Подписано электронной подписью:
Вержицкий Данил Григорьевич
Должность: Директор КГПИ КемГУ
Дата и время: 2025-04-23 00:00:00
471086fad29a3b30e244c728abc3661ab35c9d50210dcf0e75e03a5b6fdf6436

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Кузбасский гуманитарно-педагогический институт

Факультет информатики, математики и экономики

УТВЕРЖДАЮ
Декан
А.В. Фомина
«08» февраля 2024 г.

Рабочая программа дисциплины

К.М.04.04 Объектно-ориентированное проектирование и программирование

Направление подготовки
01.03.02 Прикладная математика и информатика

Направленность (профиль) подготовки
ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ

Программа бакалавриата

Квалификация выпускника
бакалавр

Форма обучения
Очная

Год набора 2024

Новокузнецк 2024

Оглавление

1 Цель дисциплины. 3

Формируемые компетенции, индикаторы достижения компетенций, знания, умения, навыки 3

Место дисциплины....3

2 Объём и трудоёмкость дисциплины по видам учебных занятий. Формы промежуточной аттестации. 3

3. Учебно-тематический план и содержание дисциплины. 3

3.1 Учебно-тематический план 3

4 Порядок оценивания успеваемости и сформированности компетенций обучающегося в текущей и промежуточной аттестации. 4

5 Материально-техническое, программное и учебно-методическое обеспечение дисциплины. 5

5.1 Учебная литература 5

5.2 Материально-техническое и программное обеспечение дисциплины....5

5.3 Современные профессиональные базы данных и информационные справочные системы....6

6 Иные сведения и (или) материалы....6

6.1. Примерные вопросы и задания для промежуточной аттестации 6

1 Цель дисциплины.

В результате освоения данной дисциплины у обучающегося должны быть сформированы компетенции основной профессиональной образовательной программы бакалавриата (далее - ОПОП): ПК-2.

Формируемые компетенции, индикаторы достижения компетенций, знания, умения, навыки

Таблица 1 – Индикаторы достижения компетенций, формируемые дисциплиной

Код и название компетенции	Индикаторы достижения компетенции по ОПОП	Знания, умения, навыки (ЗУВ), формируемые дисциплиной
ПК-2 Способен разрабатывать спецификации требований, проектировать и реализовывать программное обеспечение	2.2 Проектирует программное обеспечение 2.3 Разрабатывает программное обеспечение	**Знать:** - теоретические аспекты проектирования и разработки программного обеспечения с использованием объектного подхода. **Уметь:** - применять существующие паттерны проектирования для проектирования и разработки программных приложений. **Владеть:** - навыками проектирования и разработки программных приложений с использованием объектного подхода.

Место дисциплины

Дисциплина включена в модуль «Программирование» ОПОП ВО. Дисциплина осваивается на 2 курсе в 3 семестре.

2 Объём и трудоёмкость дисциплины по видам учебных занятий. Формы промежуточной аттестации.

Таблица 2 – Объем и трудоемкость дисциплины по видам учебных занятий

Общая трудоемкость и виды учебной работы по дисциплине, проводимые в разных формах	Объём часов по формам обучения
	ОФО
1 Общая трудоемкость дисциплины	180
2 Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)	74
Аудиторная работа (всего):	74
в том числе:	
лекции	18
лабораторные работы	56
Внеаудиторная работа (всего):	
3 Самостоятельная работа обучающихся (всего)	70
4 Промежуточная аттестация обучающегося – экзамен (3 семестр)	36

3. Учебно-тематический план и содержание дисциплины.

