

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ЛПУ»)**

Структурное подразделение Институт физико-математического образова-
ния, информационных и обслуживающих технологий
Кафедра информационных образовательных технологий и систем

УТВЕРЖДАЮ

Директор ИФМОИОТ

Е.Е. Горбенко

2023 г.



Приложение к рабочей программе учебной дисциплины

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
для проведения текущего контроля и промежуточной аттестации
обучающихся по дисциплине
«Основы объектно-ориентированного программирования»

По направлению подготовки 09.03.04 Программная инженерия

Профиль подготовки Программное обеспечение систем и комплексов

Квалификация выпускника – бакалавр

Форма обучения очная, заочная

Курс ОФО – 2 курс, ЗФО – 2 курс

Разработчик

Швыров В.В.

доцент кафедры информационных
образовательных технологий и
систем

Заведующий кафедрой

Д.А. Капустин

Протокол от «24» ноября 2023 г. №8

Луганск, 2023

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

1.1. Область применения

Фонд оценочных средств (ФОС) – неотъемлемая часть рабочей программы дисциплины «Объектно-ориентированное программирование» и предназначен для контроля и оценки образовательных достижений студентов, освоивших программу дисциплины.

1.2. Цели и задачи фонда оценочных средств

Цель ФОС – установить соответствие уровня подготовки обучающегося требованиям ФГОС ВО бакалавриат по направлению подготовки 09.03.04 Программная инженерия, утвержденным приказом Министерства образования и науки Российской Федерации от 19 сентября 2017 г. № 920 и Профессиональным стандартом, утвержденным Приказом Министерства труда и социальной защиты Российской Федерации «Об утверждении профессионального стандарта 06.001 «Программист» от 20.07.2022 № 424н.

1.3. Перечень компетенций, формируемых в процессе освоения основной образовательной программы

Процесс освоения дисциплины направлен на формирование следующих компетенций и индикаторов их достижения:

Код по ФГОС ВО	Индикатор достижения
Общепрофессиональные	
ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.1. Знает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности ОПК-2.2. Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности ОПК-2.3. Владеет навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности
ОПК-5. Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.1. Знает основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем ОПК-5.2. Умеет выполнять параметрическую настройку информационных и автоматизированных систем ОПК-5.3. Владеет навыками установки программного и аппаратного обеспечения информационных и автоматизированных систем
ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.1. Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий

	<p>ОПК-6.2. Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ</p> <p>ОПК-6.3. Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач</p>
--	--

1.4. Этапы формирования компетенций и средства оценивания уровня их сформированности

Этапы формирования компетенций	Компетенции	Контрольно-оценочные средства / способ оценивания
Тема 1. ООП принципы. Классы.	ОПК-2 ОПК-5 ОПК-6	Устный опрос. Выполнение лабораторных работ
Тема 2. Конструкторы и деструкторы.	ОПК-2 ОПК-5 ОПК-6	Устный опрос. Выполнение лабораторных работ
Тема 3. Иерархия классов. Наследование.	ОПК-2 ОПК-5 ОПК-6	Устный опрос. Выполнение лабораторных работ
Тема 4. Дружественные классы и функции.	ОПК-2 ОПК-5 ОПК-6	Устный опрос. Выполнение лабораторных работ
Тема 5. Виртуальные классы и функции.	ОПК-2 ОПК-5 ОПК-6	Устный опрос. Выполнение лабораторных работ
Тема 6. Абстрактные классы.	ОПК-2 ОПК-5 ОПК-6	Устный опрос. Выполнение лабораторных работ
Тема 7. Перегрузка операторов.	ОПК-2 ОПК-5 ОПК-6	Устный опрос. Выполнение лабораторных работ
Тема 8. Шаблоны классов.	ОПК-2 ОПК-5 ОПК-6	Устный опрос. Выполнение лабораторных работ
Тема 9. Диаграммы классов.	ОПК-2 ОПК-5 ОПК-6	Устный опрос. Выполнение лабораторных работ
Текущая аттестация	ОПК-2 ОПК-5 ОПК-6	Индивидуальное задание
Промежуточная аттестация	ОПК-2 ОПК-5 ОПК-6	Зачет

