

Скоков
Роман
Юрьевич
Ректор "
ВИЭПП"

2026.02.27
15:46:53
+03'00'

Приложение 2 к рабочей программе дисциплины
«Программная инженерия»

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОЛЖСКИЙ ИНСТИТУТ ЭКОНОМИКИ, ПЕДАГОГИКИ И ПРАВА»

Факультет информационной безопасности и права

Фонд оценочных средств
по дисциплине
«Программная инженерия»

Направление подготовки:
09.03.03 Прикладная информатика

Направленность (профиль) образовательной программы:
Прикладная информатика в юриспруденции

Уровень высшего образования:
бакалавриат

Квалификация выпускника:
«бакалавр»

Волжский 2026 г.

Содержание

1	Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы, описание показателей, критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания.....	3
2	Методические материалы и типовые контрольные задания, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы.....	9
2.1	Типовые контрольные задания, используемые для проведения входного контроля.....	9
2.2	Типовые контрольные задания, используемые для промежуточной аттестации по дисциплине.....	9
2.2.1	Примерный перечень вопросов к экзамену.....	9
2.2.2	Примерное задание на экзамен.....	10
2.3	Методические материалы и типовые контрольные задания, используемые для текущего контроля по дисциплине.....	11
2.3.1	Методические материалы, используемые для текущего контроля знаний по дисциплине.....	12
2.3.2	Вопросы, выносимые на самостоятельное изучение.....	13
2.3.3	Задания для самостоятельной работы.....	12
2.3.4	Тесты по дисциплине.....	13
2.3.5	Типовые задания.....	14
2.3.6	Задания для контрольной работы.....	14
2.3.7	Методика проведения лекции-дискуссии.....	14
2.3.8	Методика проведения интерактивного решения задач.....	15
3	Методические материалы, определяющие процедуры оценивания знаний, умений, навыков..... по дисциплине.....	16
3.1	Балльно-рейтинговая система оценки успеваемости по дисциплине.....	16

1 Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы, описание показателей, критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Перечень компетенций	Показатели оценивания компетенций	Критерии оценивания компетенций	Этапы формирования компетенций	Шкала оценивания
ОПК-1. Способен применять естественнонаучные и инженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	- знает основные методы математического анализа, моделирования и экспериментального исследования в ИТ-сфере; - умеет применять математические методы для решения задач проектирования и оптимизации программных систем; - владеет навыками выбора адекватных методов моделирования для конкретных задач разработки ПО	- имеет базовые знания методов математического анализа; - умеет на основе типовых примеров применять методы для решения простых задач; - владеет навыками выбора методов из ограниченного набора	Начальный	удовлетворительно (60–74 балла)
	- знает принципы ООП (инкапсуляция, наследование, полиморфизм, абстракция); методологии и парадигмы проектирования; стандарты оформления кода; жизненный цикл ПО; - умеет проектировать архитектуру ПО на основе ООП; выбирать оптимальные паттерны проектирования;	- Знать математический аппарат описания систем; методы формального и имитационного моделирования; - уметь строить математические и концептуальные модели систем, - владеть инструментами анализа систем (диаграммы,	Основной	хорошо (75–89 баллов)

	оценивать качество программного кода; организовывать работу команды разработчиков; - владеет инструментами проектирования (UML, диаграммы); методами рефакторинга; техниками code review; средами разработки (IDE)	матрицы, графы)		
	- знает принципы ООП (инкапсуляция, наследование, полиморфизм, абстракция); методологии и парадигмы проектирования; стандарты оформления кода; жизненный цикл ПО; - умеет проектировать архитектуру ПО на основе ООП; выбирать оптимальные паттерны проектирования; оценивать качество программного кода; организовывать работу команды разработчиков; - владеет инструментами проектирования (UML, диаграммы); методами	Знает систему естественно-научных знаний, методы теоретического и экспериментального исследования в профессиональной деятельности; Умеет - владеет методами, применяемыми на предприятии для совершенствования организационных структур управления на каждом уровне	Завершающий	отлично (90–100 баллов)

