

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кемеровский государственный университет»
Кузбасский гуманитарно-педагогический институт
федерального государственного бюджетного образовательного учреждения
высшего образования
«Кемеровский государственный университет»
Факультет информатики, математики и экономики

УТВЕРЖДАЮ
Декан
А. В. Фомина
9 февраля 2023 г.

Рабочая программа дисциплины

К.М.08.01 Объектно-ориентированное проектирование и программирование

Код, название дисциплины

Направление подготовки

02.03.03 Математическое обеспечение и администрирование

информационных систем

Код, название направления

Направленность (профиль) подготовки

Программное и математическое обеспечение информационных технологий

Программа бакалавриата

Квалификация выпускника

бакалавр

Форма обучения

Очная

Год набора 2022

Новокузнецк 2023

Оглавление

1 Цель дисциплины	3
1.1 Формируемые компетенции	3
1.2 Индикаторы достижения компетенций.....	3
1.3 Знания, умения, навыки (ЗУВ) по дисциплине	3
2 Объём и трудоёмкость дисциплины по видам учебных занятий. Формы промежуточной аттестации.	4
3. Учебно-тематический план и содержание дисциплины.....	5
3.1 Учебно-тематический план	5
3.2. Содержание занятий по видам учебной работы.....	5
4 Порядок оценивания успеваемости и сформированности компетенций обучающегося в текущей и промежуточной аттестации.....	8
5 Материально-техническое, программное и учебно-методическое обеспечение дисциплины.	9
5.1 Учебная литература	9
5.2 Материально-техническое и программное обеспечение дисциплины.....	9
5.3 Современные профессиональные базы данных и информационные справочные системы.	11
6 Иные сведения и (или) материалы.....	11
6.1.Примерные темы письменных учебных работ	11
6.2. Примерные вопросы и задания для промежуточной аттестации	11

1 Цель дисциплины.

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

ПК-3.

Содержание компетенций как планируемых результатов обучения по дисциплине см. таблицы 1 и 2.

1.1 Формируемые компетенции

Таблица 1 - Формируемые дисциплиной компетенции

Наименование вида компетенции (универсальная, общепрофессиональная, профессиональная)	Наименование категории (группы) компетенций	Код и название компетенции
профессиональная		ПК-3 Способен проектировать и конструировать программные средства, а также архитектуры программных средств

1.2 Индикаторы достижения компетенций

Таблица 2 – Индикаторы достижения компетенций, формируемые дисциплиной

Код и название компетенции	Индикаторы достижения компетенции по ОПОП	Дисциплины и практики, формирующие компетенцию ОПОП
ПК-3 Способен проектировать и конструировать программные средства, а также архитектуры программных средств	3.1 Проводит анализ требований к программным средствам 3.2 Проектирует программные средства 3.3 Конструирует программные средства	К.М.08.01 Объектно-ориентированное проектирование и программирование К.М.08.04 Математическое и программное обеспечение проектной деятельности К.М.08.06 Вычислительные системы и сети К.М.08.ДВ.01.01 Разработка программного обеспечения для математического моделирования К.М.08.ДВ.01.02 Разработка приложений для имитационного моделирования К.М.08.ДВ.02.01 Разработка трансляторов для языков программирования К.М.08.ДВ.02.02 Программирование на Java К.М.09.03(Пд) Преддипломная практика К.М.10.01(Д) Выполнение и защита выпускной квалификационной работы

1.3 Знания, умения, навыки (ЗУВ) по дисциплине

Таблица 3 – Знания, умения, навыки, формируемые дисциплиной

Код и название компетенции	Индикаторы достижения компетенции, закрепленные за дисциплиной	Знания, умения, навыки (ЗУВ), формируемые дисциплиной
ПК-3 Способен проектировать и конструировать программные	3.2 Проектирует программные средства	Знать: - теоретические аспекты

Код и название компетенции	Индикаторы достижения компетенции, закрепленные за дисциплиной	Знания, умения, навыки (ЗУВ), формируемые дисциплиной
средства, а также архитектуры программных средств	3.3 Конструирует программные средства	проектирования и конструирования программных средств с использованием объектного подхода. Уметь: - применять существующие паттерны проектирования для проектирования и конструирования программных средств. Владеть: - навыками проектирования и конструирования программных средств с использованием объектного подхода.

