

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Петровская Анна Викторовна

Должность: Директор

Дата подписания: 29.08.2025 14:35:22

Уникальный программный ключ:

798bda6555fbdebe827768f6f1710bd17a9070c31fd1b6a6ac5a1f10c8c5199

Приложение 3

к основной профессиональной образовательной программе
по направлению подготовки 09.03.03 «Прикладная информатика»
направленность (профиль) программы «Прикладная информатика в экономике»

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский экономический университет имени Г.В. Плеханова»

Краснодарский филиал РЭУ им. Г. В. Плеханова

Факультет экономики, менеджмента и торговли

Кафедра экономики и цифровых технологий

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.О.29 Основы объектно-ориентированного программирования

Направление подготовки

09.03.03 «Прикладная информатика»

Направленность (профиль) программы

Прикладная информатика в экономике

Уровень высшего образования

Бакалавриат

Год начала подготовки 2024

Краснодар – 2023 г.

Составитель:

к.т.н., доцент кафедры экономики и цифровых технологий Р.Н. Фролов

Рабочая программа одобрена на заседании кафедры экономики и цифровых технологий Краснодарского филиала РЭУ им. Г.В. Плеханова, протокол № 8 от 16.03.2023 г.

Рабочая программа составлена на основе рабочей программы по дисциплине «Основы объектно-ориентированного программирования», утвержденной на заседании базовой кафедры цифровой экономики института развития информационного общества федерального государственного бюджетного образовательного учреждения высшего образования «Российский экономический университета имени Г.В. Плеханова» протокол № 11 от 17 мая 2021 г., разработанной автором: Ивановым Е.А., ст. преподаватель, базовой кафедры Прикладной информатики и информационной безопасности

СОДЕРЖАНИЕ

I. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЙ РАЗДЕЛ.....	4
ЦЕЛЬ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	4
МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....	4
ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ.....	4
ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ.....	5
II. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	6
III. УЧЕБНО-МЕТОДИЧЕСКОЕ И МАТЕРИАЛЬНО- ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ	12
РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА	12
ПЕРЕЧЕНЬ ИНФОРМАЦИОННО-СПРАВОЧНЫХ СИСТЕМ.....	12
ПЕРЕЧЕНЬ ЭЛЕКТРОННО-ОБРАЗОВАТЕЛЬНЫХ РЕСУРСОВ.....	12
ПЕРЕЧЕНЬ ПРОФЕССИОНАЛЬНЫХ БАЗ ДАННЫХ	12
ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО- ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	12
ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	13
МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ	13
IV. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ	13
V. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ И УМЕНИЙ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ.....	14
VI. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ.. АННОТАЦИЯ К РАБОЧЕЙ ПРОГРАММЕ ДИСЦИПЛИНЫ	14 22

І. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЙ РАЗДЕЛ

Цель и задачи освоения дисциплины

Целью дисциплины «Основы объектно-ориентированного программирования» является

1. Знакомство с объектно-ориентированной парадигмой программирования.
2. Формирование фундаментальных знаний и практического опыта по основным концепциям объектно-ориентированного подхода.
3. Изучение основных понятий объектно-ориентированного подхода.
4. Изучение с практической точки зрения особенностей реализации объектно-ориентированного подхода в языке программирования (на выбор преподавателя) Си++, Java, C#.

Задачи дисциплины «Основы объектно-ориентированного программирования»:

1. Приобретение навыков проектирования объектно-ориентированных программ.
2. Приобретение навыков написания объектно-ориентированного кода программы.
3. Приобретение навыков отладки объектно-ориентированного кода.
4. Приобретение навыков работы с библиотекой универсальных шаблонов.

Место дисциплины в структуре образовательной программы

Дисциплина «Основы объектно-ориентированного программирования» относится к обязательной части учебного плана.

Объем дисциплины и виды учебной работы

Таблица 1

Показатели объема дисциплины	Всего часов по формам обучения	
	очная	заочная
Объем дисциплины в зачетных единицах	4 ЗЕТ	
Объем дисциплины в акад. часах	144	
Промежуточная аттестация: форма	экзамен, курсовой проект	экзамен, курсовой проект
Контактная работа обучающихся с преподавателем (Контакт. часы), всего:	47	24
1. Контактная работа на проведение занятий лекционного и семинарского типов, всего часов, в том числе:	40	18
• лекции	12	6
• практические занятия	28	12
• лабораторные занятия	-	-
в том числе практическая подготовка	-	-
2. Индивидуальные консультации (ИК)	1	1
3. Контактная работа по промежуточной аттестации (Катт)	2	1
4. Консультация перед экзаменом (КЭ)	2	2
5. Контактная работа по промежуточной	2	2