	рефакторинга; техниками code review; средами разработки (IDE)			
ОПК-4. Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	- знает структуру и требования к технической документации (ТЗ, спецификации, отчеты); - умеет оформлять техническую документацию в соответствии с отраслевыми стандартами; - владеет навыками составления технических спецификаций	- имеет базовые знания структуры документации; - умеет оформлять документы по шаблонам; - владеет навыками работы с готовыми шаблонами	Начальный	удовлетворительно (60–74 балла)
	- знает методы и виды тестирования ПО; принципы модульного тестирования; инструменты автоматизации тестирования; - умеет разрабатывать тестовые сценарии для ООП-приложений; организовывать процесс тестирования; анализировать результаты тестирования и устранять дефекты; - владеет фреймворками тестирования (JUnit, NUnit); методиками отладки; инструментами профилирования и анализа производительности	- знает структуру и требования к технической документации (ТЗ, спецификации, отчеты); - умеет оформлять техническую документацию в соответствии с отраслевыми стандартами; - владеет инструментами создания технической документации	Основной	хорошо (75–89 баллов)

	- знает методы и виды тестирования ПО; принципы модульного тестирования; инструменты автоматизации тестирования; - умеет разрабатывать тестовые сценарии для ООП-приложений; организовывать процесс тестирования; анализировать результаты тестирования и устранять дефекты; - владеет фреймворками тестирования (JUnit, NUnit); методиками отладки; инструментами профилирования и анализа производительности	- знает методику разработки технической документации;- требования ГОСТ, СТБ и отраслевых регламентов к сетевой инфраструктуре - умеет оформлять документацию в соответствии с установленными стандартами; - владеет методиками верификации соответствия решений стандартам	Завершающий	отлично (90–100 баллов)
ПК-3. Способен руководить интеграцией программных модулей и компонентов компьютерного программного обеспечения	- знает принципы компонентной архитектуры; интерфейсы взаимодействия модулей; методы управления зависимостями; - умеет проектировать интерфейсы взаимодействия модулей; организовывать процесс интеграции; решать конфликты версий и	- имеет базовые знания принципов компонентной архитектуры; - умеет на основе типовых примеров проектировать простые интерфейсы взаимодействия; - владеет навыками применения базовых инструментов контроля	Начальный	удовлетворительно (60–74 балла)

	зависимостей; - владеет технологиями CI/CD; системами контроля версий (Git); инструментами сборки (Maven, Gradle); контейнеризацией (Docker)	версий		
	- знает принципы компонентной архитектуры; интерфейсы взаимодействия модулей; методы управления зависимостями; - умеет проектировать интерфейсы взаимодействия модулей; организовывать процесс интеграции; решать конфликты версий и зависимостей; - владеет технологиями CI/CD; системами контроля версий (Git); инструментами сборки (Maven, Gradle); контейнеризацией (Docker)	- знает принципы компонентной архитектуры; - умеет проектировать интерфейсы взаимодействия и организовывать процесс интеграции; - владеет навыками самостоятельного применения инструментов контроля версий и сборки	Основной	хорошо (75–89 баллов)
	- знает принципы компонентной архитектуры; интерфейсы взаимодействия модулей; методы управления зависимостями; - умеет проектировать	- знает принципы компонентной архитектуры; - умеет руководить процессом интеграции, решать сложные конфликты зависимостей;	Завершающий	отлично (90–100 баллов)

	интерфейсы взаимодействия модулей; организовывать процесс интеграции; решать конфликты версий и зависимостей; - владеет технологиями CI/CD; системами контроля версий (Git); инструментами сборки (Maven, Gradle); контейнеризацией (Docker)	- владеет опытом применения технологий CI/CD и контейнеризации		
--	---	--	--	--

2. Методические материалы и типовые контрольные задания, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

2.1. Типовые контрольные задания, используемые для проведения входного контроля

1. Объясните принцип работы п/п диода.
2. Объясните принцип работы биполярного транзистора.
3. Объясните принцип работы полевого транзистора.
4. Приведите таблицы истинности двухвходовых логических элементов: "И", "ИЛИ", "ИСКЛЮЧАЮЩЕЕ ИЛИ".
5. Как классифицируются языки программирования?
6. Какие системы счисления находят применение в вычислительной технике и почему?

