организации потока данных?

а) Клиент-серверная модель

б) Модель потоков данных

в) Объектно-ориентированная модель

г) Монолитная архитектура

4. В чем заключается основное различие между модульной декомпозицией и моделью потоков данных?

5. Опишите, как выбор архитектурной модели может зависеть от специфики разрабатываемого программного продукта.

Тема 6.

1. Какой основной целью является объектно-ориентированное проектирование?

а) Разработка алгоритмов для обработки данных

б) Определение структуры и взаимодействия объектов в системе

в) Создание баз данных для хранения информации

г) Оптимизация сетевых соединений

2. Какой тип UML-диаграмм используется для моделирования динамического поведения системы?

а) Диаграмма классов

б) Диаграмма последовательностей

в) Диаграмма развертывания

г) Диаграмма компонентов

3. Что представляет собой объект в объектно-ориентированном проектировании?

- а) Абстрактное понятие без конкретных свойств
- б) Физический файл в системе
- в) Экземпляр класса с определенным состоянием и поведением
- г) Интерфейс для связи между программными модулями

4. В чем заключается отличие объектно-ориентированного анализа от объектно-ориентированного проектирования?

5. Опишите, как можно использовать UML-диаграммы для спецификации интерфейсов объектов в системе.

Тема 7.

1. Какое из перечисленных утверждений лучше всего описывает повторно используемые компоненты в программном обеспечении?

- а) Это отдельные программные модули, которые можно интегрировать в различные системы
- б) Это алгоритмы, разработанные для конкретной задачи без возможности их модификации
- в) Это программные файлы, которые нельзя изменять после компиляции
- г) Это только библиотеки стандартных функций операционной системы

2. Какое из преимуществ повторного использования программного обеспечения является наиболее значимым?

- а) Уменьшение размера кода программы
- б) Упрощение интерфейса пользователя
- в) Сокращение времени и затрат на разработку программного обеспечения
- г) Повышение требований к аппаратному обеспечению

3. Какой аспект наиболее важен при разработке повторно используемых компонентов?

- а) Спецификация интерфейсов компонента
- б) Использование редкого языка программирования
- в) Исключение документации для защиты кода
- г) Ограничение количества входных параметров функций

4. Какие основные проблемы могут возникнуть при повторном использовании программных компонентов?

5. Как проектные паттерны помогают в создании повторно

используемых компонентов?

Тема 8.

1. Какой из перечисленных элементов не относится к графическому интерфейсу пользователя?

- а) Кнопки и выпадающие списки
- б) Командная строка
- в) Меню и панели инструментов
- г) Поля ввода данных

2. Какой принцип проектирования пользовательского интерфейса способствует снижению нагрузки на пользователя?

- а) Минимализм в дизайне
- б) Использование сложных анимаций
- в) Запрет на использование горячих клавиш
- г) Максимальное количество элементов управления на экране

3. Какое из преимуществ графического интерфейса является наиболее значимым?

- а) Уменьшение объема кода программы
- б) Улучшенная интуитивность и удобство использования
- в) Снижение затрат на разработку программного обеспечения
- г) Исключение необходимости тестирования

4. Какие основные этапы включает процесс проектирования пользовательского интерфейса?

5. В чем заключаются преимущества и недостатки различных стилей взаимодействия с пользователем?

Тема 9.

1. Какое основное отличие верификации от аттестации программного обеспечения?

а) Верификация проверяет соответствие требованиям, а аттестация подтверждает пригодность ПО для эксплуатации

б) Верификация включает тестирование, а аттестация — нет

в) Аттестация проводится только разработчиками, а верификация — независимыми экспертами

г) Верификация выполняется после аттестации

2. Какой метод используется для формального анализа кода без его выполнения?

- а) Динамическое тестирование
- б) Автоматический статический анализ
- в) Инспекция интерфейсов

г) Юнит-тестирование

3. Какая основная идея метода "чистая комната" в разработке программного обеспечения?

а) Использование машинного обучения для написания кода

б) Минимизация ошибок за счет строгого формального проектирования и отсутствия отладочного тестирования

в) Разработка программ без использования сторонних библиотек

г) Автоматическое исправление ошибок с помощью нейросетей

4. В чем заключаются основные различия между инспектированием и тестированием программного обеспечения?