аттестации в период экз. сессии / сессии заочников (Каттэк)		
Самостоятельная работа (СР), всего:	95	120
в том числе:		
• самостоятельная работа в период экз. сессии (СРэк)	32	5
• в том числе самостоятельная работа на курсовую работу (курсовой проект)	18	18
• самостоятельная работа в семестре (СРс)	47	97
• изучение ЭОР (<i>при наличии</i>)	-	-
• изучение онлайн-курса или его части	-	-
• выполнение индивидуального проекта	-	-

Перечень планируемых результатов обучения по дисциплине

Таблица 2

Формируемые компетенции (код и наименование компетенции)	Индикаторы достижения компетенций (код и наименование индикатора)	Результаты обучения (знания, умения)
ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.1. Разрабатывает и реализует эффективные алгоритмы и программы с использованием современных технологий и языков программирования	ОПК-7.1. 3-2. Знает основные языки программирования и современные программные среды разработки информационных систем и технологий ОПК-7.1. 3-3. Знает современные методы разработки и реализации алгоритмов решения прикладных задач с использованием различных языков программирования ОПК-7.1. 3-4. Знает технологию разработки и отладки программ ОПК-7.1. 3-5. Знает синтаксис и семантику языков программирования, виды вычислительных процессов, типы данных
		ОПК-7.1. У-1. Умеет выбирать и применять современные языки программирования для реализации алгоритмов и написания прикладных программ в зависимости от типа прикладной задачи ОПК-7.1. У-2. Умеет программировать приложения и создавать программные прототипы решения прикладных задач ОПК-7.1. У-3. Умеет разрабатывать программный код в современной среде программирования, анализировать и адаптировать под нужды заказчика

II. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

этапы формирования и критерии оценивания сформированности компетенций
для обучающихся очной формы обучения

Таблица 3.1

№ п/п	Наименование раздела, темы дисциплины	Трудоемкость, академические часы						Индикаторы достижения компетенций	Результаты обучения (знания, умения)	Учебные задания для ауди- торных занятий	Текущий контроль	Задания для творческого рейтинга (по теме(-ам)/ раз- делу или по всему курсу в це- лом)
		Лекции	Практические занятия	Лабораторные занятия	Практическая подго- товка	Самостоятельная работа/ КЭ, Каттэк, Катт	Всего					
Семестр 2												
Раздел 1. Основы объектно-ориентированного программирования												
1.	Тема 1. Класс как абстрактный тип. Конструкторы и деструкторы <i>Введение, основные понятия; Аб- страктные типы данных; Простран- ства имен, операция “::”; Синтаксис описания класса; Управление доступом к членам класса; Классы и структуры C++; Класс как область видимости; Объявление и определение методов класса. Конструкторы и деструкто- ры; Конструктор умолчания; Кон- структор преобразования и кон- структоры с двумя и более парамет- рами; Конструктор копирования; Спе- цификатор explicit; Конструктор ко- пирования и операция присваивания; Автоматическая генерация конструкторов и деструкторов; Список иници-</i>	2	4	-	-	6	12	ОПК-7.1	ОПК-7.1. 3-2 ОПК-7.1. 3-3 ОПК-7.1. 3-4 ОПК-7.1. 3-5 ОПК-7.1. У-1 ОПК-7.1. У-2 ОПК-7.1. У-3	О.	Т.	Ин.п.

	ализации; Порядок вызова конструкторов и деструкторов.											
2.	Тема 2. Члены класса, статический полиморфизм <i>Статические члены класса; Константные члены класса; Модификатор const; Дружественные классы; Статический полиморфизм; Перегрузка бинарных операций; Перегрузка унарных операций; Перегрузка функций.</i>	2	4	-	-	6	12	ОПК-7.1	ОПК-7.1. 3-2 ОПК-7.1. 3-3 ОПК-7.1. 3-4 ОПК-7.1. 3-5 ОПК-7.1. У-1 ОПК-7.1. У-2 ОПК-7.1. У-3	О.	Т.	Ин.п.
3.	Тема 3. Отношения между классами <i>Ассоциация; ER-диаграммы; Одиночное наследование; Правила наследования; Преобразования указателей; Правила видимости при наследовании; Закрытое наследование; Перекрывание имен; Наследование и повторное использование кода.</i>	2	4	-	-	7	13	ОПК-7.1	ОПК-7.1. 3-2 ОПК-7.1. 3-3 ОПК-7.1. 3-4 ОПК-7.1. 3-5 ОПК-7.1. У-1 ОПК-7.1. У-2 ОПК-7.1. У-3	Гр.д.	К/р	Ин.п.
4.	Тема 4. Динамический полиморфизм, механизм виртуальных функций. Средства обработки ошибок и исключений <i>Виртуальные деструкторы; Реализация виртуальных функций; Абстрактные классы; Чистые виртуальные функции; Правила выбора обработчика исключения; Стандартные исключения; Последовательность действий при возникновении исключительной ситуации.</i>	2	6	-	-	10	18	ОПК-7.1	ОПК-7.1. 3-2 ОПК-7.1. 3-3 ОПК-7.1. 3-4 ОПК-7.1. 3-5 ОПК-7.1. У-1 ОПК-7.1. У-2 ОПК-7.1. У-3	О.	Т.	Ин.п.
5.	Тема 5. Множественное наследование, интерфейсы, динамическая информация о типе, параметрический полиморфизм <i>Видимость при множественном</i>	2	4	-	-	18	24	ОПК-7.1	ОПК-7.1. 3-2 ОПК-7.1. 3-3 ОПК-7.1. 3-4 ОПК-7.1. 3-5 ОПК-7.1. У-1 ОПК-7.1. У-2	О.	К/р	Ин.п.