2.2. Типовые контрольные задания, используемые для промежуточной аттестации по дисциплине

2.2.1. Примерный перечень вопросов к экзамену

Теоретические вопросы

1. Краткая история развития ЭВМ. Поколения ЭВМ. Основные области и формы использования ЭВМ.
2. Системы счисления, применяемые в ЭВМ, и их характеристика. Формы представления чисел и алфавитной информации в ЭВМ.
3. Системы кодирования информации на машинных носителях.
4. Основные сведения о кодировании информации и о носителях информации.
5. Машинные коды прямой, обратный и дополнительный.
6. Алгоритмы реализации арифметических операций над машинными кодами чисел в различных формах их представления.
7. Операции над двоично-десятичными кодами десятичных чисел.
8. Последовательность преобразования информации при вводе ее в ЭВМ и при выводе результатов.
9. Роль и место алгебры логики в цифровой вычислительной технике.
10. Функционально полные наборы логических элементов. Комбинационные схемы, основные этапы их построения
11. Классификация элементов ЭВМ.
12. Техническая реализация запоминающих и логических элементов.
13. Современные элементы в интегральном исполнении.
14. Триггеры - их типы, функциональные схемы, таблицы переходов, области применения.
15. Стандартизация системы элементов ЭВМ и их обозначений. Классификация узлов ЭВМ.
16. Регистры: параллельные, сдвиговые.
17. Счетчики. Счетчики с последовательным и параллельным переносом. Суммирующие, вычитающие и реверсивные счетчики, кольцевые счетчики.
18. Шифраторы, дешифраторы. Принципы построения схем дешифраторов и шифраторов.
19. Сумматоры - их назначение, принципы построения, структурные схемы, функционирование.
20. Стандартизация обозначений функциональных узлов ЭВМ.
21. Общие принципы функциональной и структурной организации современных ЭВМ.
22. Структура и характеристика системы команд ЭВМ. Форматы команд. Способы адресации данных в ЭВМ.

23. Структурная организация и взаимодействие узлов и устройств ЭВМ при выполнении основных команд.

24. Структура процессоров ЭВМ.

25. АЛУ: назначение, типовые структуры для различных моделей ЭВМ, алгоритмы функционирования, характеристики.

Практические вопросы

1. Программа учета домашней медиатеки.

2. Программа планирования дел «Ежедневник».

3. Информационная система учета услуг в автомастерской.

4. Программа информационной поддержки спортивных соревнований.

5. Информационно-справочная система для сотрудников ПДН.

6. Программа учета и анализа продаж в продовольственном магазине.

7. Информационная система факультета информационной безопасности и права ВИАПП «Абитуриент».

8. Программа информационного обеспечения фестиваля художественной самодеятельности студентов ВИАПП.

9. Программа информационной поддержки спартакиады ВИАПП.

10. Программа учета и анализа доходов и расходов семьи.

11. Программа формирования счетов-квитанций для жильцов ТСЖ.

12. Система управления теплицей.

13. Программа обработки данных аттестации студентов института.

14. Визуальный конструктор E-сетей.

15. Программа управления очередностью обслуживания клиентов в поликлинике.

16. Программа терминала оплаты за услуги населению

2.2.2. Примерное задание на экзамен

Запустить проект разработки программного обеспечения согласно выданному заданию. Сформировать команду, распределить роли. Выполнить основные действия и создать артефакты, соответствующие данному этапу.

Методические указания и теоретические сведения.

RUP основан на трех ключевых идеях.