3.1 Учебно-тематический план

Таблица 3 - Учебно-тематический план очной формы обучения

№ недели п/п	Разделы и темы дисциплины по занятиям	Общая трудоёмкость *(всего час.)*	Трудоемкость занятий (час.)		Формы текущего контроля и промежуточной аттестации успеваемости
			ОФО		
			Аудиторн. занятия	СРС	

			лекц.	лаб.		
Семестр 3						
	1. Объектно-ориентированное моделирование	**24**	**4**	**8**	**12**	Контрольная работа №1
1	1.1 Объектная модель проектирования	12	2	4	6	Защита отчета по ЛР № 1
2	1.2 Инструментальные средства проектирования объектной системы	12	2	4	6	Защита отчета по ЛР № 2
	2. Объектно-ориентированное программирование	**54**	**6**	**22**	**26**	Контрольная работа №2
3	2.1 Классы и объекты	9	1	4	4	Защита отчета по ЛР № 3
4	2.2 Методы и механизмы наследования. Полиморфизм	9	1	4	4	Защита отчетов по ЛР № 4-5
5	2.3 Параметризация объектов в ООП. Использование параметризированных классов	9	1	4	4	Защита отчета по ЛР № 6
6	2.4 Обработка исключений	11	1	4	6	Защита отчета по ЛР № 7
7	2.5 Потоки ввода/вывода, организация работы с файлами	9	1	4	4	Защита отчета по ЛР № 8
8	2.6 Контейнерные типы	7	1	2	4	Защита отчета по ЛР № 9
	3. Паттерны проектирования	**66**	**8**	**26**	**32**	Контрольная работа №3
9	3.1 Классификация паттернов	8	2		6	
10	3.2 Порождающие паттерны	14	2	8	4	Защита отчетов по ЛР № 10-11
11	3.3 Структурные паттерны	20	2	8	10	Защита отчетов по ЛР № 12-13
12	3.4 Паттерны поведения	24	2	10	12	Защита отчетов по ЛР № 14-16
	Промежуточная аттестация – *экзамен*	36				36
ИТОГО по семестру 3		**180**	**18**	**56**	**70**	**36**

4 Порядок оценивания успеваемости и сформированности компетенций обучающегося в текущей и промежуточной аттестации.

Для положительной оценки по результатам освоения дисциплины обучающемуся необходимо выполнить все установленные виды учебной работы. Оценка результатов работы обучающегося в баллах (по видам) приведена в таблице 4.

Таблица 4 - Балльно-рейтинговая оценка результатов учебной работы обучающихся по видам (БРС)

Учебная работа (виды)	Сумма баллов	Виды и результаты учебной работы	Оценка в аттестации	Баллы (17 недель)
Текущая учебная работа в семестре (Посещение занятий по расписанию и выполнение заданий)	**60**	Лекционные занятия (конспект) (16 занятий)	**0,25 балла** посещение 1 лекционного занятия и ведение конспекта	2 - 4
		Лабораторные работы (отчет о выполнении лабораторной работы) (16 работ).	**1,25 балла** – выполнение работы на 51-65% **2,75 балла** – выполнение работы на 85,1-100%	20 - 44
		Контрольные работы (3 работы)	**3 балла** – выполнение работы на 51-65% **4 балла** – выполнение работы на 85,1-100%	9-12
Итого по текущей работе в семестре				31 - 60
Промежуточная аттестация (экзамен)	40	Ответ на вопрос	**5 баллов** (пороговое значение) **8 баллов** (максимальное значение)	5-8
		Ответ на вопрос	**5 баллов** (пороговое значение)	5-8

Учебная работа (виды)	Сумма баллов	Виды и результаты учебной работы	Оценка в аттестации	Баллы (17 недель)
			8 баллов (максимальное значение)	
		Решение задачи 1.	**5 баллов** (пороговое значение) **12 баллов** (максимальное значение)	5 - 12
		Решение задачи 2.	**5 баллов** (пороговое значение) **12 баллов** (максимальное значение)	5 - 12
Итого по промежуточной аттестации (экзамену)				20 – 40 б.
Суммарная оценка по дисциплине: Сумма баллов текущей и промежуточной аттестации				51 – 100 б.

