

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Баламирзоев Назим Лиодинович
Должность: Ректор
Дата подписания: 25.11.2025 20:28:35
Уникальный программный ключ:
20584ea6d19eae7c3c775fcd8365441470edec7

Приложение А

(обязательное к рабочей программе дисциплины)

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Дагестанский государственный технический университет»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине «Объектно-ориентированное программирование»

Уровень образования

бакалавриат
(бакалавриат/магистратура/специалитет)

Направление подготовки
бакалавриата/магистратуры/специальность

09.03.04 Программная инженерия
(код, наименование направления подготовки специальности)

Профиль направления
подготовки/специализация

Разработка программно-информационных систем
(наименование)

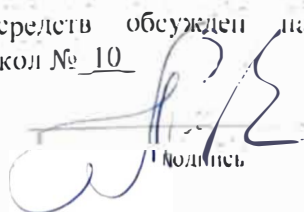
Разработчик


подпись

Алиосманова О.А.
(ФИО уч. степень, уч. звание)

Фонд оценочных средств обсужден на заседании кафедры НОВТИАС
« 20 » июня 2019 г., протокол № 10

Зав. кафедрой


подпись

Айюмов А.И., к.т.н., доцент
(ФИО уч. степень, уч. звание)

г. Каспийск 2021

СОДЕРЖАНИЕ

1.	Область применения, цели и задачи фонда оценочных средств.....	19
2.	Описание показателей и критериев оценивания компетенций, формируемых в процессе освоения дисциплины (модуля).....	20
2.1.	Перечень компетенций с указанием этапов их формирования в процессе освоения ОПОП.....	20
2.1.1.	Этапы формирования компетенций.....	21
2.2.	Показатели уровней сформированности компетенций на этапах их формирования, описание шкал оценивания.....	22
2.2.1.	Показатели уровней сформированности компетенций на этапах их Формирования.....	22
2.2.2.	Описание шкал оценивания.....	23
3.	Типовые контрольные задания, иные материалы и методические рекомендации, необходимые для оценки сформированности компетенций в процессе освоения ОПОП.....	24
3.1.	Задания и вопросы для входного контроля.....	24
3.2.	Оценочные средства и критерии сформированности компетенций.....	24
3.3.	Задания для промежуточной аттестации (зачета и (или) экзамена).....	29

1. Область применения, цели и задачи фонда оценочных средств

Фонд оценочных средств (ФОС) является неотъемлемой частью рабочей программы дисциплины «Объектно-ориентированное программирование» и предназначен для контроля и оценки образовательных достижений обучающихся (в т.ч. по самостоятельной работе студентов, далее – СРС), освоивших программу данной дисциплины.

Целью фонда оценочных средств является установление соответствия уровня подготовки обучающихся требованиям ФГОС ВО по направлению подготовки/специальности **09.03.04 Программная инженерия**.

Рабочей программой дисциплины «Объектно-ориентированное программирование» предусмотрено формирование следующей компетенции:

ПК-8 - Владение навыками использования различных технологий разработки программного обеспечения.

2. Описание показателей и критериев оценивания компетенций, формируемых в процессе освоения дисциплины (модуля)

Описание показателей и критериев оценивания компетенций, формируемых в процессе освоения дисциплины (модуля), и используемые оценочные средства приведены в таблице 1.

2.1. Перечень компетенций с указанием этапов их формирования в процессе освоения ОПОП

Таблица 1

Код и наименование формируемой компетенции	Код и наименование индикатора достижения формируемой компетенции	Критерии оценивания	Наименование контролируемых разделов и тем ¹
ПК-8 - Владение навыками использования различных технологий разработки программного обеспечения	ПК-8.1. Знает современные технологии разработки ПО (структурное, объектно-ориентированное)	Студент должен знать современные технологии разработки программного обеспечения для решения профессиональных задач с использованием структурного и объектно—ориентированного подходов.	Темы №1-17
	ПК-8.2. Умеет использовать современные технологии разработки ПО	Студент должен уметь использовать современные технологии разработки программного обеспечения для решения профессиональных задач с использованием структурного и объектно—ориентированного подходов.	Темы №1-17
	ПК-8.3. Имеет навыки использования современных технологий разработки ПО	Студент должен владеть навыками использования современных технологий разработки программного обеспечения для решения профессиональных задач с использованием структурного и объектно—ориентированного подходов.	Темы №1-17

¹ Наименования разделов и тем должен соответствовать рабочей программе дисциплины.