Весь ход работ направляется итоговыми целями проекта, выраженными в виде вариантов использования (usecases) — сценариев взаимодействия результирующей программной системы с пользователями или другими системами, при выполнении которых пользователи получают значимые для них результаты и услуги. Разработка начинается с выделения вариантов использования и на каждом шаге контролируется степенью приближения к их реализации. Основным решением, принимаемым в ходе проекта, является архитектура результирующей программной системы. Архитектура устанавливает набор компонентов, из которых будет построено ПО, ответственность каждого из компонентов (т.е. решаемые им подзадачи в рамках общих задач системы), четко определяет интерфейсы, через которые они могут взаимодействовать, а также способы взаимодействия компонентов друг с другом.

Архитектура является одновременно основой для получения качественного ПО и базой для планирования работ и оценок проекта в терминах времени и ресурсов, необходимых для достижения определенных результатов. Она оформляется в виде набора графических моделей на языке UML. Основой процесса разработки являются планируемые и управляемые итерации, объем которых (реализуемая в рамках итерации функциональность и набор компонентов) определяется на основе архитектуры.

Этап НАЧАЛО (Inception)

Главное назначение этапа — запустить проект. Основная цель этой фазы — достичь компромисса между всеми заинтересованными лицами относительно задач проекта и выделяемых на него ресурсов. На этой стадии определяются основные цели проекта, руководитель и бюджет, основные средства выполнения — технологии,

инструменты, ключевые исполнители. Также, возможно, происходит апробация выбранных технологий, чтобы убедиться в возможности достичь целей с их помощью, и составляются предварительные планы проекта. На эту фазу может уходить около 10% времени и 5% трудоемкости одного цикла.

Цели этапа НАЧАЛО: определить область применения проектируемой системы (ее предназначение, границы, интерфейсы с внешней средой, критерий признания — приемки); определить элементы Use Case, критические для системы (основные сценарии поведения, задающие ее функциональность и покрывающие главные проектные решения); определить общие черты архитектуры, обеспечивающей основные сценарии, создать демонстрационный макет; определить общую стоимость и план всего проекта и обеспечить детализированные оценки для этапа развития; идентифицировать основные элементы риска.

Основные действия этапа НАЧАЛО: формулировка области применения проекта — выявление требований и ограничений, рассматриваемых как критерий признания конечного продукта; планирование и подготовка бизнес-варианта и альтернатив развития для управления риском, определение персонала, проектного плана, а также выявление зависимостей между стоимостью, планированием и полезностью; синтезирование предварительной архитектуры, развитие компромиссных решений проектирования; определение решений разработки, покупки и повторного использования, для которых можно оценить стоимость, планирование и ресурсы.

В итоге этапа НАЧАЛО создаются следующие артефакты: Концепция (Vision) — спецификация представления основных проектных требований, ключевых характеристик и главных ограничений; начальная модель Use Case (20% от полного представления); начальный словарь проекта (глоссарий); начальный бизнес-вариант (содержание бизнеса, критерий успеха — прогноз дохода, прогноз рынка, финансовый прогноз); начальное оценивание риска; проектный план, в котором показаны этапы и итерации.

Примеры оформления документов RUP (артефактов) смотри в каталоге «Примеры документов RUP». Более подробное описание процесса разработки по RUP смотри в документе «ПроцессРазработки_по_RUP.doc».

2.3. Методические материалы и типовые контрольные задания, используемые для текущего контроля по дисциплине

2.3.1. Методические материалы, используемые для текущего контроля знаний по дисциплине

Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
Контрольный опрос	Метод оценки уровня освоения компетенций на основе непосредственного или опосредованного взаимодействия преподавателя и студента	Примерный перечень вопросов к экзамену, вопросы для самостоятельного изучения
Собеседование	Относительно свободный диалог между преподавателем и студентом на заданную тему	Вопросы для самостоятельного изучения, задания для самостоятельной работы
Тестовые задания	Эмпирико-аналитическая процедура оценки уровня освоения компетенций	Тесты по дисциплине (раздел 2.3.4)
Самостоятельное решение задач	Метод, при котором обучающиеся приобретают навыки	Типовые задания (раздел 2.3.5), задания для