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

Таблица 4 – 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	ОЗФО	ЗФО
1 Общая трудоемкость дисциплины	144		
2 Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)	66		
Аудиторная работа (всего):	66		
в том числе:			
лекции	32		
практические занятия, семинары			
практикумы			
лабораторные работы	34		
в интерактивной форме			
в электронной форме			
Внеаудиторная работа (всего):			
в том числе, индивидуальная работа обучающихся с преподавателем			
подготовка курсовой работы /контактная работа			
групповая, индивидуальная консультация и иные виды учебной деятельности, предусматривающие групповую или индивидуальную работу обучающихся с преподавателем)			
творческая работа (эссе)			
3 Самостоятельная работа обучающихся (всего)	78		
4 Промежуточная аттестация обучающегося	Зачет с оценкой – 4 семестр		

3. Учебно-тематический план и содержание дисциплины.

3.1 Учебно-тематический план

Таблица 5 - Учебно-тематический план очной формы обучения

№ недели п/п	Разделы и темы дисциплины по занятиям	Общая трудоём- кость (всего час.)	Трудоемкость занятий (час.)						Формы текущего контроля и промежуточно- й аттестации успеваемости
			ОФО			ЗФО			
			Аудиторн. занятия		СРС	Аудиторн. занятия		СРС	
			лекц.	лаб.		лекц.	лаб.		
Семестр 4									
	1. Объектно-ориентированное моделирование	14	6	4	4				Контрольная работа №1
1	1.1 Объектная модель проектирования	4	2		2				Защита отчета по ЛР № 1
2	1.2 Инструментальные средства проектирования объектной системы	10	4	4	2				Защита отчета по ЛР № 2
	2. Объектно-ориентированное программирование	66	12	20	34				Контрольная работа №2
3	2.1 Классы и объекты	8	2	2	4				Защита отчета по ЛР № 3
4	2.2 Методы и механизмы наследования. Полиморфизм	10	2	2	6				Защита отчетов по ЛР № 4-5
5	2.3 Параметризация объектов в ООП. Использование параметризованных классов	12	2	2	8				Защита отчета по ЛР № 6
6	2.4 Обработка исключений	12	2	4	8				Защита отчета по ЛР № 7
7	2.5 Потоки ввода/вывода, организация работы с файлами	12	2	4	6				Защита отчета по ЛР № 8
8	2.6 Контейнерные типы	12	2	2	8				Защита отчета по ЛР № 9
	3. Паттерны проектирования	64	14	14	36				Контрольная работа №3
9	3.1 Классификация паттернов	6	2		4				
10	3.2 Порождающие паттерны	16	4	4	8				Защита отчетов по ЛР № 10-11
11	3.3 Структурные паттерны	20	4	6	10				Защита отчетов по ЛР № 12-13
12	3.4 Паттерны поведения	22	4	6	12				Защита отчетов по ЛР № 14-16
	Промежуточная аттестация – зачет с оценкой								Зачет с оценкой
ИТОГО по семестру 4		144	32	34	78				

3.2. Содержание занятий по видам учебной работы

Таблица 6 – Содержание дисциплины

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
	Семестр 4	
<i>Содержание лекционного курса</i>		
1	Объектно-ориентированное моделирование	