	наследовании; Виртуальные базовые классы; Интерфейсы; RTTI; Динамическое приведение типа (dynamic_cast); Параметры шаблона; Шаблоны функций; Специализация шаблонной функции; Шаботонные классы; Эквивалентность типов.								ОПК-7.1. У-3			
6.	Тема 6. Библиотека универсальных шаблонов Контейнеры; Распределители памяти; Итераторы; Алгоритмы; Достоинства и недостатки STL-подхода.	2	6	-	-	18	26	ОПК-7.1	ОПК-7.1. 3-2 ОПК-7.1. 3-3 ОПК-7.1. 3-4 ОПК-7.1. 3-5 ОПК-7.1. У-1 ОПК-7.1. У-2 ОПК-7.1. У-3	Гр.д.	Т.	Ин.п.
	Итого	12	28	-	-	57	105					
	Консультация перед экзаменом (КЭ)	-	-	-	-	-/2	2	-	-	-	-	-
	Контактная работа по промежуточной аттестации в период экз. сессии / сессии заочников (Каттэк)	-	-	-	-	-/2	2	-	-	-	-	-
	Индивидуальные консультации (ИК)					-/1	1					
	Контактная работа по промежуточной аттестации (Катт)					-/2	2					
	Самостоятельная работа в период экз. сессии (СРэк)	-	-	-	-	32/-	32	-	-	-	-	-
	Итого	12	28	-	-	79/7	144	х	х	х	х	х

**этапы формирования и критерии оценивания сформированности компетенций
для обучающихся заочной формы обучения**

Таблица 3.2

№ п/п	Наименование раздела, темы дисциплины	Трудоемкость, академические часы						Индикаторы достижения компетенций	Результаты обучения (знания, умения)	Учебные задания для ауди- торных занятий	Текущий контроль	Задания для творческого рейтинга (по теме(-ам)/ раз- делу или по всему курсу в це- лом)
		Лекции	Практические занятия	Лабораторные занятия	Практическая подго- товка	Самостоятельная работа/ КЭ, Катгэк, Катг	Всего					
Семестр 2												
Раздел 1. Основы объектно-ориентированного программирования												
1.	Тема 1. Класс как абстрактный тип. Конструкторы и деструкторы <i>Введение, основные понятия; Аб- страктные типы данных; Простран- ства имен, операция “::”; Синтаксис описания класса; Управление доступом к членам класса; Классы и структуры C++; Класс как область видимости; Объявление и определение методов класса. Конструкторы и деструкто- ры; Конструктор умолчания; Кон- структор преобразования и кон- структоры с двумя и более парамет- рами; Конструктор копирования; Спе- цификатор explicit; Конструктор ко- пирования и операция присваивания; Автоматическая генерация конструкторов и деструкторов; Список иници- ализации; Порядок вызова конструкторов</i>	1	2	-	-	16	19	ОПК-7.1	ОПК-7.1. 3-2 ОПК-7.1. 3-3 ОПК-7.1. 3-4 ОПК-7.1. 3-5 ОПК-7.1. У-1 ОПК-7.1. У-2 ОПК-7.1. У-3	О.	Т.	Ин.п.