	творческого мышления и самостоятельного решения проблем	контрольной работы
Лекция-дискуссия	Оценочное средство, позволяющее включить обучающихся в процесс обсуждения спорного вопроса	Методика проведения лекции-дискуссии (раздел 2.3.7)
Интерактивное решение задач	Метод модерации, при котором при решении задач принимают участие все обучающиеся под руководством преподавателя-модератора	Методика проведения интерактивного решения задач (раздел 2.3.8), типовые задания

2.3.2. Вопросы, выносимые на самостоятельное изучение

1. Значение вычислительной техники в современной экономике.
2. История развития ЭВМ.
3. Обобщенная структурная схема ЭВМ. Назначение основных устройств.
4. Объясните, что понимают под аппаратными и программными средствами ЭВМ?
5. Классификация ЭВМ.
6. Системы счисления, применяемые в ЭВМ.
7. Алгебра логики. Понятие о функционально полных наборах логических элементов.
8. Общая классификация элементов ЭВМ.
9. Какой тип логики самый быстродействующий?.
10. Способы адресации данных в ЭВМ.
11. Микропрограммный и аппаратный способы управления ЭВМ.
12. Общая классификация запоминающих устройств.
13. Оперативные запоминающие устройства.
14. Общая классификация внешних устройств.
15. Режимы работы ЭВМ.
16. Понятие о персональной ЭВМ (ПЭВМ) и общая классификация ПЭВМ.
17. Периферийные устройства ПЭВМ и их классификация.
18. Телекоммуникационные средства для передачи данных по каналам связи.
19. Области применения ПЭВМ.
20. Определение, назначение и принципы построения вычислительных сетей (ВСт).

2.3.3. Задания для самостоятельной работы

Разработать диаграмму классов для системы онлайн-банкинга с указанием отношений между сущностями (клиент, счет, транзакция, карта).

Реализовать иерархию классов для системы обработки геометрических фигур с применением полиморфизма для расчета площади и периметра.

Применить паттерн "Наблюдатель" для реализации системы уведомлений в приложении социальной сети.

Провести рефакторинг предоставленного фрагмента кода с нарушением принципов SOLID, обосновав каждое изменение.

Создать репозиторий Git для учебного проекта, настроить ветвление по модели Git Flow и выполнить несколько операций слияния.

Разработать диаграмму последовательности для сценария авторизации пользователя в веб-приложении.

Подготовить техническое задание для разработки простого мобильного приложения "Список задач" с указанием функциональных и нефункциональных требований.

Реализовать паттерн "Стратегия" для поддержки различных способов оплаты в интернет-магазине.

2.3.4. Тесты по дисциплине

1. Где ошибка в записи числовых констант:

- a. 128.256;
- b. 2.385e-12;
- c. \$28b;
- d. 0x368.

2 Как правильно записать текстовую константу:

- a. 'Иванов';
- b. {Петров};
- c. // Сидоров;
- d. "Иркутск".

3. Расположите типы в порядке возрастания размера:

- a. extended;
- b. shortint;
- c. string;
- d. boolean.

4. К какому типу данных применяется операция поразрядного сдвига shl: целому;

- a. вещественному;
- b. символьному;
- c. строковому;

5. Каков тип выражения $X \geq 0$:

- a. целый;
- b. логический;
- c. указательный;
- d. это не выражение, а инструкция;

6 В каком разделе программы используется служебное слово function:

- a. заголовок;
- b. раздел описания;
- c. раздел инструкций;
- d. раздел инициализации.

7 Формальные параметры – это:

- a. параметры вычислительного алгоритма;
- b. локальные переменные в процедуре;
- c. передаваемые значения при вызове функции;
- d. переменные типа класс.