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
1.1	Объектная модель проектирования	Основные принципы объектной модели: иерархия, контроль типов, инкапсуляция, параллелизм, абстракция, модульность, персистентность. Объектно-ориентированное проектирование: объектная декомпозиция, система обозначения. Объектно-ориентированный анализ. Основные концепции объектного подхода. Элементы объектной модели. Преимущества объектной модели.
1.2	Инструментальные средства проектирования объектной системы	Язык UML. Диаграмма в UML. Типология диаграмм: структурные диаграммы, диаграммы поведения. Диаграммы пакетов, компонентов, развертывания, прецедентов использования, деятельности, классов, последовательностей, обзора взаимодействий, композитных структур, конечных автоматов, синхронизации, объектов, коммуникации.
2	Объектно-ориентированное программирование	
2.1	Классы и объекты	Базовые конструкции объектно-ориентированных программ: классы и объекты. Инициализация и разрушение объекта. Компоненты класса. Конструкторы и деструкторы. Перегрузка и переопределение методов класса. Принцип инкапсуляции. Область действия класса и доступ к компонентам класса. Управление доступом к компонентам класса. Организация внешнего доступа к локальным компонентам класса. Интерфейсные (дружественные) методы. Статические и константные компоненты. Указатели и ссылки. Операторы для динамического выделения и освобождения памяти. Статические и динамические объекты. Проху-классы.
2.2	Методы и механизмы наследования. Полиморфизм	Базовые и производные классы. Основные правила построения производных классов. Конструкторы и деструкторы при наследовании. Композиция и наследование. Простое и множественное наследование. Переопределение членов базового класса в производном. Понятие раннего и позднего связывания. Использование виртуального механизма для реализации принципа полиморфизма. Виртуальные методы класса и механизм их использования. Абстрактные классы, их назначение и свойства.
2.3	Параметризация объектов в ООП. Использование параметризованных классов	Введение в параметризованные классы. Параметризованные классы и методы, их свойства. Совместное использование параметризации и принципов наследования. Организация внешнего доступа к компонентам параметризованных классов. Параметризованные классы и статические члены.
2.4	Обработка исключений	Основы обработки исключений. Генерация исключений. Перехватывание исключений. Повторная генерация исключения. Обработка неожиданных исключений. Генерация исключений в конструкторах. Исключения и наследование. Спецификация исключений. Иерархия исключений стандартной библиотеки.
2.5	Потоки ввода/вывода, организация работы с файлами	Потоки, общее понятие. Организация ввода из потока и вывод в поток. Контроль состояния потока и исправление ошибок.

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
		Неформатированный ввод-вывод. Манипуляторы потоков (стандартные и определяемые пользователем). Файлы и потоки их взаимосвязь. Файлы последовательного и произвольного доступа. Организация ввода и вывода объектов.
2.6	Контейнерные типы	Введение в стандартную библиотеку шаблонов (классов коллекций), основные понятия. Классы контейнеры и итераторы. Типы контейнерных классов, адаптеры контейнеров. Алгоритмы и их использование с контейнерными классами.
3	Паттерны проектирования	
3.1	Классификация паттернов	Понятие «Паттерн проектирования». Принципы описания паттернов проектирования. Классификация паттернов проектирования: порождающие паттерны, структурные паттерны, паттерны поведения. Основы использования паттернов. Влияние паттернов на функционирование программных систем.
3.2	Порождающие паттерны	Назначение и особенности порождающих паттернов. Паттерн «Одиночка»: назначение, случаи применения, особенности реализации паттерна, достоинства. Паттерн «Абстрактная фабрика»: назначение, случаи применения, особенности реализации паттерна, достоинства. Паттерн «Строитель»: назначение, случаи применения, алгоритм реализации паттерна, достоинства. Паттерн «Фабричный метод»: назначение, случаи применения, алгоритм реализации паттерна, классический вариант фабричного метода, достоинства, две основные разновидности паттерна. Паттерн «Прототип»: назначение, случаи применения, алгоритм реализации паттерна, основные участники механизма на основе паттерна «Прототип», достоинства, использование диспетчера прототипов.
3.3	Структурные паттерны	Назначение структурных паттернов. Паттерн «Адаптер»: назначение, случаи применения, результаты реализации адаптера класса, результаты реализации адаптера объектов, алгоритм реализации паттерна, достоинства. Паттерн «Мост»: назначение, случаи применения, особенности, алгоритм реализации паттерна, результаты применения, достоинства. Паттерн «Компоновщик»: назначение, случаи применения, особенности реализации, алгоритм реализации паттерна, результаты применения, достоинства. Паттерн «Декоратор»: назначение, случаи применения, особенности реализации, алгоритм реализации паттерна, результаты применения, достоинства.
3.4	Паттерны поведения	Назначение паттернов поведения. Применение поведенческих паттернов. Паттерн «Цепочка обязанностей»: назначение, случаи