	<i>торов и деструкторов.</i>											
2.	Тема 2. Члены класса, статический полиморфизм <i>Статические члены класса; Константные члены класса; Модификатор const; Дружественные классы; Статический полиморфизм; Перегрузка бинарных операций; Перегрузка унарных операций; Перегрузка функций.</i>	1	2	-	-	16	19	ОПК-7.1	ОПК-7.1. 3-2 ОПК-7.1. 3-3 ОПК-7.1. 3-4 ОПК-7.1. 3-5 ОПК-7.1. У-1 ОПК-7.1. У-2 ОПК-7.1. У-3	О.	Т.	Ин.п.
3.	Тема 3. Отношения между классами <i>Ассоциация; ER-диаграммы; Одиночное наследование; Правила наследования; Преобразования указателей; Правила видимости при наследовании; Закрытое наследование; Перекрывание имен; Наследование и повторное использование кода.</i>	1	2	-	-	16	19	ОПК-7.1	ОПК-7.1. 3-2 ОПК-7.1. 3-3 ОПК-7.1. 3-4 ОПК-7.1. 3-5 ОПК-7.1. У-1 ОПК-7.1. У-2 ОПК-7.1. У-3	Гр.д.	К/р	Ин.п.
4.	Тема 4. Динамический полиморфизм, механизм виртуальных функций. Средства обработки ошибок и исключений <i>Виртуальные деструкторы; Реализация виртуальных функций; Абстрактные классы; Чистые виртуальные функции; Правила выбора обработчика исключения; Стандартные исключения; Последовательность действий при возникновении исключительной ситуации.</i>	1	2	-	-	19	22	ОПК-7.1	ОПК-7.1. 3-2 ОПК-7.1. 3-3 ОПК-7.1. 3-4 ОПК-7.1. 3-5 ОПК-7.1. У-1 ОПК-7.1. У-2 ОПК-7.1. У-3	О.	Т.	Ин.п.
5.	Тема 5. Множественное наследование, интерфейсы, динамическая информация о типе, параметрический полиморфизм <i>Видимость при множественном наследовании; Виртуальные базовые</i>	1	2	-	-	24	27	ОПК-7.1	ОПК-7.1. 3-2 ОПК-7.1. 3-3 ОПК-7.1. 3-4 ОПК-7.1. 3-5 ОПК-7.1. У-1 ОПК-7.1. У-2 ОПК-7.1. У-3	О.	К/р	Ин.п.

	классы; Интерфейсы; RTTI; Динамическое приведение типа (dynamic_cast); Параметры шаблона; Шаблоны функций; Специализация шаблонной функции; Шаблоны классов; Эквивалентность типов.											
6.	Тема 6. Библиотека универсальных шаблонов Контейнеры; Распределители памяти; Итераторы; Алгоритмы; Достоинства и недостатки STL-подхода.	1	2	-	-	23	26	ОПК-7.1	ОПК-7.1. 3-2 ОПК-7.1. 3-3 ОПК-7.1. 3-4 ОПК-7.1. 3-5 ОПК-7.1. У-1 ОПК-7.1. У-2 ОПК-7.1. У-3	Гр.д.	Т.	Ин.п.
	Итого	6	12	-	-	114	126					
	Консультация перед экзаменом (КЭ)	-	-	-	-	-/2	2	-	-	-	-	-
	Контактная работа по промежуточной аттестации в период экз. сессии / сессии заочников (Каттэк)	-	-	-	-	-/2	2	-	-	-	-	-
	Индивидуальные консультации (ИК)					-/1	1					
	Контактная работа по промежуточной аттестации (Катт)					-/2	2					
	Самостоятельная работа в период экз. сессии (СРэк)	-	-	-	-	5/-	5	-	-	-	-	-
	Итого	6	12	-	-	119/7	144	х	х	х	х	х

Формы учебных заданий на аудиторных занятиях: Опрос (О.), Групповая дискуссия (Гр.д.)

Формы текущего контроля: Тест (Т.), Контрольные работы (К/р)

Формы заданий для творческого рейтинга: Индивидуальный проект (Ин.п.)

III. УЧЕБНО-МЕТОДИЧЕСКОЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Основная литература:

1. Литвиненко, В. А. Основы объектно-ориентированного программирования задач на графах : учебное пособие / В. А. Литвиненко ; Южный федеральный университет. - Ростов-на-Дону ; Таганрог: Издательство Южного федерального университета, 2019. - 133 с. - ISBN 978-5-9275-3472-2. - Текст: электронный. - URL: <https://znanium.com/read?id=375021>

Дополнительная литература:

1. Объектно-ориентированное программирование Бакалавриат: учебное пособие / Николаев Е.И. — Ставрополь : Северо-Кавказский федеральный университет, 2015. — 225 с. — Режим доступа: <https://book.ru/book/928784>
2. Объектно-ориентированное программирование с примерами на C# : учеб. пособие / П.Б. Хорев. — М. : ФОРУМ : ИНФРА-М, 2019. — 200 с. — (Высшее образование: Бакалавриат). - Режим доступа: <https://znanium.com/read?id=339308>

Нормативные правовые документы:

1. ГОСТ Р 54593-2011. Информационные технологии. Свободное программное обеспечение. Общие положения.
2. ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения.
3. ГОСТ Р 56920-2016. Системная и программная инженерия. Тестирование программного обеспечения. Часть 1. Понятия и определения

ПЕРЕЧЕНЬ ИНФОРМАЦИОННО-СПРАВОЧНЫХ СИСТЕМ

1. <https://www.consultant.ru> -Справочно-правовая система Консультант Плюс;
2. <https://www.garant.ru/> - Справочно-правовая система Гарант.