1.5. Описание показателей формирования компетенций

Код по ФГОС ВО	Результаты сформированности
ОПК-2	<p>Знает: современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности</p> <p>Умеет: выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности</p> <p>Владеет навыками: навыки применения современных информационных технологий и программных средств, в том числе отечественного</p>

	производства, при решении задач профессиональной деятельности
ОПК-5	Знает: основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем Умеет: выполнять параметрическую настройку информационных и автоматизированных систем Владеет навыками: инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем
ОПК-6	Знает: основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий Умеет: применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ Владеет навыками: программирования, отладки и тестирования прототипов программно-технических комплексов задач

1.6. Критерии оценивания компетенций на разных этапах их формирования

Вид учебной работы	Количество баллов		
	ОФО	О-ЗФО	ЗФО
Устные ответы на семинарских занятиях			
Выполнение и защита практических / лабораторных работ	45		45
Самостоятельная работа	10		10
Иные виды учебной работы (подготовка презентации, написание реферата, решение задач и др.)	15		15
Зачет	30		30
Всего	100		

Накопительная система оценивания по 100-балльной шкале

Четырехбалльная система оценивания экзамена	100-балльная шкала	Буквенная шкала, соответствующая 100-балльной шкале	Система оценивания зачета
Отлично	90-100	А – отлично – теоретическое содержание курса освоено полностью, без пробелов; необходимые практические навыки работы с освоенным материалом сформированы; все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	Зачтено
Хорошо	83-89	В – очень хорошо – теоретическое содержание курса освоено полностью, без пробелов; необходимые практические навыки работы с освоенным материалом в основном сформированы; все предусмотренные программой обучения учебные задания выполнены, качество выполнения большинства из них оценено числом баллов, близким к максимальному	
Хорошо	75-82	С – хорошо – теоретическое содержание	

		курса освоено полностью; некоторые практические навыки работы с освоенным материалом сформированы недостаточно; все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками	
Удовлетворительно	63-74	D – удовлетворительно – теоретическое содержание курса освоено частично, но пробелы не носят существенного характера; необходимые практические навыки работы с освоенным материалом в основном сформированы; большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки	
Удовлетворительно	50-62	E – посредственно – теоретическое содержание курса освоено частично; некоторые практические навыки работы не сформированы, многие предусмотренные учебной программой обучения учебные задания не выполнены либо качество выполненных некоторых из них оценено числом баллов, близким к минимальному	
Неудовлетворительно	21-49	FX – неудовлетворительно – теоретическое содержание курса освоено частично; необходимые практические навыки работы с освоенным материалом не сформированы; большинство предусмотренных учебной программой обучения учебных заданий не выполнено либо качество их выполнения оценено числом баллов, близким к минимальному; при дополнительно самостоятельной работе над материалом курса возможно повышение качества выполнения учебных заданий	Не зачтено
Неудовлетворительно	0-20	F – неудовлетворительно – теоретическое содержание курса не освоено; необходимые практические навыки работы не сформированы; все выполненные учебные задания содержат грубые ошибки; дополнительная самостоятельная работа над материалом курса не приведет к какому-либо значимому повышению качества выполнения учебных заданий	

2. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА

1.1. Оценочные средства текущего контроля

1. Основные принципы и назначение ООП.
2. Понятие полиморфизма.
3. Понятие инкапсуляции.
4. Понятие наследования.
5. Создание пользовательских классов. Методы и свойства класса.
6. Спецификаторы доступа.
7. Интерфейс и инициализация класса. Объявление, определение, реализация.
8. Указатель this.
9. Принципы наследования и композиции при конструировании новых классов. Преимущества и недостатки наследования по сравнению с композицией.
10. Инициализация данных класса. Конструктор по умолчанию, конструктор с параметрами. Вызов конструкторов объектов, являющихся членами класса.
11. Статические методы и свойства класса. Константные методы.
12. Назначение деструктора. Порядок вызова деструкторов при наследовании. Виртуальные деструкторы.
13. Способы передачи объекта в функцию (по значению, по ссылке, по константной ссылке). Конструктор копирования.
14. Перегрузка оператора присваивания.
15. Дружественные функции и дружественные классы.
16. Перегрузка операторов.
17. Перегрузка операторов с помощью добавления методов класса. Перегрузка операторов с помощью дружественных функций.
18. Виртуальные функции.
19. Чистые виртуальные функции и абстрактные классы.
20. Виртуальный деструктор.
21. Множественное наследование.
22. Виртуальное наследование.
23. Шаблоны классов.
24. Шаблоны функций.
25. Диаграммы классов