2.1.1. Этапы формирования компетенций

Сформированность компетенций по дисциплине «Объектно-ориентированное программирование» определяется на следующих этапах:

1. Этап текущих аттестаций
2. Этап промежуточных аттестаций

Таблица 2

Код и наименование формируемой компетенции	Код и наименование индикатора достижения формируемой компетенции	Этапы формирования компетенции					
		Этап текущих аттестаций					Этап промежуточной аттестации
		1-5 неделя	6-10 неделя	11-15 неделя	1-17 неделя		18-20 неделя
		Текущая аттестация №1	Текущая аттестация №2	Текущая аттестация №3	СРС	КР/КП	Промежуточная аттестация
1		2	3	4	5	6	7
ПК-8 - Владение навыками использования различных технологий разработки программного обеспечения	ПК-8.1. Знает современные технологии разработки ПО (структурное, объектно-ориентированное)	Контрольная работа №1	Контрольная работа №2	Контрольная работа №3	Контрольная работа №1, 2, 3	Контрольная работа №1, 2, 3	Экзамен
	ПК-8.2. Умеет использовать современные технологии разработки ПО	Контрольная работа №1	Контрольная работа №2	Контрольная работа №3	Контрольная работа №1, 2, 3	Контрольная работа №1, 2, 3	Экзамен
	ПК-8.3. Имеет навыки использования современных технологий разработки ПО	Контрольная работа №1	Контрольная работа №2	Контрольная работа №3	Контрольная работа №1, 2, 3	Контрольная работа №1, 2, 3	Экзамен

СРС – самостоятельная работа студентов;

КР – курсовая работа;

КП – курсовой проект.

2.2. Показатели уровней сформированности компетенций на этапах их формирования, описание шкал оценивания

2.2.1. Показатели уровней сформированности компетенций на этапах их формирования

Результатом освоения дисциплины «Объектно-ориентированное программирование» является установление одного из уровней сформированности компетенций: высокий, повышенный, базовый, низкий.

Таблица 3

Уровень	Универсальные компетенции	Общепрофессиональные/профессиональные компетенции
Высокий (оценка «отлично», «зачтено»)	Сформированы четкие системные знания и представления по дисциплине. Ответы на вопросы оценочных средств полные и верные. Даны развернутые ответы на дополнительные вопросы. Обучающимся продемонстрирован высокий уровень освоения компетенции	Обучающимся усвоена взаимосвязь основных понятий дисциплины, в том числе для решения профессиональных задач. Ответы на вопросы оценочных средств самостоятельны, исчерпывающие, содержание вопроса/задания оценочного средства раскрыто полно, профессионально, грамотно. Даны ответы на дополнительные вопросы. Обучающимся продемонстрирован высокий уровень освоения компетенции
Повышенный (оценка «хорошо», «зачтено»)	Знания и представления по дисциплине сформированы на повышенном уровне. В ответах на вопросы/задания оценочных средств изложено понимание вопроса, дано достаточно подробное описание ответа, приведены и раскрыты в тезисной форме основные понятия. Ответ отражает полное знание материала, а также наличие, с незначительными пробелами, умений и навыков по изучаемой дисциплине. Допустимы единичные негрубые ошибки. Обучающимся продемонстрирован повышенный уровень освоения компетенции	Сформированы в целом системные знания и представления по дисциплине. Ответы на вопросы оценочных средств полные, грамотные. Продemonстрирован повышенный уровень владения практическими умениями и навыками. Допустимы единичные негрубые ошибки по ходу ответа, в применении умений и навыков
Базовый (оценка «удовлетворительно», «зачтено»)	Ответ отражает теоретические знания основного материала дисциплины в объеме, необходимом для дальнейшего освоения ОПОП. Обучающийся допускает неточности в ответе, но обладает необходимыми знаниями для их устранения. Обучающимся продемонстрирован базовый уровень освоения компетенции	Обучающийся владеет знаниями основного материала на базовом уровне. Ответы на вопросы оценочных средств неполные, допущены существенные ошибки. Продemonстрирован базовый уровень владения практическими умениями и навыками, соответствующий минимально необходимому уровню для решения профессиональных задач
Низкий (оценка «неудовлетворительно», «не зачтено»)	Демонстрирует полное отсутствие теоретических знаний материала дисциплины, отсутствие практических умений и навыков	

Показатели уровней сформированности компетенций могут быть изменены, дополнены и адаптированы к конкретной рабочей программе дисциплины.