ПЕРЕЧЕНЬ ПРОФЕССИОНАЛЬНЫХ БАЗ ДАННЫХ

1. <https://zbmath.org> - Математическая база данных, охватывающая материалы с конца 19 века.
2. <http://pravo.gov.ru/ips/> - БД «Законодательство России»

ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ "ИНТЕРНЕТ", НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. <http://cppstudio.com/cat/274/> (Язык программирования C++)
2. <http://cppstudio.com/cat/293/> (Алгоритмы и структуры данных в C++)
3. <https://prog-cpp.ru/category/cpp-posts/> (Язык C++)

ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Лицензионное программное обеспечение:

- Операционная система Windows 10, Microsoft Office Professional Plus: 2019 год (MS Word, MS Excel, MS Power Point, MS Access)
- Антивирусная программа Касперского Kaspersky Endpoint Security для бизнеса Расширенный Rus Edition

Свободно распространяемое программное обеспечение:

- Visual Studio community

МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Дисциплина «Основы объектно-ориентированного программирования» обеспечена:

для проведения занятий лекционного типа:

- учебной аудиторией, оборудованной учебной мебелью, мультимедийными средствами обучения для демонстрации лекций-презентаций;

для проведения занятий семинарского типа (*практические занятия*):

- компьютерным классом;
для самостоятельной работы, в том числе для курсового проектирования;
- помещением для самостоятельной работы, оснащенным компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к электронной информационно-образовательной среде университета.

IV. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

- Положение о курсовых работах (проектах) в ФГБОУ ВО "РЭУ им. Г.В. Плеханова".
- Методические указания по проведению практических занятий и организации самостоятельной работы обучающихся

V. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ И УМЕНИЙ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ

Результаты текущего контроля и промежуточной аттестации формируют рейтинговую оценку работы обучающегося. Распределение баллов при формировании рейтинговой оценки работы обучающегося осуществляется в соответствии с «Положением о рейтинговой системе оценки успеваемости и качества знаний студентов в процессе освоения дисциплины **«Основы объектно-ориентированного программирования»** в федеральном государственном бюджетном образовательном учреждении высшего образования «Российский экономический университет имени Г.В. Плеханова».

Таблица 4

Виды работ	Максимальное количество баллов
Выполнение учебных заданий на аудиторных занятиях	20
Текущий контроль	20
Творческий рейтинг	20
Промежуточная аттестация (экзамен)	40
ИТОГО	100

В соответствии с Положением о рейтинговой системе оценки успеваемости и качества знаний обучающихся «преподаватель кафедры, непосредственно ведущий занятия со студенческой группой, обязан проинформировать группу о распределении рейтинговых баллов по всем видам работ на первом занятии учебного модуля (семестра), количестве модулей по учебной дисциплине, сроках и формах контроля их освоения, форме промежуточной аттестации, снижении баллов за несвоевременное выполнение выданных заданий. Обучающиеся в течение учебного модуля (семестра) получают информацию о текущем количестве набранных по дисциплине баллов через личный кабинет студента».

VI. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Оценочные материалы по дисциплине разработаны в соответствии с Положением об оценочных материалах в федеральном государственном бюджетном образовательном учреждении высшего образования «Российский экономический университет имени Г.В. Плеханова».

Тематика курсовых проектов

1. Объектно-ориентированный подход к разбору и вычислению арифметических выражений.
2. Объектно-ориентированный подход к разбору и обработке XML.
3. Задача коммивояжера. Реализация задачи с применением ООП.
4. Векторный графический редактор с применением ООП.
5. Реализация компьютерной игры, основанной на объектах.
6. Реализация генетического алгоритма на примере игры «жизнь» с применением ООП.
7. Класс для осуществления операций над матрицами.
8. Класс для работы с векторами в трехмерном пространстве.
9. Система управления местами в гостинице с применением ООП.
10. Система учета пациентов больницы с применением ООП.
11. Система управления складом аптеки с применением ООП.
12. Система учета заказов клиентов ресторана с применением ООП.
13. Построение графиков функций с применением ООП.
14. Объектно-ориентированная модель Аквариума.
15. Задача Прима-Краскала.
16. Модель солнечной системы, на основе ООП.

Перечень вопросов к экзамену:

1. Введение в ООП, основные понятия.