5. Какие ключевые этапы включает автоматический статический анализ программы?

Тема 10.

1. Какой из перечисленных методов относится к функциональному тестированию?

а) Тестирование методом «белого ящика»

б) Тестирование методом «черного ящика»

в) Тестирование кода с использованием инструментов статического анализа

г) Отладка программы вручную

2. Какое основное различие между нисходящим и восходящим процессами тестирования?

а) В нисходящем тестировании используются эмуляторы, а в восходящем — реальные компоненты

б) В нисходящем тестировании проверяют сначала высокоуровневые модули, а в восходящем — низкоуровневые

в) В восходящем тестировании тестирование начинается с пользовательского интерфейса

г) Нисходящее тестирование выполняется вручную, а восходящее — автоматически

3. Что является основной целью тестирования интерфейсов?

а) Поиск логических ошибок в алгоритмах

б) Проверка соответствия пользовательского интерфейса требованиям

в) Обнаружение ошибок во взаимодействии между модулями системы

г) Оптимизация производительности приложения

4. Какие основные этапы включает процесс тестирования программного обеспечения?

5. В чем заключаются особенности тестирования объектно-ориентированных систем по сравнению с функционально-ориентированными?

Тема 11.

1. Какой из факторов наиболее влияет на сплоченность команды разработчиков?

- а) Единая цель и доверие между участниками
- б) Высокие зарплаты сотрудников
- в) Использование одинаковых языков программирования
- г) Работа в разных часовых поясах

2. Какой метод используется для оценки стоимости программного продукта с учетом множества факторов?

- а) Прогнозирование на основе пользовательских отзывов
- б) Оценка стоимости по методу аналогий
- в) Алгоритмическое моделирование, например, модель СОСОМО
- г) Простая калькуляция стоимости за час работы программиста

3. Какой из перечисленных факторов НЕ влияет на стоимость программного продукта?

- а) Сложность проекта
- б) Количество разработчиков в команде
- в) Географическое расположение заказчика
- г) Используемая операционная система

4. Какие ключевые факторы следует учитывать при подборе членов команды разработки программного обеспечения?

5. В чем суть алгоритмического моделирования стоимости программного продукта и как работает модель СОСОМО?

Тема 12.

1. Какой основной целью является контроль качества программного обеспечения?

- а) Ускорение процесса разработки
- б) Обнаружение и устранение дефектов на различных этапах жизненного цикла ПО

- в) Снижение затрат на тестирование
- г) Автоматизация всех процессов разработки

2. Какой из перечисленных стандартов используется для обеспечения качества программного обеспечения?

- а) HTTP/2
- б) ISO/IEC 25010

в) TCP/IP

г) CSS

3. Какой из факторов наиболее влияет на качество процесса создания программного обеспечения?

а) Использование единого языка программирования

б) Применение методов планирования, контроля и измерения показателей качества

в) Размер команды разработчиков

г) Отсутствие документации для ускорения работы

4. Какие основные этапы включает процесс измерения показателей программного обеспечения?

5. Как оценка уровня развития процессов производства влияет на качество программного продукта?

Тема 13.

1. Какой основной проблемой обладают наследуемые системы?

а) Использование современных языков программирования

б) Сложность поддержки и интеграции с новыми технологиями

в) Высокая степень документированности

г) Автоматическое обновление без вмешательства разработчиков

2. Какой из подходов наиболее часто используется при модернизации устаревших программных систем?

а) Полная замена системы на новую

б) Инкрементальная модернизация с заменой отдельных компонентов

в) Остановка работы системы для анализа кода

г) Создание параллельной копии системы без изменений

3. Что является ключевым этапом при оценке наследуемой системы перед модернизацией?

а) Выбор цветовой схемы интерфейса

б) Анализ логической структуры и технического долга системы

в) Определение количества пользователей

г) Проверка уровня продаж продукта

4. Какие основные стратегии модернизации программных систем существуют, и в чем их различие?

5. Как эволюция системной архитектуры влияет на процесс модернизации программного обеспечения?

Тема 14.