2.2.2. Описание шкал оценивания

В ФГБОУ ВО «ДГТУ» внедрена модульно-рейтинговая система оценки учебной деятельности студентов. В соответствии с этой системой применяются пятибалльная, двадцатибалльная и стобалльная шкалы знаний, умений, навыков.

Шкалы оценивания			Критерии оценивания
пятибалльная	двадцатибалльная	стобалльная	
«Отлично» - 5 баллов	«Отлично» - 18-20 баллов	«Отлично» - 85 – 100 баллов	Показывает высокий уровень сформированности компетенций, т.е.: <ul style="list-style-type: none"> – продемонстрирует глубокое и прочное усвоение материала; – исчерпывающе, четко, последовательно, грамотно и логически стройно излагает теоретический материал; – правильно формирует определения; – демонстрирует умения самостоятельной работы с нормативно-правовой литературой; – умеет делать выводы по излагаемому материалу.
«Хорошо» - 4 балла	«Хорошо» - 15 - 17 баллов	«Хорошо» - 70 - 84 баллов	Показывает достаточный уровень сформированности компетенций, т.е.: <ul style="list-style-type: none"> – демонстрирует достаточно полное знание материала, основных теоретических положений; – достаточно последовательно, грамотно логически стройно излагает материал; – демонстрирует умения ориентироваться в нормальной литературе; – умеет делать достаточно обоснованные выводы по излагаемому материалу.
«Удовлетворительно» - 3 балла	«Удовлетворительно» - 12 - 14 баллов	«Удовлетворительно» - 56 – 69 баллов	Показывает пороговый уровень сформированности компетенций, т.е.: <ul style="list-style-type: none"> – демонстрирует общее знание изучаемого материала; – испытывает серьезные затруднения при ответах на дополнительные вопросы; – знает основную рекомендуемую литературу; – умеет строить ответ в соответствии со структурой излагаемого материала.
«Неудовлетворительно» - 2 балла	«Неудовлетворительно» - 1-11 баллов	«Неудовлетворительно» - 1-55 баллов	Ставится в случае: <ul style="list-style-type: none"> – незнания значительной части программного материала; – не владения понятийным аппаратом дисциплины; – допущения существенных ошибок при изложении учебного материала; – неумение строить ответ в соответствии со структурой излагаемого вопроса; – неумение делать выводы по излагаемому материалу.

3. Типовые контрольные задания, иные материалы и методические рекомендации, необходимые для оценки сформированности компетенций в процессе освоения ОПОП

3.1. Задания и вопросы для входного контроля

1. Формат программы на языке C++.
2. Директивы препроцессора.
3. Основные типы данных в C++.
4. Объявление переменных и констант в C++.
5. Операции C++.
6. Управляющие конструкции C++.
7. Объявление функций в C++. Способы передачи параметров функциям.
8. Составные типы данных в C++: массивы, структуры и объединения.
9. Указатели в C++. Работа с массивами и структурами с помощью указателей.
10. Динамическое распределение памяти.

3.2. Оценочные средства и критерии сформированности компетенций

Контрольная работа №1 по темам «Составные типы данных в C++. Динамическое распределение памяти», «ООП как направление развития теории и практики создания ПО», «Понятие класса и объекта в C++. Конструкторы и деструкторы в C++».

Комплект заданий для контрольной работы

- Время выполнения 90 мин.
- Количество вариантов контрольной работы - 6.
- Количество заданий в каждом варианте контрольной работы - 3.
- Форма работы – самостоятельная, индивидуальная.