В промежуточной аттестации оценка выставляется в ведомость в 100-балльной шкале и в буквенном эквиваленте (таблица 5)

Таблица 5 – Соотнесение 100-балльной шкалы и буквенного эквивалента оценки

Сумма набранных баллов	Уровни освоения дисциплины и компетенций	*Экзамен*		*Зачет*
		Оценка	*Буквенный эквивалент*	*Буквенный эквивалент*
86 - 100	Продвинутый	5	отлично	Зачтено
66 - 85	Повышенный	4	хорошо	
51 - 65	Пороговый	3	удовлетворительно	
0 - 50	Первый	2	неудовлетворительно	Не зачтено

5 Материально-техническое, программное и учебно-методическое обеспечение дисциплины.

5.1 Учебная литература

Основная учебная литература

Тузовский, А. Ф. Объектно-ориентированное программирование : учебное пособие для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2023. — 213 с. — (Высшее образование). — ISBN 978-5-534-16316-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/530800.

Зыков, С. В. Программирование. Объектно-ориентированный подход : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2023. — 155 с. — (Высшее образование). — ISBN 978-5-534-00850-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/512425.

Дополнительная учебная литература

Казанский, А. А. Программирование на Visual C# : учебное пособие для вузов / А. А. Казанский. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 192 с. — (Высшее образование). — ISBN 978-5-534-12338-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/512404.

5.2 Материально-техническое и программное обеспечение дисциплины.

Учебные занятия по дисциплине проводятся в учебных аудиториях КГПИ ФГБОУ ВО «КемГУ»:

610 Учебная аудитория (мультимедийная)для проведения: - занятий лекционного типа; - текущего контроля и промежуточной аттестации. **Специализированная (учебная) мебель:** доска меловая, кафедра, столы, стулья. **Оборудование для презентации учебного материала**: стационарное - компьютер, экран, проектор. **Используемое программное обеспечение:** MSWindows (MicrosoftImaginePremium 3 year по сублицензионному договору № 1212/КМР от 12.12.2018 г. до 12.12.2021 г.), LibreOffice (свободно распространяемое ПО), FoxitReader (свободно распространяемое ПО), Firefox 14 (свободно распространяемое ПО), Яндекс.Браузер (отечественное свободно распространяемое ПО). **Интернет с обеспечением доступа в ЭИОС**.	Учебный корпус №4. 654079, Кемеровская область, г. Новокузнецк, пр-кт Металлургов, д. 19
502 Компьютерный класс. Учебная аудитория (мультимедийная) для проведения: - занятий лекционного типа; - занятий семинарского (практического) типа; - занятий лабораторного типа; - групповых и индивидуальных консультаций; - самостоятельной работы; - текущего контроля и промежуточной аттестации. **Специализированная (учебная) мебель**: доска меловая, столы компьютерные, стулья. **Оборудование для презентации учебного материала**: *стационарное* - компьютер, экран, проектор, наушники. **Лабораторное оборудование**: стационарное – компьютеры для обучающихся (16 шт.). **Используемое программное обеспечение**: MSWindows (MicrosoftImaginePremium 3 year по сублицензионному договору № 1212/КМР от 12.12.2018 г. до 12.12.2021 г.), LibreOffice (свободно распространяемое ПО), Firefox 14 (свободно распространяемое ПО), Яндекс.Браузер (отечественное свободно распространяемое ПО), MicrosoftVisualStudio (MicrosoftImaginePremium 3 year по сублицензионному договору № 1212/КМРот 12.12.2018 г. до 12.12.2021 г.), Среда статистических вычислений Rv.4.0.2 (свободно распространяемое ПО). **Интернет с обеспечением доступа в ЭИОС.**	Учебный корпус №4. 654079, Кемеровская область, г. Новокузнецк, пр-кт Металлургов, д. 19

5.3 Современные профессиональные базы данных и информационные справочные системы.

Перечень СПБД и ИСС по дисциплине

CITForum.ru - on-line библиотека свободно доступных материалов по информационным технологиям на русском языке - http://citforum.ru

Научная электронная библиотека eLIBRARY.RU – крупнейший российский информационный портал в области науки, технологии, медицины и образования, содержащий рефераты и полные тексты - www.elibrary.ru

6 Иные сведения и (или) материалы.