Типовые задания по дисциплине

Задание 1

По примеру создать класс с несколькими полями, конструктором по умолчанию и конструктором с параметрами, с методами, меняющим какое-то поле и выводящим информацию объекта на экран. Предмет описания класса взять согласно варианту. Сделать отчет о проделанной работе.

1. Книги
2. Журналы
3. Фильмы
4. Игры

Задание 1

Написать программу согласно указанному варианту и согласно методическим указаниям. Сделать отчет по проведенной работе.

1	<p>Описать класс с именем STUDENT, содержащий следующие поля:</p> <ul style="list-style-type: none"> ■ NAME - фамилия и инициалы; ■ GROUP - номер группы; ■ SES - успеваемость (массив из пяти элементов). <p>Написать программу, которая выполняет следующие действия:</p> <ul style="list-style-type: none"> ■ ввод с клавиатуры данных в массив STUD1, состоящий из десяти объектов типа STUDENT; ■ функцию сортировки для работы с массивом STUD1, записи должны быть упорядочены по возрастанию содержания поля GROUP; ■ функцию для работы с классом для вывода на дисплей фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента более 4.0, если таких студентов нет, вывести соответствующее сообщение.
2	<p>Описать класс с именем STUDENT, содержащий следующие поля:</p> <ul style="list-style-type: none"> ■ NAME - фамилия и инициалы; ■ GROUP - номер группы; ■ SES - успеваемость (массив из пяти элементов). <p>Написать программу, которая выполняет следующие функции:</p> <ul style="list-style-type: none"> ■ ввод с клавиатуры данных в массив STUD1, состоящий из десяти объектов типа STUDENT; ■ функцию сортировки для работы с массивом STUD1, записи должны быть упорядочены по возрастанию среднего балла; ■ функцию для работы с классом для вывода на дисплей фамилий и номеров групп для всех студентов, имеющих оценки только 4 и 5, если таких студентов нет, вывести соответствующее сообщение.
3	<p>Описать класс с именем STUDENT, содержащий следующие поля:</p> <ul style="list-style-type: none"> ■ NAME - фамилия и инициалы; ■ GROUP - номер группы; ■ SES - успеваемость (массив из пяти элементов). <p>Написать программу, которая выполняет следующие действия:</p> <ul style="list-style-type: none"> ■ ввод с клавиатуры данных в массив STUD1, состоящий из десяти объектов типа STUDENT; ■ функцию сортировки для работы с массивом STUD1, записи должны быть упорядочены по алфавиту; ■ функцию для работы с классом для вывода на дисплей фамилий и номеров групп для всех студентов, имеющих хотя бы одну оценку 2, если таких студентов нет, вывести соответствующее сообщение.

Задание 1

№ варианта	Задание
1	Описать класс CPoint, который хранит координаты точки, для него описать класс-наследник ColorePoint, который хранит цвет точки. Для этих классов сделать два класса-наследника CLine и CColoredLine, соответственно. Все классы должны иметь методы для вывода данных, изменения данных.
2	Описать класс CTransport, который хранит скорость, год выпуска, название и стоимость транспортного средства. Для него описать классы-наследники CPlane, CCar, CShip. Добавить в эти классы поля о количестве пассажиров для каждого, высоту полета - для самолета, порт для корабля, владельца и цвет для машины. Все классы должны иметь методы для вывода, изменения и ввода данных.
3	Описать класс CPoint который хранит координаты точки, для него описать класс-наследник SKrug, добавить в него поле радиус и метод нахождения площади. Описать класс SKvadrat, который хранит сторону и методы подсчета площади и периметра, Описать класс-наследник CPrjamougolnik,

	который будет хранить еще одну сторону. Все классы должны иметь методы для вывода, изменения и ввода данных.
--	--

Задание 1

Написать программу согласно указанному варианту и согласно методическим указаниям.
Сделать отчет по проведенной работе.