Вариант 1

Задание 1. Анализ причин сложности современных программных систем. Основы структурного и объектного подходов к программированию. Преимущества и недостатки объектного подхода.

Задание 2. Уровни доступа к членам класса.

Задание 3. Конструктор с параметрами. Пример программы.

Вариант 2

Задание 1. Основные принципы ООП: абстракция и инкапсуляция. Преимущества инкапсуляции.

Задание 2. Объявление объектов. Создание массивов объектов. Пример программы.

Задание 3. Конструктор со списком инициализаторов. Пример программы.

Вариант 3

Задание 1. Понятие класса и объекта в ООП. Состояние и поведение объекта.

Задание 2. Указатели на объекты класса. Динамические объекты и массивы объектов. Пример программы.

Задание 3. Конструктор копирования. Поверхностное и глубинное копирование. Пример программы.

Вариант 4

Задание 1. Суть наследования и его виды.

Задание 2. Синтаксис и семантика конструктора. Указатель this. Пример программы.

Задание 3. Конструктор копирования. Поверхностное и глубинное копирование. Пример программы.

Вариант 5

Задание 1. Мономорфизм и полиморфизм.

Задание 2. Свойства конструкторов.

Задание 3. Синтаксис и семантика деструктора. Пример программы.

Вариант 6

Задание 1. Структура класса C++. Пример программы.

Задание 2. Конструктор по умолчанию. Пример программы.

Задание 3. Свойства деструкторов.

Контрольная работа №2 по темам «Члены класса, общие для всех экземпляров. Константные данные в C++», «Друзья класса (friend) в C++», «Механизмы наследования в C++. Простое и множественное наследование», «Перегрузка функций в C++».

Комплект заданий для контрольной работы

- Время выполнения 90 мин.
- Количество вариантов контрольной работы - 4.
- Количество заданий в каждом варианте контрольной работы - 3.
- Форма работы – самостоятельная, индивидуальная.

Вариант 1

Задание 1. Статические переменные и статические функции класса (static). Назначение. Пример программы.

Задание 2. Методы класса, дружественные другому классу. Пример программы.

Задание 3. Виртуальные базовые классы. Пример программы.

Вариант 2

Задание 1. Константные переменные и функции класса (const). Пример программы.

Задание 2. Дружественные классы. Пример программы.

Задание 3. Замещение членов базового класса. Пример программы.

Вариант 3

Задание 1. Функции, дружественные одному классу. Пример программы.

Задание 2. Объявление классов при простом наследовании. Спецификаторы наследуемого доступа. Пример программы.

Задание 3. Назначение перегрузки функций. Пример программы.

Вариант 4

Задание 1. Функции, дружественные нескольким классам. Пример программы.

Задание 2. Конструкторы и деструкторы производных классов (очередность вызова). Пример программы.

Задание 3. Перегрузка конструкторов. Пример программы.

Контрольная работа №3 по темам ««Перегрузка операторов в C++»», «Полиморфизм и виртуальные функции в C++», «Параметризованные типы в C++. Стандартная библиотека шаблонов C++», «Классы VCL-библиотеки».

Комплект заданий для контрольной работы

- Время выполнения 90 мин.
- Количество вариантов контрольной работы - 6.

- Количество заданий в каждом варианте контрольной работы - 4.
- Форма работы – самостоятельная, индивидуальная.

Вариант 1

- Задание 1. Назначение перегрузки операторов. Правила перегрузки операторов.
 Задание 2. Механизмы раннего и позднего связывания.
 Задание 3. Синтаксис шаблонного класса. Пример программы.
 Задание 4. Ассоциативные контейнеры. Пример.

Вариант 2

- Задание 1. Перегрузка бинарных операторов. Пример программы.
 Задание 2. Виртуальные функции. Таблица виртуальных функций. Пример программы.
 Задание 3. Специализация шаблонов функций и классов. Пример программы.
 Задание 4. Итераторы. Обратные итераторы. Пример.

Вариант 3

- Задание 1. Перегрузка унарных операторов. Пример программы.
 Задание 2. Чисто виртуальные функции и абстрактные классы. Пример программы.
 Задание 3. Достоинства и недостатки шаблонов.
 Задание 4. Алгоритмы библиотеки STL. Пример.