8 Какая команда обеспечивает вывод данных на экран:

- a. fopen(...)
- b. printf
- c. fclose(...)
- d. scanf(...)

9 Какая команда завершает выполнение подпрограммы?

- a. Return
- b. #include
- c. Break
- d. switch

2.3.5. Типовые задания

Задача 1: Разработка архитектуры системы управления библиотекой

1. Создать гостевую книгу, где любой человек может оставить комментарий в текстовом поле и добавить его. Все добавленные комментарии выводятся над текстовым полем
2. Создать страницу, на которой можно загрузить несколько фотографий в галерею. Все загруженные фото должны помещаться в папку uploads и выводиться на странице в виде таблицы.

2.3.6. Задания для контрольной работы

2.3.6.1. Задание к теоретической части контрольной работы

КОНТРОЛЬНАЯ РАБОТА №1

1. Краткая история развития ЭВМ. Поколения ЭВМ. Основные области и формы использования ЭВМ.
2. Системы счисления, применяемые в ЭВМ, и их характеристика. Формы представления чисел и алфавитной информации в ЭВМ.
3. Системы кодирования информации на машинных носителях.
4. Основные сведения о кодировании информации и о носителях информации.
5. Машинные коды прямой, обратный и дополнительный.
6. Алгоритмы реализации арифметических операций над машинными кодами чисел в различных формах их представления.
7. Операции над двоично-десятичными кодами десятичных чисел.
8. Последовательность преобразования информации при вводе ее в ЭВМ и при выводе результатов.

2.3.6.2. Задание к практической части контрольной работы

КОНТРОЛЬНАЯ РАБОТА №2

1. Счетчики. Счетчики с последовательным и параллельным переносом. Суммирующие, вычитающие и реверсивные счетчики, кольцевые счетчики.
2. Шифраторы, дешифраторы. Принципы построения схем дешифраторов и шифраторов.
3. Сумматоры - их назначение, принципы построения, структурные схемы, функционирование.
4. Стандартизация обозначений функциональных узлов ЭВМ.
5. Общие принципы функциональной и структурной организации современных ЭВМ.
6. Структура и характеристика системы команд ЭВМ. Форматы команд. Способы адресации данных в ЭВМ.
7. Структурная организация и взаимодействие узлов и устройств ЭВМ при выполнении основных команд.
8. Структура процессоров ЭВМ.
9. АЛУ: назначение, типовые структуры для различных моделей ЭВМ, алгоритмы функционирования, характеристики.
10. АЛУ для сложения и вычитания чисел с фиксированной запятой.

2.3.7. Методика проведения лекции-дискуссии

Цель: Достижение определенного мнения по обсуждаемой проблеме путем осознания участниками своих мнений, суждений, оценок по теме дискуссии; уточнения взаимных позиций в споре; выработки уважительного отношения к другому мнению.

Задачи:

- Развитие умений анализировать проблему или проблемную ситуацию
- Поиск и разработка перспективных идей

- Развитие способности продуцировать множество решений
- Нахождение оптимального решения среди нескольких альтернатив

Методика проведения:

- Назначение секретаря лекции-дискуссии, его инструктаж по выполняемым функциям
- Объявление критериев оценки
- Свободный обмен мнениями в промежутках между логически оформленными разделами сообщения учебного материала
- Подведение итогов дискуссии и оценка участников дискуссии по материалам, подготовленным секретарем, переход к информационной лекции

Критерии оценки: секретарь лекции-дискуссии – 0,5–1,5 баллов; участие в дискуссии – 0,5–1 балл.

2.3.8. Методика проведения интерактивного решения задач

Цель: Проверка навыков решения профессиональных задач и развитие мыслительных операций обучающихся, направленных на достижение результатов при решении профессиональных задач.