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
		применения, особенности реализации, алгоритм реализации паттерна, результаты применения, достоинства. Паттерн «Команда»: назначение, случаи применения, особенности реализации, алгоритм реализации паттерна, результаты применения, достоинства. Паттерн «Наблюдатель»: назначение, случаи применения, особенности реализации, алгоритм реализации паттерна, результаты применения, достоинства.
<i>Содержание лабораторных занятий</i>		
1	Объектно-ориентированное моделирование	
1.2	Инструментальные средства проектирования объектной системы	Лабораторная работа №1 «Построение диаграммы вариантов использования». Лабораторная работа №2 «Построение диаграммы классов».
2	Объектно-ориентированное программирование	
2.1	Классы и объекты	Лабораторная работа №3 «Создание классов».
2.2	Методы и механизмы наследования. Полиморфизм	Лабораторная работа №4 «Наследование классов». Лабораторная работа №5 «Полиморфизм».
2.3	Параметризация объектов в ООП. Использование параметризованных классов	Лабораторная работа №6 «Параметризованные классы».
2.4	Обработка исключений	Лабораторная работа №7 «Обработка исключений».
2.5	Потоки ввода/вывода, организация работы с файлами	Лабораторная работа №8 «Работа с файлами».
2.6	Контейнерные типы	Лабораторная работа №9 «Конвейерные типы».
3	Паттерны проектирования	
3.2	Порождающие паттерны	Лабораторная работа № 10. Программная реализация паттерна проектирования «Одиночка». Лабораторная работа № 11. Программная реализация паттерна проектирования «Прототип».
3.3	Структурные паттерны	Лабораторная работа № 12. Программная реализация паттерна проектирования «Компоновщик». Лабораторная работа № 13. Программная реализация паттерна проектирования «Декоратор».
3.4	Паттерны поведения	Лабораторная работа № 14. Программная реализация паттерна проектирования «Цепочка обязанностей». Лабораторная работа № 15. Программная реализация паттерна проектирования «Команда». Лабораторная работа № 16. Программная реализация паттерна проектирования «Наблюдатель».
	Промежуточная аттестация - зачет	

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 7.

Таблица 7 - Шкала и показатели оценивания результатов учебной работы обучающихся по видам в балльно-рейтинговой системе (БРС)

Учебная работа (виды)	Сумма баллов	Виды и результаты учебной работы	Оценка в аттестации (шкала и показатели оценивания)	Баллы (17 недель)
Текущая учебная работа в семестре (Посещение занятий по расписанию и выполнение заданий)	80	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).	2,5 балла – выполнение работы на 51-65% 4 балла – выполнение работы на 85,1-100%	40 - 64
		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		51 - 80
Промежуточная аттестация (зачет с оценкой)	20	Ответ на вопрос	2 балла (пороговое значение) 5 баллов (максимальное значение)	2-5
		Ответ на вопрос	2 балла (пороговое значение) 5 баллов (максимальное значение)	2-5
		Решение задачи 1.	3 балла (пороговое значение) 5 баллов (максимальное значение)	3 - 5
		Решение задачи 2.	3 балла (пороговое значение) 5 баллов (максимальное значение)	3 - 5
Итого по промежуточной аттестации (зачету с оценкой)				10 – 20 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 8)

Таблица 8 – 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> (дата обращения:

03.02.2023).

Дополнительная учебная литература

Казанский, А. А. Объектно-ориентированный анализ и программирование на Visual Basic 2013 : учебник для вузов / А. А. Казанский. — Москва : Издательство Юрайт, 2023. — 290 с. — (Высшее образование). — ISBN 978-5-534-01122-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/512345> (дата обращения: 03.02.2023).

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> (дата обращения: 03.02.2023).

5.2 Материально-техническое и программное обеспечение дисциплины.