6.1. Примерные вопросы и задания для промежуточной аттестации

Семестр 3

Таблица 6 - Примерные теоретические вопросы и практические задания к *экзамену*

Разделы и темы	Примерные теоретические вопросы	Примерные практические задания
1. ***Объектно-ориентированное моделирование***		
1.1 Объектная модель проектирования	1. Основные принципы объектной модели: иерархия,	1. Изобразить графически отношение «зависимость» на

	контроль типов, инкапсуляция, параллелизм, абстракция, модульность, персистентность. 2. Объектно-ориентированное проектирование: объектная декомпозиция, система обозначения. 3. Объектно-ориентированный анализ. 4. Основные концепции объектного подхода. 5. Элементы объектной модели. Преимущества объектной модели.	примере объектов «пользователь» и «администратор». 2. Изобразить графически отношение «обобщение» на примере объектов «врач», «хирург» и «нейрохирург». 3. Изобразить графически отношение «реализация» на примере объектов «кассир» и «чек». 4. Изобразить графически отношение «ассоциация» на примере объектов «целый тип» и «массив данных целого типа».
1.2 Инструментальные средства проектирования объектной системы	6. Язык UML. Диаграмма в UML. 7. Типология диаграмм: структурные диаграммы, диаграммы поведения. 8. Диаграммы пакетов. 9. Диаграммы компонентов. 10. Диаграммы развертывания. 11. Диаграммы прецедентов использования. 12. Диаграммы деятельности. 13. Диаграммы классов. 14. Диаграммы последовательностей. 15. Диаграммы обзора взаимодействий. 16. Диаграммы композитных структур. 17. Диаграммы конечных автоматов. 18. Диаграммы синхронизации. 19. Диаграммы объектов. 20. Диаграммы коммуникации.	5. Построить диаграмму вариантов использования для приложения заказа такси. 6. Построить диаграмму вариантов использования банкомата. 7. Построить диаграмму вариантов использования системы online заказов. 8. Построить диаграмму классов для системы online заказов.
2. ***Объектно-ориентированное программирование***		
2.1 Классы и объекты	21. Базовые конструкции объектно-ориентированных программ: классы и объекты. 22. Инициализация и разрушение объекта. 23. Компоненты класса. 24. Конструкторы и деструкторы. 25. Перегрузка и переопределение методов класса. 26. Принцип инкапсуляции. 27. Область действия класса и доступ к компонентам класса. Управление доступом к компонентам класса. Организация внешнего	9. Построить описание класса, содержащего информацию о почтовом адресе организации. Предусмотреть возможность раздельного изменения составных частей адреса, создания и уничтожения объектов этого класса. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса. 10. Создать класс Worker, в котором будут следующие private поля - name (имя), age (возраст), salary (зарплата) и следующие public методы setName,

	доступа к локальным компонентам класса. 28. Интерфейсные (дружественные) методы. 29. Статические и константные компоненты. 30. Указатели и ссылки. 31. Операторы для динамического выделения и освобождения памяти. 32. Статические и динамические объекты. 33. Proxy-классы.	getName, setAge, getAge, setSalary, getSalary. Создать 2 объекта этого класса: 'Иван', возраст 25, зарплата 1000 и 'Вася', возраст 26, зарплата 2000. Вывести на экран сумму зарплат Ивана и Васи. Выведите на экран сумму возрастов Ивана и Васи.
2.2 Методы и механизмы наследования. Полиморфизм	34. Базовые и производные классы. Основные правила построения производных классов. 35. Конструкторы и деструкторы при наследовании. 36. Простое и множественное наследование. 37. Переопределение членов базового класса в производном. 38. Понятие раннего и позднего связывания. 39. Использование виртуального механизма для реализации принципа полиморфизма. 40. Виртуальные методы класса и механизм их использования. 41. Абстрактные классы, их назначение и свойства.	11. Изобразите графически возможное содержимое объекта «квадратное уравнение» (поля и методы). Изобразите иерархическую схему наследования. Изобразите схему взаимодействия объектов при множественном наследовании. 12. Приведите синтаксис любого класса, в котором отображалось бы свойство наследования. 13. Приведите синтаксис любого класса, в котором отображалось бы свойство полиморфизма. 14. Создать класс User, в котором будут следующие protected поля: name (имя), age (возраст), public методы setName, getName, setAge, getAge. Создать класс Worker, который наследует от класса User и вносит дополнительное private поле salary (зарплата), а также методы public getSalary и setSalary. Создать объект этого класса 'Иван', возраст 25, зарплата 1000. Создать второй объект этого класса 'Вася', возраст 26, зарплата 2000. Найти сумму зарплата Ивана и Васи. Сделать класс Student, который наследует от класса User и вносит дополнительные private поля стипендия, курс, а также геттеры и сеттеры для них.
2.3 Параметризация объектов в ООП. Использование параметризированных классов	42. Введение в параметризированные классы. 43. Параметризированные классы и методы, их свойства. 44. Совместное использование параметризации и принципов наследования. 45. Организация внешнего доступа к компонентам параметризированных классов. 46. Параметризированные классы и статические члены.	15. Реализовать параметризованный класс «Матрица», типы элементов которого могут быть комплексными числами. Возможность класса: Разность двух матриц и присвоение результата третьей переменной-матрице осуществляется одной строкой C=A-B. 16. Реализовать параметризованный класс «Матрица», типы элементов которого могут быть комплексными числами. Возможность класса: Деление матрицы на вещественное число и присвоение результата третьей переменной-матрице