Вариант 4

- Задание 1. Перегрузка оператора доступа к членам класса. Пример программы.
 Задание 2. Назначение виртуальных деструкторов. Пример программы.
 Задание 3. Назначение и состав библиотеки STL.
 Задание 4. Иерархия классов VCL-библиотеки.

Вариант 5

- Задание 1. Перегрузка и неоднозначность. Пример программы.
 Задание 2. Назначение шаблонных функций.
 Задание 3. Последовательные контейнеры. Пример.
 Задание 4. Классы командных кнопок, флажков и радиокнопок VCL-библиотеки.

Вариант 6

- Задание 1. Виды и реализации полиморфизма.
 Задание 2. Конкретизация шаблонной функции. Пример программы.
 Задание 3. Контейнеры – адаптеры. Пример.
 Задание 4. Обработка событий средствами VCL-библиотеки.

Критерии оценки уровня сформированности компетенций при проведении контрольной работы:

- оценка «отлично»: продемонстрировано грамотное последовательное решение задач (заданий) при правильно выбранном алгоритме. Даны верные ответы на все вопросы и условия задач (заданий). При необходимости сделаны пояснения и выводы (содержательные, достаточно полные, правильные, учитывающие специфику проблемной ситуации в задаче или с незначительными ошибками);

- оценка «хорошо»: грамотное последовательное решение задач (заданий) при правильно выбранном алгоритме. Однако, ответы на вопросы и условия задач (заданий) содержат незначительные ошибки. Пояснения и выводы отсутствуют или даны неверно;

- оценка «удовлетворительно»: обучающийся ориентируется в материале, но применяет его неверно, выбирает неправильный алгоритм решения задач (неверные исходные данные, неверная последовательность решения и др. ошибки), допускает вычислительные ошибки. Пояснения и выводы отсутствуют или даны неверно;

- оценка «неудовлетворительно»: обучающийся слабо ориентируется в материале, выбирает неправильный алгоритм решения, допускает значительное количество вычислительных ошибок. Пояснения и выводы отсутствуют.

Курсовая работа/курсовой проект

Примерные темы курсовых работ/курсовых проектов

1. Разработка библиотеки классов, реализующих графические примитивы на плоскости с возможностью аффинных преобразований (окружность, треугольник, прямоугольник). Предусмотреть удобный интерфейс.
2. Разработка библиотеки классов, реализующих графические примитивы в трёхмерном пространстве с возможностью аффинных преобразований (шар, пирамида, куб). Предусмотреть удобный интерфейс.
3. Разработка библиотеки классов, реализующих графические примитивы на плоскости с реализацией операций над множествами.
4. Разработка программы, демонстрирующей движение окружностей, с использованием принципов объектно-ориентированного программирования. По траектории, представляющей собой окружность, происходит движение шаров. Каждый шар характеризуется скоростью, направлением движения и радиусом. Шары имеют синий цвет. Однако, если происходит пересечение шаров, то все шары, попадающие в пересечение, закрашиваются красным цветом на время пересечения. Предусмотреть добавление и удаление шаров, изменение их параметров.
5. Разработка библиотеки классов, реализующих векторы в n-мерном пространстве.
6. Разработка библиотеки классов, реализующих строки и операции над ними, включая работу с регулярными выражениями.
7. Реализация различных типов графов и операций над ними.
8. Разработка объектной модели и программы, описывающих сотрудников предприятия/организации с их функциями (сотрудник, менеджер, ...).
9. Разработка системы классов, описывающих различные транспортные средства.
10. Моделирование замкнутой биологической системы (корм, травоядное, хищник).
11. Моделирование муравейника (несколько типов муравьёв, источники питания, внешние раздражители,...).
12. Моделирование планетарной системы.
13. Система классов для реализации матричной арифметики.
14. Разработка библиотеки классов для работы с комплексными числами. Написать тестовую программу для работы с библиотекой «Комплексное число».
15. Разработка программы для игры «Шахматы».
16. Моделирование дорожного движения на заданной карте дорог. Программа – эмулятор гоночной трассы (со случайными препятствиями, с 4 автомобилями). На бесконечной трассе шириной 10 единиц двигаются 4 автомобиля (вернее двигается сама трасса сверху вниз). Сверху появляются препятствия (не более 3-х общей шириной не более 6 единиц). Препятствия реализовать объектами, создавать их динамически, при выходе их за нижнюю границу удалять из памяти. Автомобили также реализовать в виде объектов (ширина автомобиля 2 единицы), автомобили могут двигаться и по горизонтали, и по вертикали, не накладываясь друг на друга. Для автомобилей предусмотреть обязательное наличие между препятствиями двух «коридоров» по 2 единицы шириной. При столкновении автомобиля с препятствием автомобиль удаляется с трассы и из памяти. Игра заканчивается, когда на трассе остается один автомобиль (о чем вывести сообщение). Всеми автомобилями управляет компьютер.