№ вари- анта	Задание
1	Описать класс, который хранит информацию об играх, которые есть в игротекке: название, жанр, создатель, дата приобретения, поле, содержащее информацию о том, была пройдена эта игра. Описать дружественный класс, который будет иметь метод, чтобы менять значение поля о прохождении игры и жанр игры.
2	Создать класс, который хранит информацию о заказе, принимаемом швейным ателье: номер заказ, заказчик, вид пошива, дата приема, стоимость заказа, поле, содержащее информацию о том, был отдан этот заказ или нет. Написать дружественный класс, который содержит метод, чтобы менять значение поля о выдаче заказа и стоимости.
3	Создать класс лекарств, сохраняющий информацию о товарах аптеки: код товара, название препарата, производитель, лекарственная группа, необходимость наличия рецепта. Написать дружественный класс, который имеет метод, чтобы менять значение поля о наличии рецепта и лекарственной группы.

Задание 1

Написать программу согласно указанному варианту и согласно методическим указаниям.
Сделать отчет о проделанной работе.

Варианты задач

1
Создать класс «поставки товара» со следующими атрибутами: <наименование товара>, <стоимость ед. товара>, <количество товара>, <страна поставки товара>. Добавить метод возвращающий информацию о товаре имеющем цену в заданном интервале. Создать дружественный класс продавец, который может изменять цену на товар и его количество
2
Создать класс «расписание занятий» со следующими атрибутам: <предмет>, <преподаватель>, <номер группы>, <день недели>, <номер пары>, <аудитория>. Добавить метод возвращающий информацию о загруженности заданной аудитории в заданный день недели на заданной паре. Создать дружественный класс диспетчер, который может менять номер пары и аудиторию.
3
Создать класс «поставки товара» со следующими атрибутами: <наименование товара>, <стоимость ед. товара>, <количество товара>, <страна поставки товара>. Добавить метод возвращающий информацию о количестве товара определенного вида поступающего из заданной страны. Создать дружественный класс поставщик, который может изменять цену и количество товара.

Задание 1

Написать программу согласно указанному варианту и согласно методическим указаниям.
Сделать отчет о проделанной работе.

№	варианты заданий
---	------------------

1.	Написать шаблон класса, который вычисляет периметр треугольника. Класс должен содержать конструктор по умолчанию, виртуальную функцию для вычисления периметра. Создать производный класс для равностороннего треугольника, в котором переопределить функцию вычисления периметра. В базовый класс добавить перегруженный оператор $>$, который возвращает истину, если периметр первого треугольника больше второго. В главной функции вычислить периметры треугольника со сторонами 2, 3 и 5, а также равностороннего треугольника со стороной 4. Определить, используя перегруженный оператор, периметр первого треугольника больше ли периметра второго.
2.	Написать шаблон класса, вычисляющего объем параллелепипеда. Класс должен содержать конструктор по умолчанию, виртуальную функцию для вычисления объема. Создать производный класс для куба, в котором переопределить функцию вычисления объема. В базовый класс добавить перегруженный оператор $<$, который возвращающий истину, если объем первой фигуры меньше второго. В главной функции вычислить объем параллелепипеда со сторонами 2.3, 1.5, 3.0, а также куба с длиной ребра 2.0. Используя перегруженный оператор определить, объем параллелепипеда больше куба.
3.	Написать шаблон класса, который вычисляет длину окружности эллипса. Класс должен содержать конструктор по умолчанию, виртуальную функцию для вычисления длины окружности. Создать производный класс для круга. В базовый класс добавить перегруженный оператор $<$, который возвращающий истину, если площадь первой фигуры меньше площади второй. Продемонстрировать работу программы.