Задачи:

1. Проверка навыков применения обучающимися ранее усвоенных знаний при решении профессиональных задач
2. Формирование навыков совместной деятельности подчиненных (обучающихся) и руководителя (преподавателя)
3. Овладение обучающимися знаниями и общими принципами решения проблемных профессиональных задач
4. Развитие навыков активной интеллектуальной деятельности
5. Развитие коммуникативных навыков (навыков общения)
6. Развитие навыков обучающихся по выработке собственной позиции по ходу решения профессиональных задач

Методика проведения интерактивного решения задач: Первый этап «подготовка проекта решения задач». Преподавателем разрабатывается проект хода решения задачи с началом или фрагментами решения.

Второй этап «организационный»:

1. Объявление темы и цели решения задачи
2. Назначение секретаря и его инструктаж по выполняемым функциям
3. Объявление критериев оценки

Третий этап «интеллектуальный»:

1. Объявление условий решения задач
2. Индивидуальное решение задачи обучающимися, исходя из собственного мнения
3. Высказывание обучающимися мнений по ходу решения задач
4. Обсуждение результатов и методики индивидуального решения задач обучающимися и принятие плана верного хода решения

Четвертый этап «подведение итогов решения задачи»:

1. Формулирование вывода решения задачи обучающимися
2. Подведение итога интерактивного решения задачи преподавателем
3. Оценка преподавателем обучающихся по материалам, подготовленным секретарем

Критерии оценки: секретарь – 1 балл; участие в решении задач – 1–3 балла.

3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков по дисциплине

3.1. Балльно-рейтинговая система оценки успеваемости по дисциплине

В целях оперативного контроля уровня усвоения материала учебной дисциплины и стимулирования активной учебной деятельности студентов (очной формы обучения) используется балльно-рейтинговая система оценки успеваемости. В соответствии с этой системой оценки студенту в ходе изучения дисциплины предоставляется возможность набрать не менее 60 баллов за текущую работу в семестре и до 40 баллов на экзамене (итоговый контроль). Результирующая оценка по дисциплине складывается из суммы баллов текущего и итогового контроля.

К экзамену допускаются студенты, посетившие все лекции и практические занятия, прошедшие промежуточную аттестацию, с рейтингом не менее 30 баллов.

Баллы за текущую работу в семестре по дисциплине складываются из следующих видов деятельности студента:

Балльно-рейтинговая система по дисциплине «Объектно-ориентированное проектирование и программирование»

Виды контроля	Максимальная сумма баллов на выполнение одного вида задания	Всего возможных за семестр работ	Итого максимальная сумма баллов
Присутствие на лекционных и практических занятиях	0,12	72	9
Работа на практических занятиях	1,0	36	36
СРС, в т.ч.:			55
– написание конспектов	0,1	10	1
– защита рефератов	0,3	7	2
– выступление с докладом	0,2	5	1
– самостоятельное решение задач	1	12	12
– написание эссе	0,2	4	8
– контроль СРС (контрольные опросы, тестирование, решение кейсов и др. виды контроля)	0,8	10	8

– выполнение лабораторных работ	2	10	20
– участие в интерактивных занятиях	1	3	3
Итого за семестр	–	–	100
Экзамен	40,0	1	40
Итого по дисциплине	–	–	140

Примечание: Максимальная итоговая оценка – 100 баллов (баллы свыше 100 учитываются как 100).

Текущий контроль по лекционному материалу проводит лектор. Текущий контроль по практическим занятиям проводит преподаватель, проводивший эти занятия.

Балльно-рейтинговая система предусматривает поощрение студентов за работу в семестре проставлением «премиальных» баллов.

Нормы выставления дополнительных баллов по дисциплине «Объектно-ориентированное проектирование и программирование»

Виды работ	Максимальная сумма баллов
Решение дополнительных задач повышенной сложности	5
Участие в олимпиадах, конкурсах, конференциях в рамках предмета дисциплины	5
Публикация научных работ по тематике дисциплины	5
Итого	15

Шкала итоговой оценки

Количество баллов	Оценка
91–100	«отлично»
76–90	«хорошо»
60–75	«удовлетворительно»
0–59	«неудовлетворительно»