17. Программа – эмулятор пианино (или любого другого музыкального инструмента). Стандартными средствами языка C++ создать объект нота, от которого создать 7 потомков – 7 нот. Вывести на экран клавиатуру (клавиши пианино). Клавиатуру также реализовать с помощью объектов (иерархию объектов клавиатуры выбрать произвольно).
18. Программа – дизайнер комнаты (с заранее заданными элементами мебели).
19. Разработка программы для игры «Жизнь».
20. Разработка объектной модели и программы «Автозаправка», в которой присутствуют объекты, представляющие транспорт, а также объекты, представляющие топливо.
21. Разработка программы для игры «Ханойская башня». На игровом поле находятся 3 стержня, на которые можно нанизывать кольца только в порядке убывания размера кольца. Изначально все кольца находятся в порядке убывания (самое широкое – снизу, самое узкое – наверху) на первом стержне. Цель игры – переместить все кольца на другой столбик. Сложность игры зависит от количества колец (минимальное количество колец – три). Создать объекты для кольца (с переменным размером) и стержня, у которого будет свойство – семейство колец.
22. Разработка объектной модели и программы «Аквариум».
23. ЯПОНСКИЙ КРОССВОРД. На клеточном поле размером 15x15 случайным образом генерируется рисунок (или отображается заранее заданный рисунок). Затем по горизонтали и вертикали (для каждой строки и столбца соответственно) отображаются последовательности чисел, характеризующие закрашенные клетки. Цель игрока по заданным последовательностям чисел пометить клетки (после чего должен появиться заданный рисунок). Создать объект клетка, от него создать потомок – объект строка, у которого при создании (в конструкторе) нужно расставить (сгенерировать) последовательности закрашенных клеток, после чего необходимо подсчитать и отобразить последовательность чисел для этой строки. Для столбцов подсчет последовательностей чисел произвести после создания всех строк (для чего некоторые поля объекта строка надо будет сделать видимыми – public). Пример готового японского кроссворда размером 5x5.

		1	1	2	1	1
		2	1	1	1	2
	5					
	1					
	0					
1	1					
	5					

24. Составить графическую программу, моделирующую движение транспорта на дорожном перекрестке. По 2 перпендикулярным дорогам должны двигаться автомобили, на пересечении дорог расположены светофоры, которые отображают знак светофора. Автомобили должны останавливаться либо продолжать движение в зависимости от знака светофора, чтобы избежать столкновения автомобилей.
25. Разработка объектной модели «Автомобильный транспорт».
26. Создать объектную модель для автоматизации работы библиотеки.
27. Составить объектную модель «Телефонный справочник».
28. Составить объектную модель «Оружие». Разработать общий класс, в котором будут все настройки для оружия (ударная сила, размеры, т.д.). Создать отдельные классы для разного оружия (холодное, огнестрельное, ракетницы и т.д.). Общие параметры собрать в общем классе.
29. Реализовать программу, моделирующую движение общественного транспорта до остановок. Карта должна состоять из 3 остановок. На каждой остановке должны

появляться люди. В транспорт должно помещаться ограниченное количество человек. У каждого человека должен быть номер нужного ему транспорта и номер остановки, в которой он должен выйти. Номер появившегося транспорта и номер нужного для человека транспорта должен быть задан случайно.

30. Разработать приложение для игры «Тетрис».
31. Разработать приложение для игры «Змейка».
32. Разработать приложение для игры «Сапер».
33. Разработка приложения «Калькулятор».