1. Какова основная цель реинжиниринга программного обеспечения?

а) Полная замена старого кода на новый

б) Улучшение структуры и производительности существующей системы без изменения ее основной функциональности

в) Разработка нового программного обеспечения с нуля

г) Оптимизация аппаратных ресурсов системы

2. Какой этап не относится к процессу реинжиниринга программного обеспечения?

а) Анализ системы

б) Создание новых программных модулей

в) Разработка аппаратного обеспечения

г) Преобразование исходного кода

3. Что означает создание абстракций данных в рамках реинжиниринга?

а) Добавление новых баз данных в систему

б) Оптимизация структуры данных для улучшения логики работы программы

в) Исключение всех структур данных из программы

г) Уменьшение количества комментариев в коде

4. Какие основные преимущества дает реинжиниринг программного обеспечения?

5. Какие этапы включает процесс реинжиниринга, и в чем заключается их основное назначение?

6.5 Вопросы для подготовки к экзамену

1) Каково использование UML для описания спецификаций программного обеспечения?

2) Каково описание аспектов поведения программного обеспечения с применением вариантов использования?

3) Каково представление поведения системы с применением диаграмм вариантов использования?

4) Какие Вы знаете модели процесса разработки программного обеспечения?

5) Какие Вы знаете подходы к разработке программного обеспечения?

6) Что Вы знаете о построении концептуальной модели предметной области?

7) Что Вы знаете о диаграмме последовательностей системы? Что такое системные события и операции?

8) Что такое диаграммы компоновки и размещения программных компонентов?

9) Каково определение отношений классов с использованием UML?

- 10) Каково назначение диаграмм вариантов использования?
- 11) Каково назначение диаграмм классов?
- 12) Каково назначение диаграммы пакетов?
- 13) Каково назначение диаграмм последовательностей действий?
- 14) Каково назначение диаграмм кооперации?
- 15) Каково назначение диаграмм деятельности?
- 16) Каково назначение диаграмм состояний объектов?
- 17) Каково назначение диаграмм компонентов?
- 18) Каково назначение диаграмм размещения?
- 19) Какие подходы к разработке программного обеспечения: проектирование исходя из выполняемых ПО функций (задач) Вы знаете?
- 20) Какие подходы к разработке программного обеспечения: проектирование, основанное на исследовании потоков данных Вы знаете?
- 21) Какие подходы к разработке программного обеспечения: проектирование, основанное на структуре данных Вы знаете?
- 22) Какие подходы к разработке программного обеспечения: проектирование на базе абстрактных типов данных?
- 23) Какие подходы к разработке программного обеспечения: объектно-ориентированное проектирование Вы знаете?
- 24) Что такое модели процесса разработки программного обеспечения: "водопад" (Waterfall)?
- 25) Что такое модели процесса разработки программного обеспечения: прототипирование?
- 26) Что такое модели процесса разработки программного обеспечения: итерация?
- 27) Что такое модели процесса разработки программного обеспечения: спираль?
- 28) Что такое стандартизация в сфере информационных технологий?
- 29) Каковы структура и область применения стандарта ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология? Что такое процессы жизненного цикла программных средств»?
- 30) Какова структура процесса разработки согласно ГОСТ Р ИСО/МЭК 12207-2010?
- 31) Каковы структура и область применения Единой системы программной документации (ГОСТ 19)?
- 32) Какие Вы знаете программные и эксплуатационные документы согласно ЕСПД (ГОСТ 19)?
- 33) Какие Вы знаете стадии разработки согласно ЕСПД (ГОСТ 19)?

34) Каковы структура и содержание ГОСТ 19?201-78? Техническое задание?

35) Каковы структура и содержание ГОСТ 19?301-79? Программа и методика испытаний?

36) Каковы структура и содержание ГОСТ 19?402-78? Описание программы?

37) Каковы структура и содержание ГОСТ 19?404-79? Пояснительная записка?

38) Каковы структура и содержание ГОСТ 19?503-79? Руководство системного программиста?