Задание 1

№ п/п	варианты заданий
1.	Написать программу, используя класс Дробное число со знаком. Число должно быть представлено двумя полями: целая часть - длинное целое со знаком, дробная часть - беззнаковое короткое целое. Реализовать арифметические операции сложения, вычитания, умножения и операции сравнения. В функции main проверить эти методы.
2.	Написать программу, используя класс Деньги для работы с денежными суммами. Число должно быть представлено двумя полями: типа long для рублей и типа unsigned char - для копеек. Дробная часть (копейки) при выводе на экран должна быть отделена от целой части запятой. Реализовать сложение, вычитание, деление сумм, деление суммы на дробное число, умножение на дробное число и операции сравнения. В функции main проверить эти методы.
3.	Написать программу, используя класс Равнобокая трапеция, члены класса: координаты 4-х точек. Предусмотреть в классе конструктор и методы: проверка, является ли фигура равнобокой трапецией; вычисления и вывода сведений о фигуре: длины сторон, периметр, площадь. В функции main продемонстрировать работу с классом: дано N трапеций, найти количество трапеций, у которых площадь больше средней площади.
4.	Написать программу, используя класс Прямоугольник, члены класса: координаты 4-х точек. Предусмотреть в классе конструктор и методы: проверка, является ли фигура прямоугольником; вычисления и вывода сведений о фигуре: длины сторон, периметр, площадь. В функции main продемонстрировать работу с классом: дано

	Н прямоугольником, найти количество прямоугольников, у которых площадь больше средней площади.
5.	Написать программу, используя класс Окружность, члены класса: координаты центра и радиус. Предусмотреть в классе конструктор и методы: вычисления и вывода сведений о фигуре: длины окружности, площадь. В функции main продемонстрировать работу с классом: дано N окружностей, найти количество окружностей, у которых площадь больше средней площади и количество окружностей с заданным радиусом.
6.	Написать программу, используя класс Комплексное число. Число должно быть представлено двумя полями: Действительное число - дробное со знаком, мнимой часть - дробное со знаком. Реализовать арифметические операции сложения, вычитания, умножения и операции сравнения. В функции main проверить эти методы.
7.	Написать программу, используя класс Треугольник, члены класса: координаты 3-х точек. Предусмотреть в классе конструктор и методы: проверка, является ли фигура треугольником; вычисления и вывода сведений о фигуре: длины сторон, периметр, площадь. В функции main продемонстрировать работу с классом: дано N треугольником, найти количество треугольников, у которых площадь больше средней площади.
8.	Написать программу, используя класс Треугольник, члены класса: координаты 3-х точек. Предусмотреть в классе конструктор и методы: проверка, является ли фигура треугольником; вычисления и вывода сведений о фигуре: длины сторон, периметр, площадь, тип. В функции main продемонстрировать работу с классом: дано N треугольником, найти количество прямоугольных треугольников, равнобедренных, равносторонних.

1.2. Оценочные средства для промежуточной аттестации

1. Какие основные принципы ООП существуют?
2. Что такое класс и объект в C++?
3. Как описать класс в ООП?
4. Зачем нужны объекты в ООП?
5. Как объявить класс в C++?
6. Чем класс отличается от структуры?
7. Как создать объект класса в C++?
8. Опишите класс студент с полями фамилия, имя, отчество, группа, курс, форма обучение.
9. Опишите класс лекарств, сохраняющий информацию о товарах аптеки: код товара, название препарата, производитель, лекарственная группа, необходимость наличия рецепта.
10. Описать класс «поставки товара» со следующими атрибутами: <наименование товара>, <стоимость ед. товара>, <количество товара>, <страна поставки товара>. Добавить метод возвращающий информацию о товаре имеющем цену в заданном интервале.
11. Что такое конструктор класса и как его объявить в C++?
12. Что такое конструктор по умолчанию?
13. Что дало ООП программистам?
14. Разница между процедурным программированием и ООП?
15. Зачем использовать ООП?
16. Каковы основные концепции ООП?
17. Какие языки относятся к концепции ООП?
18. Что такое абстракция?