2. Абстрактные типы данных; Пространства имен, операция “::”.
3. Синтаксис описания класса. Управление доступом к членам класса.
4. Класс как область видимости; Объявление и определение методов класса.
5. Конструкторы и деструкторы; Конструктор умолчания
6. Конструктор преобразования и конструкторы с двумя и более параметрами
7. Конструктор копирования; Спецификатор `explicit`
8. Конструктор копирования и операция присваивания
9. Автоматическая генерация конструкторов и деструкторов
10. Список инициализации; Порядок вызова конструкторов и деструкторов.
11. Статические члены класса. Константные члены класса.
12. Модификатор `const`.
13. Дружественные классы.
14. Статический полиморфизм.
15. Перегрузка бинарных операций.
16. Перегрузка унарных операций.
17. Перегрузка функций.
18. Ассоциация. ER-диаграммы.
19. Одиночное наследование.
20. Правила наследования.
21. Преобразования указателей.
22. Правила видимости при наследовании.
23. Закрытое наследование.
24. Перекрывание имен.
25. Наследование и повторное использование кода.
26. Виртуальные деструкторы.
27. Реализация виртуальных функций;
28. Абстрактные классы.
29. Чистые виртуальные функции;
30. Правила выбора обработчика исключения;
31. Стандартные исключения.
32. Последовательность действий при возникновении исключительной ситуации.
33. Видимость при множественном наследовании.
34. виртуальные базовые классы;
35. Интерфейсы; RTTI;
36. Динамическое приведение типа (`dynamic_cast`);
37. Параметры шаблона;
38. Шаблоны функций;
39. Специализация шаблонной функции;
40. Шаблонные классы;
41. Эквивалентность типов.
42. Контейнеры;
43. Распределители памяти;
44. STL: контейнеры
45. STL: итераторы
46. STL: алгоритмы
47. Достоинства и недостатки STL-подхода.
48. Классы и структуры `C++`
49. Указатель `this`
50. Указатель на член класса

51. Конструктор умолчания
52. Конструктор преобразования
53. Конструктор копирования
54. Список инициализации
55. Деструкторы, порядок вызова конструкторов и деструкторов
56. Статические члены класса
57. Константные члены класса
58. Дружественные классы
59. Перегрузка операций
60. Правила наследования
61. Правила видимости при наследовании
62. Перекрытие имен при наследовании
63. Виртуальные функции
64. Виртуальные деструкторы
65. Абстрактные классы, чистые виртуальные функции
66. Обработка исключений
67. Виртуальные базовые классы
68. Видимость при множественном наследовании
69. Интерфейсы
70. Динамическая информация о типе
71. Шаблоны функций
72. Шаблонные классы

Практические задания для экзамена:

1. Создать структуру, которая будет содержать данные о студентах института (фамилия, имя, группа, средний балл). С помощью функции-члена класса необходимо вывести на экран фамилию, имя студента и номер группы. Создать аналогичный класс.
2. Создать структуру, которая будет содержать данные о книге (название, издательство, автор, количество страниц).
3. С помощью функции-члена класса необходимо вывести на экран название, издательство и автора книги. Создать аналогичный класс.
4. Написать программу, вычисляющую с помощью функции члена класса средний балл студента за сессию. Количество экзаменов является величиной переменной и задается пользователем. Список студентов обрабатывается как класс.
5. Написать программу, вычисляющую с помощью функции члена класса суммарную величину продаж книжного магазина. Количество продаж является величиной переменной и задается пользователем. Список всех продаж, состоящий из товара и цены, обрабатывается как класс.

Тестовые задания:

- 1.1. Что называется конструктором?
 - а) метод, имя которого совпадает с именем класса и который вызывается автоматически при создании объекта класса
 - б) метод, имя которого совпадает с именем класса и который вызывается автоматически при объявлении класса (до создания объекта класса)

- в) метод, имя которого необязательно совпадает с именем класса и который вызывается при создании объекта класса
- г) метод, имя которого совпадает с именем класса и который необходимо явно вызывать из головной программы при объявлении объекта класса

1.2. Объект - это

- а) переменная, содержащая указатель на класс
- б) экземпляр класса
- в) класс, который содержит в себе данные и методы их обработки

1.3. Выберите правильное объявление производного класса

- а) `class MoreDetails::Details;`
- б) `class MoreDetails: public class Details;`
- в) `class MoreDetails: public Details;`
- г) `class MoreDetails: class(Details)`

1.4. Возможность и способ обращения производного класса к элементам базового определяется

- а) ключами доступа: `private`, `public`, `protected` в теле производного класса
- б) только ключом доступа `protected` в заголовке объявления производного класса
- в) ключами доступа: `private`, `public`, `protected` в заголовке объявления производного класса
- г) ключами доступа: `private`, `public`, `protected` в теле базового класса