7. Учебно-методическое и информационное обеспечение дисциплины

7.1 Рекомендуемая литература

Основная литература

1. Петрухин, В. А. Методы и средства инженерии программного обеспечения : краткий курс / В. А. Петрухин, Е. М. Лаврищева. - Москва : ИНТУИТ, 2016. - 337 с. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2146205>. (дата обращения: 26.08.2024). – Режим доступа: по подписке.
2. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2024. — 400 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0812-9. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2136716>. (дата обращения: 26.08.2024). – Режим доступа: по подписке.

Дополнительная литература

1. Конструирование программного обеспечения : учебное пособие / под ред. Л.Г. Гагариной. — Москва : ИНФРА-М, 2024. — 319 с. — (Высшее образование). — DOI 10.12737/1893880. - ISBN 978-5-16-017861-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1893880>. (дата обращения: 26.08.2024). – Режим доступа: по подписке..

7.2 Базы данных, электронно-библиотечные системы, информационно-справочные и поисковые системы

1. Научная библиотека ДонГТУ : официальный сайт.— Алчевск. — URL: library.dstu.education.— Текст : электронный.
2. Научно-техническая библиотека БГТУ им. Шухова : официальный сайт. — Белгород. — URL: <http://ntb.bstu.ru/jirbis2/>.— Текст : электронный.
3. Консультант студента : электронно-библиотечная система.— Москва. — URL: <http://www.studentlibrary.ru/cgi-bin/mb4x>.— Текст : электронный.
4. Университетская библиотека онлайн : электронно-библиотечная система.— URL: http://biblioclub.ru/index.php?page=main_ub_red.— Текст : электронный.
5. Сайт кафедры ИСИБ <http://scs.dstu.education>

8 Материально-техническое обеспечение дисциплины

Материально-техническая база обеспечивает проведение всех видов деятельности в процессе обучения, соответствует требованиям ФГОС ВО.

Материально-техническое обеспечение представлено в таблице 6.

Таблица 6 – Материально-техническое обеспечение

Наименование оборудованных учебных кабинетов	Адрес (местоположение) учебных кабинетов
Специальные помещения: <i>Мультимедийная аудитория. (60 посадочных мест), оборудованная специализированной (учебной) мебелью (скамья учебная – 20 шт., стол– 1 шт., доска аудиторная– 1 шт.), учебное ПК (монитор + системный блок), мультимедийная стойка с оборудованием – 1 шт., широкоформатный экран.</i> Аудитории для проведения лекций: <i>Компьютерные классы (22 посадочных места), оборудованный учебной мебелью, компьютерами с неограниченным доступом к сети Интернет, включая доступ к ЭБС: ПК– 12 шт.; Доска – 1 шт.</i>	ауд. <u>207</u> корп. <u>4</u> ауд. <u>211</u> корп. <u>4</u>

Лист согласования рабочей программы дисциплины

Разработал

ст. преподаватель кафедры
интеллектуальных систем
и информационной безопасности
(должность)


(подпись)

А.И. Фомин
(Ф.И.О.)

И.о. заведующего кафедрой
интеллектуальных систем и
информационной безопасности
(наименование кафедры)

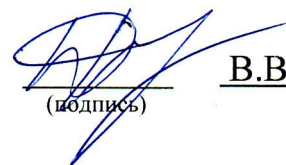

(подпись)

Е.Е. Бизянов
(Ф.И.О.)

Протокол № 1 заседания кафедры
интеллектуальных систем
и информационной безопасности

от 27.08.2024 г.

Декан факультета
информационных технологий и
автоматизации производственных
процессов


(подпись)

В.В. Дьячкова
(Ф.И.О.)

Согласовано

Председатель методической
комиссии по направлению
подготовки 09.04.01
Информатика и вычислительная техника


(подпись)

Е.Е. Бизянов
(Ф.И.О.)

Начальник учебно-методического центра


(подпись)

О.А. Коваленко
(Ф.И.О.)

Лист изменений и дополнений

Номер изменения, дата внесения изменения, номер страницы для внесения изменений	
ДО ВНЕСЕНИЯ ИЗМЕНЕНИЙ:	ПОСЛЕ ВНЕСЕНИЯ ИЗМЕНЕНИЙ:
Основание:	
Подпись лица, ответственного за внесение изменений	