		осуществляется одной строкой C=A/b, где b – вещественное число
2.4 Обработка исключений	47. Основы обработки исключений. 48. Генерация исключений. 49. Перехватывание исключений. 50. Повторная генерация исключения. 51. Обработка неожидаемых исключений. 52. Генерация исключений в конструкторах. 53. Исключения и наследование. 54. Спецификация исключений. 55. Иерархия исключений стандартной библиотеки.	17. Написать программу перевода числа из восьмеричной системы счисления в шестеричную. Для проверки корректности работы использовать средство CException. 18. Написать класс MyMath со следующими статическими методами. - int pow(int x, int n) -возведение числа x в степень n. Выбросить исключение IllegalArgumentException, если на вход подаются отрицательный x или отрицательный n (различить случаи для x и n с помощью текста сообщения об ошибке). Выбросить исключение ArithmeticException, если x и n одновременно равны 0. - int sum(int [] arr) - подсчет суммы целочисленного массива. Выбросить исключение ArithmenticException, если длина массива равна 0 (не путайте с массивом, равным null). В другом классе вызвать описанные методы, поместив их в блок try-catch (для каждого метода - отдельный блок). Для каждого метода обработать возникающие в нем исключения.
2.5 Потоки ввода/вывода, организация работы с файлами	56. Потоки, общее понятие. Организация ввода из потока и вывод в поток. Контроль состояния потока и исправление ошибок. 57. Неформатированный ввод-вывод. 58. Манипуляторы потоков (стандартные и определяемые пользователем). 59. Файлы и потоки их взаимосвязь. 60. Файлы последовательного и произвольного доступа. 61. Организация ввода и вывода объектов.	19. Для хранения данных о ноутбуках описать структуру NOTEBOOK вида: наименование, габариты (длина, ширина, высота), вес, цена. Написать функцию, которая читает данные о ноутбуках из файла note.txt (предварительно создать и заполнить файл) в структуру приведенного вида. Написать функцию, записывающую данные из структуры в конец бинарного файла. Структура бинарного файла: первые 2 байта (целое) – число записей в файле; далее записи в формате NOTEBOOK. Написать программу, записывающую в файл данные лишь о тех ноутбуках, диагональ дисплея которых больше 11 дюймов.
2.6 Контейнерные типы	62. Введение в стандартную библиотеку шаблонов (классов коллекций), основные понятия. 63. Классы контейнеры и итераторы. Типы контейнерных классов, адаптеры контейнеров. Алгоритмы и их использование с контейнерными классами.	20. В файле находится произвольное количество целых чисел. Написать программу, которая будет считывать их в вектор и выводит на экран в том же порядке. 21. Продемонстрировать 3 способа определить объект для последовательного контейнера.
3. ***Паттерны проектирования***		