1.5. Дружественная функция - это

- а) функция другого класса, среди аргументов которой есть элементы данного класса
- б) функция, объявленная в классе с атрибутом `friend`, но не являющаяся членом класса;
- в) функция, являющаяся членом класса и объявленная с атрибутом `friend`;
- г) функция, которая в другом классе объявлена как дружественная данному

1.6. Переопределение операций имеет вид:

- а) имя_класса, ключевое слово `operation`, символ операции
- б) имя_класса, ключевое слово `operator`, символ операции, в круглых скобках могут быть указаны аргументы
- в) имя_класса, ключевое слово `operator`, список аргументов
- г) имя_класса, два двоеточия, ключевое слово `operator`, символ операции

1.7. Существует ли в C++ готовый набор шаблонов:

- а) нет
- б) да, существует специальная библиотека STL
- в) зависит от версии компилятора

Примеры вопросов для опроса:

1. Какие компоненты базового класса присутствуют в производном классе?
2. Что происходит при вызове виртуальной функции через полиморфный объект?
3. Что такое абстрактный класс?
4. Каким образом интерфейс может решить проблему неоднозначности?
5. Какие основные операции можно реализовать с помощью итераторов?

Примеры тем групповых дискуссий:

1. Доступность членов базового класса и модификаторы доступа при наследовании.
2. Приведите примеры, в которых может потребоваться наследование.
3. Стали бы вы использовать шаблоны в своих программах? Почему?
4. Есть ли смысл писать свой класс для работы с деревьями, если он уже есть в STL?

Примеры заданий для контрольной работы:

Задание 1.

Разработать класс для рисования графических примитивов с названием Canvas.

В конструкторе класса должен создаваться связный список (можно использовать класс Vector из STL), в который будут добавляться объекты классов графических примитивов и их тип

1. Точка
2. Отрезок
3. Эллипс (+ свойство закрашен/не закрашен)
4. Прямоугольник (+ свойство закрашен/не закрашен)
5. Многоугольник (+ свойство закрашен/не закрашен)

Эллипс и прямоугольник задаются координатами левого верхнего и правого нижнего углов (для эллипса – это прямоугольник в который он вписан)

Координаты начинаются в левом верхнем углу.

В деструкторе список объектов должен уничтожаться.

Методы:

1. Добавление объекта в список
2. Удаление объекта из списка
3. Очистка списка
4. Вывод информации об объектах находящихся в списке
5. Метод Show для построения изображения из объектов списка (будет сделан в конце реализации класса)

Для каждого примитива должен быть разработан соответствующий класс.

Все классы (кроме класса точка) должны использовать класс точка для хранения координат точек.

Описание классов сделать в файле Canvas.h

Реализацию методов классов сделать в файле Canvas.cpp

В основном файле подключить модуль Canvas.h и создать объекты.

Тематика индивидуальных проектов:

Задание 1.

Разработать класс "Сотрудник" с закрытыми информационными членами.

Информационные члены класса: Место работы, Должность, Стаж, Личные данные (Ф.И.О, пол, возраст), Заработная плата (накопление на счете, сотрудник её не снимает)

Действия над объектами класса: получение информации о месте работы, занимаемой должности, стаже работы, заработной плате; изменение должности; начисление заработной платы; вывод личных данных; операции сравнения объектов; присваивание.

Статический информационный член класса: Количество сотрудников.

Статический метод: Получение количества сотрудников

На основе класса создать массив из объектов (от 2 до 5), заполнить и вывести информацию о количестве объектов и о каждом в отдельности.

Задание 5.

Разработать иерархию классов для нахождения корней квадратного уравнения $ax^2 + bx + c = 0$
 А - базовый класс для класса В, В базовый для класса С.

Компоненты классов А,В,С - коэффициенты a,b,c. Корни уравнения и методы решения находятся в классе С.

Структура экзаменационного билета

Наименование оценочного средства	Максимальное количество баллов
Вопрос 1	13
Вопрос 2	13
Практическое задание	14

Показатели и критерии оценивания планируемых результатов освоения компетенций и результатов обучения, шкала оценивания

Таблица 5

Шкала оценивания		Формируемые компетенции	Индикатор достижения компетенции	Критерии оценивания	Уровень освоения компетенций
85 – 100 баллов	«отлично»	ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.1. Разрабатывает и реализует эффективные алгоритмы и программы с использованием современных технологий и языков программирования	Знает верно и в полном объеме: основные языки программирования и современные программные среды разработки информационных систем и технологий; современные методы разработки и реализации алгоритмов решения прикладных задач с использованием различных языков программирования; технологию разработки и отладки программ; синтаксис и семантику языков программирования, виды вычислительных процессов, типы данных. Умеет верно и в полном объеме: выбирать и применять со-	Продвинутый