Учебные занятия по дисциплине проводятся в учебных аудиториях КГПИ КемГУ:

615 Учебная аудитория (мультимедийная) для проведения: - занятий лекционного типа. Специализированная (учебная) мебель: доска меловая, кафедра, столы, стулья. Оборудование для презентации учебного материала: стационарное - компьютер, экран, проектор, акустическая система (колонки). Используемое программное обеспечение: Ubuntu Linux(свободно распространяемое ПО), LibreOffice (свободно распространяемое ПО), Яндекс.Браузер (отечественное свободно распространяемое ПО). Интернет с обеспечением доступа в ЭИОС.	654079, Кемеровская область, г. Новокузнецк, пр-кт Металлургов, д. 19
509 Компьютерный класс. Учебная аудитория (мультимедийная) для проведения: - занятий лабораторного типа; - групповых и индивидуальных консультаций; - самостоятельной работы; - текущего контроля и промежуточной аттестации. Специализированная (учебная) мебель: доска меловая, кафедра, столы, стулья, Оборудование для презентации учебного материала: стационарное - компьютер преподавателя, экран, проектор. Оборудование: стационарное- компьютеры для обучающихся (18 шт.), наушники. Используемое программное обеспечение: MS Windows (Microsoft Imagine Premium 3 year по	654079, Кемеровская область, г. Новокузнецк, пр-кт Металлургов, д. 19

<p>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), Microsoft Visual Studio (Microsoft Imagine Premium 3 year по сублицензионному договору № 1212/КМР от 12.12.2018 г. до 12.12.2021 г.). Интернет с обеспечением доступа в ЭИОС.</p>	
---	--

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

Единое окно доступа к образовательным ресурсам - <http://window.edu.ru/>

6 Иные сведения и (или) материалы.

6.1. Примерные темы письменных учебных работ

6.2. Примерные вопросы и задания для промежуточной аттестации

Семестр 4

Таблица 9 - Примерные теоретические вопросы и практические задания к зачету с оценкой

Разделы и темы	Примерные теоретические вопросы	Примерные практические задания
1. Объектно-ориентированное моделирование		
1.1 Объектная модель проектирования	<p>1. Основные принципы объектной модели: иерархия, контроль типов, инкапсуляция, параллелизм, абстракция, модульность, персистентность.</p> <p>2. Объектно-ориентированное проектирование: объектная декомпозиция, система обозначения.</p> <p>3. Объектно-ориентированный анализ.</p> <p>4. Основные концепции объектного подхода.</p> <p>5. Элементы объектной модели. Преимущества объектной модели.</p>	<p>1. Изобразить графически отношение «зависимость» на примере объектов «пользователь» и «администратор».</p> <p>2. Изобразить графически отношение «обобщение» на примере объектов «врач», «хирург» и «нейрохирург».</p> <p>3. Изобразить графически отношение «реализация» на примере объектов «кассир» и «чек».</p> <p>4. Изобразить графически отношение «ассоциация» на примере объектов «целый тип» и «массив данных целого типа».</p>
1.2 Инструментальные средства проектирования	6. Язык UML. Диаграмма в	5. Построить диаграмму вариантов