3.1 Классификация паттернов	64. Понятие «Паттерн проектирования». Принципы описания паттернов проектирования. 65. Классификация паттернов проектирования: порождающие паттерны, структурные паттерны, паттерны поведения. 66. Основы использования паттернов. Влияние паттернов на функционирование программных систем.	
3.2 Порождающие паттерны	67. Назначение и особенности порождающих паттернов. 68. Паттерн «Одиночка»: назначение, случаи применения, особенности реализации паттерна, достоинства. 69. Паттерн «Абстрактная фабрика»: назначение, случаи применения, особенности реализации паттерна, достоинства. 70. Паттерн «Строитель»: назначение, случаи применения, алгоритм реализации паттерна, достоинства. 71. Паттерн «Фабричный метод»: назначение, случаи применения, алгоритм реализации паттерна, классический вариант фабричного метода, достоинства, две основные разновидности паттерна. 72. Паттерн «Прототип»: назначение, случаи применения, алгоритм реализации паттерна, основные участники механизма на основе паттерна «Прототип», достоинства, использование диспетчера прототипов.	22. Реализовать паттерн одиночка. 23. Реализовать паттерн абстрактная фабрика. 24. Реализовать паттерн строитель. 25. Реализовать паттерн фабричный метод. 26. Реализовать паттерн прототип.
3.3 Структурные паттерны	73. Назначение структурных паттернов. 74. Паттерн «Адаптер»: назначение, случаи применения, результаты реализации адаптера класса, результаты реализации адаптера объектов, алгоритм реализации паттерна, достоинства. 75. Паттерн «Мост»: назначение, случаи	27. Реализовать паттерн адаптер классов. 28. Реализовать паттерн адаптер объектов. 30. Реализовать паттерн компоновщик. 31. Реализовать паттерн декоратор.

	применения, особенности, алгоритм реализации паттерна, результаты применения, достоинства. 76. Паттерн «Компоновщик»: назначение, случаи применения, особенности реализации, алгоритм реализации паттерна, результаты применения, достоинства. 77. Паттерн «Декоратор»: назначение, случаи применения, особенности реализации, алгоритм реализации паттерна, результаты применения, достоинства.	
3.4 Паттерны поведения	78. Назначение паттернов поведения. Применение поведенческих паттернов. 79. Паттерн «Цепочка обязанностей»: назначение, случаи применения, особенности реализации, алгоритм реализации паттерна, результаты применения, достоинства. 80. Паттерн «Команда»: назначение, случаи применения, особенности реализации, алгоритм реализации паттерна, результаты применения, достоинства. 81. Паттерн «Наблюдатель»: назначение, случаи применения, особенности реализации, алгоритм реализации паттерна, результаты применения, достоинства.	32. Реализовать паттерн цепочка обязанностей. 33. Реализовать паттерн команда. 34. Реализовать паттерн наблюдатель.
Компетенции		
ПК-2 Способен разрабатывать спецификации требований, проектировать и реализовывать программное обеспечение	**Задание 1.** Определить класс vector2 как вектор на плоскости с данными x и y. Определить для него операции сложения, вычитания и скалярного произведения, определить функцию присваивания значения координатам вектора и функцию вывода значений на консоль. Определить класс vector3 как вектор в пространстве, породив его от класса vector2. Переопределить для него функции и операции. Требуется написать код, определяющий указанные классы. **Задание 2.** Предметная область: кинопрокат. Пользователь может выбрать определённую киноленту, при заказе киноленты указывается язык звуковой дорожки, который совпадает с языком файла субтитров. Система должна поставлять фильм с требуемыми характеристиками, причём при смене языка звуковой дорожки должен меняться и язык файла субтитров и наоборот. 1. Построить UML-диаграмму классов.	

	2. Разработать систему Кинопрокат на основе паттерна Абстрактная фабрика.

Составитель (и): старший преподаватель кафедры МФММ Гаврилова Ю.С.
(*фамилия, инициалы и должность преподавателя (ей)*)