Критерии оценки уровня сформированности компетенций при выполнении курсовой работы/курсового проекта:

- оценка «отлично»: продемонстрировано блестящее владение проблемой исследования, материал выстроен логично, последовательно, обучающийся аргументированно отстаивает свою точку зрения. Во введении приводится обоснование выбора конкретной темы, четко определены цель и задачи работы (проекта). Использован достаточный перечень источников и литературы для методологической базы исследования. Обучающийся грамотно использует профессиональные термины, актуальные исходные данные. Проведен самостоятельный анализ (исследование) объекта. По результатам работы сделаны логичные выводы. Оформление работы соответствует методическим рекомендациям. Объем и содержание работы соответствует требованиям. На защите обучающийся исчерпывающе отвечает на все дополнительные вопросы;

- оценка «хорошо»: обучающийся демонстрирует повышенный уровень владения проблемой исследования, логично, последовательно и аргументированно отстаивает ее концептуальное содержание. Во введении содержатся небольшие неточности в формулировках цели, задач. В основной части допущены незначительные погрешности в расчетах (в исследовании). Выводы обоснованы, аргументированы. Оформление работы соответствует методическим рекомендациям. Объем работы соответствует требованиям. На защите обучающийся отвечает на все дополнительные вопросы;

- оценка «удовлетворительно»: обучающийся демонстрирует базовый уровень владения проблемой исследования. Во введении указаны цель и задачи исследования, но отсутствуют их четкие формулировки. Работа является компиляцией чужих исследований с попыткой формулировки собственных выводов в конце работы. Изложение материала логично и аргументировано. Наблюдается отступление от требований в оформлении и объеме работы. При ответе на вопросы обучающийся испытывает затруднения;

- оценка «неудовлетворительно»: обнаруживается несамостоятельность выполнения курсовой работы, некомпетентность в исследуемой проблеме. Нарушена логика изложения. Работа не соответствует требованиям, предъявляемым к оформлению и содержанию. На защите курсовой работы обучающийся не отвечает на вопросы.

3.3. Задания для промежуточной аттестации (зачета и (или) экзамена)

Список вопросов к экзамену

1. Основы структурного и объектного подходов к программированию. Преимущества и недостатки объектного подхода.
2. Уровни доступа к членам класса.
3. Конструктор с параметрами. Пример программы.
4. Основные принципы ООП: абстракция и инкапсуляция.
5. Объявление объектов. Создание массивов объектов. Пример программы.
6. Конструктор со списком инициализаторов. Пример программы.
7. Понятие класса и объекта в ООП. Состояние и поведение объекта.
8. Указатели на объекты класса. Динамические объекты и массивы объектов. Пример программы.

9. Конструктор копирования. Поверхностное и глубинное копирование. Пример программы.
10. Синтаксис и семантика конструктора. Указатель `this`. Пример программы.
11. Конструктор копирования. Поверхностное и глубинное копирование. Пример программы.
12. Суть наследования и его виды. Полиморфизм.
13. Свойства конструкторов.
14. Синтаксис и семантика деструктора. Пример программы.
15. Структура класса C++. Пример программы.
16. Конструктор по умолчанию. Пример программы.
17. Свойства деструкторов.
18. Статические переменные и статические функции класса (`static`). Назначение. Пример программы.
19. Методы класса, дружественные другому классу. Пример программы.
20. Виртуальные базовые классы. Пример программы.
21. Константные переменные и функции класса (`const`). Пример программы.
22. Дружественные классы. Пример программы.
23. Замещение членов базового класса. Пример программы.
24. Функции, дружественные одному классу. Пример программы.
25. Объявление классов при простом наследовании. Спецификаторы наследуемого доступа. Пример программы.
26. Назначение перегрузки функций. Пример программы.
27. Функции, дружественные нескольким классам. Пример программы.
28. Конструкторы и деструкторы производных классов (очередность вызова). Пример программы.
29. Перегрузка конструкторов. Пример программы.
30. Назначение перегрузки операторов. Правила перегрузки операторов.
31. Механизмы раннего и позднего связывания.
32. Синтаксис шаблонного класса. Пример программы.
33. Ассоциативные контейнеры. Пример.
34. Перегрузка бинарных операторов. Пример программы.
35. Виртуальные функции. Таблица виртуальных функций. Пример программы.
36. Специализация шаблонов функций и классов. Пример программы.
37. Итераторы. Обратные итераторы. Пример.
38. Перегрузка унарных операторов. Пример программы.
39. Чисто виртуальные функции и абстрактные классы. Пример программы.
40. Достоинства и недостатки шаблонов.
41. Алгоритмы библиотеки STL. Пример.
42. Перегрузка оператора доступа к членам класса. Пример программы.
43. Назначение виртуальных деструкторов. Пример программы.
44. Назначение и состав библиотеки STL.
45. Перегрузка и неоднозначность. Пример программы.
46. Назначение шаблонных функций.
47. Последовательные контейнеры. Пример.
48. Виды и реализации полиморфизма.
49. Конкретизация шаблонной функции. Пример программы.
50. Контейнеры – адаптеры. Пример.
51. Иерархия классов VCL-библиотеки.
52. Объекты, компоненты, элементы управления VCL-библиотеки.
53. Классы командных кнопок, флажков и радиокнопок VCL-библиотеки.
54. Классы текстовых компонентов, списков и панелей VCL-библиотеки.
55. Классы стандартных диалогов VCL-библиотеки.
56. Обработка событий средствами VCL-библиотеки.