19. Что такое перегрузка методов?
20. Что такое переопределение методов?
21. Типы наследования в ООП
22. Типы конструкторов
23. Что такое многоуровневое наследование?
24. Что такое гибридное наследование?
25. Что такое иерархическое наследование?
26. Каковы ограничения наследования?
27. Что такое суперкласс?
28. Проведите различие между перегрузкой и переопределением.
29. Что такое базовый класс?
30. В чем разница между модификаторами доступа public, private и protected?
31. Как достичь абстракции данных?
32. Проведите различие между абстракцией данных и инкапсуляцией
33. Зачем нужны защищенные члены класса?
34. Что такое деструктор класса и как его объявить в C++?
35. Как вызвать деструктор класса?
36. Как вызвать конструктор класса?
37. Что такое наследование в ООП и как его реализовать в C++?
38. Можно ли обойтись без ООП?
39. Можно ли обойтись без наследования в ООП?
40. Опишите иерархию классов: человек-> студент со всеми необходимыми полями.
41. Как объявить и использовать наследование в C++?
42. Что происходит, когда открытые члены базового класса наследуются как открытые?
43. Что происходит, когда открытые члены базового класса наследуются как закрытые?
44. Что такое полиморфизм в ООП и как его реализовать в C++?
45. Как объявить и использовать полиморфизм в C++ с помощью виртуальных функций?
46. Что такое абстрактный класс (abstract class) в объектно-ориентированном программировании?
47. Как объявить абстрактный класс в C++?
48. Что такое метод в ООП?
49. Что такое виртуальная функция (virtual function) в C++?
50. Как объявить виртуальную функцию в абстрактном классе?
51. Что такое чисто виртуальная функция (pure virtual function) в C++?
52. Как объявить чисто виртуальную функцию в абстрактном классе?
53. Можно ли создать объект абстрактного класса в C++?
54. Как использовать абстрактные классы для создания интерфейсов в C++?
55. Как использовать виртуальные функции для достижения полиморфизма в C++?
56. Что такое переопределение функций (function overriding) в C++?
57. Как переопределить виртуальную функцию в производном классе?
58. Что такое виртуальный деструктор (virtual destructor) в C++ и зачем он нужен?
59. Как объявить и использовать виртуальный деструктор в абстрактном классе?
60. Как использовать абстрактные классы и виртуальные функции для создания иерархии классов в C++?
61. Как использовать указатели и ссылки на абстрактные классы в C++?
62. Как проверить, является ли объект экземпляром определенного абстрактного класса в C++?
63. Какие преимущества и ограничения связаны с использованием абстрактных классов и виртуальных функций в C++?

64. Как объявить и использовать абстрактный класс и интерфейс в C++?
65. Что такое инкапсуляция в ООП и как ее реализовать в C++?
66. Что такое модификаторы доступа в C++ и как они работают?
67. Как использовать наследование и полиморфизм для создания иерархии классов в C++?
68. Что такое композиция и агрегация в C++?
69. Как использовать композицию и агрегацию для создания связанных объектов в C++?
70. Какие модификаторы доступа существуют в C++ и как они работают?
71. Зачем нужен спецификатор `private`?
72. Зачем нужен спецификатор `public`?
73. Зачем нужен спецификатор `protected`?
74. Что такое конструкторы копирования класса в C++ и как они объявляются?
75. Что такое множественное наследование в C++ и как его использовать?
76. Как создать иерархию классов?
77. Как объявить и использовать множественное наследование в C++?
78. Что такое полиморфизм в C++ и как он реализуется?
79. Что такое исключения в C++ и как они обрабатываются?
80. Что такое перегрузка функций в C++?
81. Что такое перегрузка операторов в C++?
82. Как объявить и использовать перегрузку функций в C++?
83. Как объявить и использовать перегрузку операторов в C++?
84. Что такое статические члены класса в C++ и как они работают?
85. Как объявить и использовать статические члены класса в C++?
86. Что такое константные члены класса в C++ и как они работают?
87. Как объявить и использовать константные члены класса в C++?
88. Объявление элементов класса спецификацией `static`
89. Объявление элементов класса спецификацией `const`
90. Объявление объекта и доступ к его элементам
91. Объекты в качестве возвращаемого значения функции
92. При наследовании одного класса другим, когда вызываются конструкторы классов?
93. При наследовании одного класса другим, когда вызываются их деструкторы?
94. Что такое исключения в C++ и как они обрабатываются?
95. Как объявить и использовать исключения в C++?
96. Что такое шаблоны (templates) в C++ и как они работают?
97. Как объявить и использовать шаблоны (templates) в C++?
98. Как объявить и использовать шаблоны (templates) для создания обобщенных функций и классов в C++?
99. Что такое стандартные библиотеки (STL) в C++ и как они используются?
100. Контейнеры что это?
101. Как использовать контейнеры и алгоритмы из стандартных библиотек (STL) в C++?
102. Что такое файлы и потоки ввода-вывода в C++?
103. Как открыть, записать и закрыть файл в C++?
104. Как использовать потоки ввода-вывода для чтения и записи данных в C++?
105. Что такое вектор (vector) в C++ и как он отличается от обычного массива?
106. Как объявить и инициализировать вектор в C++?
107. Как добавить элемент в конец вектора в C++?