объектной системы	<p>UML.</p> <p>7. Типология диаграмм: структурные диаграммы, диаграммы поведения.</p> <p>8. Диаграммы пакетов.</p> <p>9. Диаграммы компонентов.</p> <p>10. Диаграммы развертывания.</p> <p>11. Диаграммы прецедентов использования.</p> <p>12. Диаграммы деятельности.</p> <p>13. Диаграммы классов.</p> <p>14. Диаграммы последовательностей.</p> <p>15. Диаграммы обзора взаимодействий.</p> <p>16. Диаграммы композитных структур.</p> <p>17. Диаграммы конечных автоматов.</p> <p>18. Диаграммы синхронизации.</p> <p>19. Диаграммы объектов.</p> <p>20. Диаграммы коммуникации.</p>	<p>использования для приложения заказа такси.</p> <p>6. Построить диаграмму вариантов использования банкомата.</p> <p>7. Построить диаграмму вариантов использования системы online заказов.</p> <p>8. Построить диаграмму классов для системы online заказов.</p>
2. Объектно-ориентированное программирование		
2.1 Классы и объекты	<p>21. Базовые конструкции объектно-ориентированных программ: классы и объекты.</p> <p>22. Инициализация и разрушение объекта.</p> <p>23. Компоненты класса.</p> <p>24. Конструкторы и деструкторы.</p> <p>25. Перегрузка и переопределение методов класса.</p> <p>26. Принцип инкапсуляции.</p> <p>27. Область действия класса и доступ к компонентам класса. Управление доступом к компонентам класса. Организация внешнего доступа к локальным компонентам класса.</p> <p>28. Интерфейсные (дружественные) методы.</p> <p>29. Статические и константные компоненты.</p> <p>30. Указатели и ссылки.</p> <p>31. Операторы для динамического выделения и освобождения памяти.</p> <p>32. Статические и динамические объекты.</p> <p>33. Proxu-классы.</p>	<p>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.</p> <p>10. Создать класс Worker, в котором будут следующие private поля - name (имя), age (возраст), salary (зарплата) и следующие public методы setName, 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.</p>
2.2 Методы и механизмы	34. Базовые и производные	11. Изобразите графически

наследования. Полиморфизм	<p>классы. Основные правила построения производных классов.</p> <p>35. Конструкторы и деструкторы при наследовании.</p> <p>36. Простое и множественное наследование.</p> <p>37. Переопределение членов базового класса в производном.</p> <p>38. Понятие раннего и позднего связывания.</p> <p>39. Использование виртуального механизма для реализации принципа полиморфизма.</p> <p>40. Виртуальные методы класса и механизм их использования.</p> <p>41. Абстрактные классы, их назначение и свойства.</p>	<p>возможное содержимое объекта «квадратное уравнение» (поля и методы). Изобразите иерархическую схему наследования. Изобразите схему взаимодействия объектов при множественном наследовании.</p> <p>12. Приведите синтаксис любого класса, в котором отображалось бы свойство наследования.</p> <p>13. Приведите синтаксис любого класса, в котором отображалось бы свойство полиморфизма.</p> <p>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.</p>
2.3 Параметризация объектов в ООП. Использование параметризованных классов	<p>42. Введение в параметризованные классы.</p> <p>43. Параметризованные классы и методы, их свойства.</p> <p>44. Совместное использование параметризации и принципов наследования.</p> <p>45. Организация внешнего доступа к компонентам параметризованных классов.</p> <p>46. Параметризованные классы и статические члены.</p>	<p>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 $C=A-B$.</p> <p>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</p>
2.4 Обработка исключений	<p>47. Основы обработки исключений.</p> <p>48. Генерация исключений.</p> <p>49. Перехватывание исключений.</p> <p>50. Повторная генерация исключения.</p> <p>51. Обработка неожиданных исключений.</p> <p>52. Генерация исключений в</p>	<p>17. Написать программу перевода числа из восьмеричной системы счисления в шестеричную. Для проверки корректности работы использовать средство CException.</p> <p>18. Написать класс MyMath со следующими статическими методами.</p> <ul style="list-style-type: none"> - int pow(int x, int n) -возведение числа x в степень n. Выбросить исключение IllegalArgumentException, если на