57. Структура приложения на C# и Java.
58. Создание классов и объектов C# и Java.
59. Методы-члены класса и виртуальные методы C# и Java.
60. Механизмы наследования C# и Java.
61. Классы библиотеки .NET Framework.
62. Приложения – диалоги библиотеки классов .NET Framework.

Зачеты и экзамены могут быть проведены в письменной форме, а также в письменной форме с устным дополнением ответа. Зачеты служат формой проверки качества выполнения студентами лабораторных работ, усвоения семестрового учебного материала по дисциплине (модулю), практических и семинарских занятий (при отсутствии экзамена по дисциплине).

По итогам зачета, соответствии с модульно – рейтинговой системой университета, выставляются баллы с последующим переходом по шкале баллы – оценки за зачет, выставляемый как по наименованию «зачтено», «не зачтено», так и дифференцированно т.е. с выставлением отметки по схеме – «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно», определяемое решением Ученого совета университета и прописываемого в учебном плане.

Экзамен по дисциплине (модулю) служит для оценки работы студента в течении семестра (года, всего срока обучения и др.) и призван выявить уровень, качество и систематичность полученных им теоретических и практических знаний, приобретения навыков самостоятельной работы, развития творческого мышления, умения синтезировать полученные знания и применять их в решении практических задач. По итогам экзамена, в соответствии с модульно – рейтинговой системой университета выставляются баллы, с последующим переходом по шкале оценок на оценки: «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно», свидетельствующие о приобретенных компетенциях или их отсутствии.

Форма экзаменационного билета (пример оформления)

<p><u>Министерство науки и высшего образования РФ</u></p> <p><u>ФГБОУ ВО "Дагестанский государственный технический университет"</u></p> <p>Дисциплина (модуль) <u>«Объектно-ориентированное программирование»</u></p> <p>Код, направление подготовки/специальность <u>09.03.04 Программная инженерия</u></p> <p>Профиль (программа, специализация) <u>Разработка программно-информационных систем</u></p> <p>Кафедра <u>ПОВТиАС</u> Курс <u>2</u> Семестр <u>4</u></p> <p>Форма обучения – <u>очная/заочная</u></p> <p style="text-align: center;">ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № <u>1</u></p> <p>1. Основные принципы ООП: абстракция и инкапсуляция.</p> <p>2. Конструктор копирования. Поверхностное и глубинное копирование. Пример программы.</p> <p style="text-align: right; margin-right: 100px;">Экзаменатор.....Камилова А.М.</p> <p style="text-align: right; margin-right: 100px;">Утвержден на заседании кафедры (протокол №__ от _____20__ г.)</p> <p style="text-align: right; margin-right: 100px;">Зав. кафедрой ПОВТиАС.....Айгумов Т.Г.</p>
--