108. Как получить доступ к элементам вектора в C++?
109. Как удалить элемент из вектора в C++?
110. Что такое множество (set) в C++ и как оно работает?
111. Как объявить и инициализировать множество в C++?
112. Как добавить элемент в множество в C++?
113. Как удалить элемент из множества в C++?
114. Что такое ассоциативный массив (map) в C++ и как он работает?
115. Как объявить и инициализировать ассоциативный массив в C++?
116. Как добавить элемент в ассоциативный массив в C++?
117. Как получить доступ к элементам ассоциативного массива в C++?
118. Что такое итераторы в C++ и как они используются?
119. Что такое константные итераторы в C++ и как они используются?
120. Что такое реверсные итераторы в C++ и как они используются?
121. Класс string зачем нужен и как использовать?
122. Что такое статическое связывание и динамическое связывание в C++?
123. Как работает статическое и динамическое связывание в C++?
124. Как объявить и использовать пространства имен (namespaces) в C++?
125. Что такое лямбда-выражения (lambda expressions) в C++ и как их использовать?
126. Как объявить и использовать лямбда-выражения в C++?
127. Делегаты в ООП?
128. Что такое предикаты (predicates) в C++ и как их использовать?
129. Что такое раннее и позднее связывание (early binding vs. late binding) в программировании?
130. Как работает раннее связывание и какие преимущества оно имеет?
131. Как работает позднее связывание и какие преимущества оно имеет?
132. Что такое диаграмма классов (class diagram) в объектно-ориентированном программировании?
133. Какие элементы используются в диаграммах классов и как они представляются?
134. Какие отношения между классами могут быть представлены на диаграммах классов?
135. Как представляются классы, атрибуты и методы на диаграммах классов?
136. Какие виды отношений между классами могут быть представлены на диаграммах классов?
137. Что такое ассоциация (association) и как она представляется на диаграммах классов?
138. Что такое наследование (inheritance) и как оно представляется на диаграммах классов?
139. Что такое агрегация (aggregation) и композиция (composition) и как они представляются на диаграммах классов?
140. Что такое интерфейс (interface) и как он представляется на диаграммах классов?
141. Какие дополнительные элементы могут быть представлены на диаграммах классов, например, стереотипы, шаблоны и т. д.?
142. Как читать и интерпретировать диаграмму классов?
143. Какие инструменты и программы можно использовать для создания диаграмм классов?
144. Как использование диаграмм классов может помочь в процессе проектирования и разработки программного обеспечения?
145. Какие языки программирования поддерживают использование диаграмм классов для описания структуры классов и их отношений?

146. Какие типичные ошибки можно совершить при создании диаграмм классов и как их избежать?
147. Постройте диаграмму классов к иерархии: Транспорт->Автомобиль: пассажирский; грузовой.
148. Описать класс Деньги для работы с денежными суммами. Число должно быть представлено двумя полями: типа long для рублей и типа unsigned char - для копеек. Дробная часть (копейки) при выводе на экран должна быть отделена от целой части запятой. Реализовать сложение, вычитание.
149. Описать класс Дробное число со знаком. Число должно быть представлено двумя полями: целая часть - длинное целое со знаком, дробная часть - беззнаковое короткое целое. Реализовать арифметические операции сложения, вычитания, умножения и операции сравнения.
150. Описать шаблон класса, вычисляющего объем параллелепипеда.
151. Описать класс CPoint, который хранит координаты точки, для него описать класс-наследник ColorePoint, который хранит цвет точки.
152. Описать класс, который хранит информацию об играх, которые есть в игротекке: название, жанр, создатель, дата приобретения, поле, содержащее информацию о том, была пройдена эта игра. Описать дружественный класс, который будет иметь метод, чтобы менять значение поля о прохождении игры и жанр игры.