	<p>конструкторах.</p> <p>53. Исключения и наследование.</p> <p>54. Спецификация исключений.</p> <p>55. Иерархия исключений стандартной библиотеки.</p>	<p>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 ArithmeticException, 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.</p>
2.5 Потоки ввода/вывода, организация работы с файлами	<p>56. Потоки, общее понятие. Организация ввода из потока и вывод в поток. Контроль состояния потока и исправление ошибок.</p> <p>57. Неформатированный ввод-вывод.</p> <p>58. Манипуляторы потоков (стандартные и определяемые пользователем).</p> <p>59. Файлы и потоки их взаимосвязь.</p> <p>60. Файлы последовательного и произвольного доступа.</p> <p>61. Организация ввода и вывода объектов.</p>	<p>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.</p>
2.6 Контейнерные типы	<p>62. Введение в стандартную библиотеку шаблонов (классов коллекций), основные понятия.</p> <p>63. Классы контейнеры и итераторы. Типы контейнерных классов, адаптеры контейнеров. Алгоритмы и их использование с контейнерными классами.</p>	<p>20. В файле находится произвольное количество целых чисел. Написать программу, которая будет считывать их в вектор и выводит на экран в том же порядке.</p> <p>21. Продемонстрировать 3 способа определить объект для последовательного контейнера.</p>
3. Паттерны проектирования		
3.1 Классификация паттернов	<p>64. Понятие «Паттерн проектирования». Принципы описания паттернов проектирования.</p> <p>65. Классификация паттернов проектирования: порождающие паттерны, структурные паттерны, паттерны поведения.</p> <p>66. Основы использования</p>	

	паттернов. Влияние паттернов на функционирование программных систем.	
3.2 Порождающие паттерны	<p>67. Назначение и особенности порождающих паттернов.</p> <p>68. Паттерн «Одиночка»: назначение, случаи применения, особенности реализации паттерна, достоинства.</p> <p>69. Паттерн «Абстрактная фабрика»: назначение, случаи применения, особенности реализации паттерна, достоинства.</p> <p>70. Паттерн «Строитель»: назначение, случаи применения, алгоритм реализации паттерна, достоинства.</p> <p>71. Паттерн «Фабричный метод»: назначение, случаи применения, алгоритм реализации паттерна, классический вариант фабричного метода, достоинства, две основные разновидности паттерна.</p> <p>72. Паттерн «Прототип»: назначение, случаи применения, алгоритм реализации паттерна, основные участники механизма на основе паттерна «Прототип», достоинства, использование диспетчера прототипов.</p>	<p>22. Реализовать паттерн одиночка.</p> <p>23. Реализовать паттерн абстрактная фабрика.</p> <p>24. Реализовать паттерн строитель.</p> <p>25. Реализовать паттерн фабричный метод.</p> <p>26. Реализовать паттерн прототип.</p>
3.3 Структурные паттерны	<p>73. Назначение структурных паттернов.</p> <p>74. Паттерн «Адаптер»: назначение, случаи применения, результаты реализации адаптера класса, результаты реализации адаптера объектов, алгоритм реализации паттерна, достоинства.</p> <p>75. Паттерн «Мост»: назначение, случаи применения, особенности, алгоритм реализации паттерна, результаты применения, достоинства.</p> <p>76. Паттерн «Компоновщик»: назначение, случаи применения, особенности реализации, алгоритм</p>	<p>27. Реализовать паттерн адаптер классов.</p> <p>28. Реализовать паттерн адаптер объектов.</p> <p>30. Реализовать паттерн компоновщик.</p> <p>31. Реализовать паттерн декоратор.</p>

	<p>реализации паттерна, результаты применения, достоинства.</p> <p>77. Паттерн «Декоратор»: назначение, случаи применения, особенности реализации, алгоритм реализации паттерна, результаты применения, достоинства.</p>	
3.4 Паттерны поведения	<p>78. Назначение паттернов поведения. Применение поведенческих паттернов.</p> <p>79. Паттерн «Цепочка обязанностей»: назначение, случаи применения, особенности реализации, алгоритм реализации паттерна, результаты применения, достоинства.</p> <p>80. Паттерн «Команда»: назначение, случаи применения, особенности реализации, алгоритм реализации паттерна, результаты применения, достоинства.</p> <p>81. Паттерн «Наблюдатель»: назначение, случаи применения, особенности реализации, алгоритм реализации паттерна, результаты применения, достоинства.</p>	<p>32. Реализовать паттерн цепочка обязанностей.</p> <p>33. Реализовать паттерн команда.</p> <p>34. Реализовать паттерн наблюдатель.</p>

Составитель (и): Гаврилова Ю. С., старший преподаватель кафедры математики, физики и математического моделирования

(фамилия, инициалы и должность преподавателя (ей))