				временные языки программирования для реализации алгоритмов и написания прикладных программ в зависимости от типа прикладной задачи; программировать приложения и создавать программные прототипы решения прикладных задач; разрабатывать программный код в современной среде программирования, анализировать и адаптировать под нужды заказчика.	
70 – 84 баллов	«хорошо»	ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.1. Разрабатывает и реализует эффективные алгоритмы и программы с использованием современных технологий и языков программирования	Знает с незначительными замечаниями: основные языки программирования и современные программные среды разработки информационных систем и технологий; современные методы разработки и реализации алгоритмов решения прикладных задач с использованием различных языков программирования; технологию разработки и отладки программ; синтаксис и семантику языков программирования, виды вычислительных процессов, типы данных. Умеет с незначительными замечаниями: выбирать и применять современные языки программирования для реализации алгоритмов и написания прикладных программ в зависимости от типа прикладной задачи; программировать приложения и создавать программные прототипы решения прикладных задач; разрабатывать программный код в современной среде программирования, анализировать и адаптировать под нужды заказчика.	Повышенный
50 – 69 баллов	«удовлетворительно»	ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.1. Разрабатывает и реализует эффективные алгоритмы и программы с использованием современных технологий и языков программирования	Знает на базовом уровне, с ошибками: основные языки программирования и современные программные среды разработки информационных систем и технологий; современные методы разработки и реализации алгоритмов решения прикладных задач с использованием различных языков программирования; технологию разработки и отладки программ; синтаксис и семантику языков программирования, виды вычислительных процессов, типы данных.	Базовый

				Умеет на базовом уровне, с ошибками: выбирать и применять современные языки программирования для реализации алгоритмов и написания прикладных программ в зависимости от типа прикладной задачи; программировать приложения и создавать программные прототипы решения прикладных задач; разрабатывать программный код в современной среде программирования, анализировать и адаптировать под нужды заказчика.	
менее 50 баллов	«неудовлетворительно»	ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.1. Разрабатывает и реализует эффективные алгоритмы и программы с использованием современных технологий и языков программирования	Не знает на базовом уровне: основные языки программирования и современные программные среды разработки информационных систем и технологий; современные методы разработки и реализации алгоритмов решения прикладных задач с использованием различных языков программирования; технологию разработки и отладки программ; синтаксис и семантику языков программирования, виды вычислительных процессов, типы данных. Не умеет на базовом уровне: выбирать и применять современные языки программирования для реализации алгоритмов и написания прикладных программ в зависимости от типа прикладной задачи; программировать приложения и создавать программные прототипы решения прикладных задач; разрабатывать программный код в современной среде программирования, анализировать и адаптировать под нужды заказчика.	Компетенции не сформированы

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский экономический университет имени Г.В. Плеханова»
Институт математики, информационных систем и цифровой экономики
Факультет экономики, менеджмента и торговли
Кафедра экономики и цифровых технологий

АННОТАЦИЯ К РАБОЧЕЙ ПРОГРАММЕ ДИСЦИПЛИНЫ

Б1.О.29 Основы объектно-ориентированного программирования

Направление подготовки	09.03.03 «Прикладная информатика»
Направленность (профиль) программы	Прикладная информатика в экономике
Уровень высшего образования	<i>Бакалавриат</i>

Краснодар – 2023 г.

1. Цель и задачи дисциплины:

Целью дисциплины «Основы объектно-ориентированного программирования» является:

1. Знакомство с объектно-ориентированной парадигмой программирования.
2. Формирование фундаментальных знаний и практического опыта по основным концепциям объектно-ориентированного подхода.
3. Изучение основных понятий объектно-ориентированного подхода.
4. Изучение с практической точки зрения особенностей реализации объектно-ориентированного подхода в языке программирования (на выбор преподавателя) Си++, Java, C#.

Задачи дисциплины «Основы объектно-ориентированного программирования»:

1. Приобретение навыков проектирования объектно-ориентированных программ.
2. Приобретение навыков написания объектно-ориентированного кода программы.
3. Приобретение навыков отладки объектно-ориентированного кода.
4. Приобретение навыков работы с библиотекой универсальных шаблонов.

2. Содержание дисциплины:

№ п/п	Наименование разделов / тем дисциплины
1.	Тема 1. Класс как абстрактный тип. Конструкторы и деструкторы
2.	Тема 2. Члены класса, статический полиморфизм
3.	Тема 3. Отношения между классами
4.	Тема 4. Динамический полиморфизм, механизм виртуальных функций. Средства обработки ошибок и исключений
5.	Тема 5. Множественное наследование, интерфейсы, динамическая информация о типе, параметрический полиморфизм
6.	Тема 6. Библиотека универсальных шаблонов
Трудоемкость дисциплины составляет 43.е. / 144 часов.	

Форма контроля – экзамен, курсовой проект

Составитель:

к.т.н., доцент Р.Н. Фролов