

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

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

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

УТВЕРЖДАЮ

Врио директора ИФМОИОТ

Е.А. Журавлёва

«14» января 2026 г.



Приложение к рабочей программе учебной дисциплины

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

По направлению подготовки 44.03.01 Педагогическое образование

Профиль подготовки Компьютерные системы и образовательная робототехника

Квалификация выпускника – бакалавр

Форма обучения очная

Курс ОФО – 2 курс

Разработчик
канд. физ.-мат. наук, доцент кафедры
информационных образовательных
технологий и систем Швыров В.В.

заведующий кафедрой
информационных образовательных
технологий и систем
Капустин Д.А. Капустин

Протокол

от «13» января 2026 г. № 11

Луганск, 2026

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

1.1. Область применения

Фонд оценочных средств (ФОС) – неотъемлемая часть рабочей программы дисциплины (модуля) Технологии объектно-ориентированного программирования и предназначен для контроля и оценки образовательных достижений студентов, освоивших программу дисциплины (модуля).

1.2. Цели и задачи фонда оценочных средств

Цель ФОС – установить соответствие уровня подготовки обучающегося требованиям ФГОС ВО – бакалавриат по направлению подготовки 44.03.01 «Педагогическое образование», утвержденным приказом Министерства образования и науки Российской Федерации от 22.02.2018 № 121 (с изменениями и дополнениями) и Профессиональным стандартом, утвержденным Приказом Министерства труда и социальной защиты Российской Федерации «Об утверждении профессионального стандарта «Педагог (педагогическая деятельность в дошкольном, начальном общем, основном общем, среднем общем образовании) (воспитатель, учитель)» от 08.10.2013 № 544н.

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
Профессиональные	
ПК-4. Способен разрабатывать требования и проектировать программное обеспечение, а также программно-аппаратные комплексы.	ПК-4.1. Знать: методы и технологию анализа и проектирования требований к программному обеспечению процессов и систем с заданной структурой и функциональными свойствами.
	ПК-4.2. Уметь: осуществлять разработку требований и проектирование технических заданий на разработку программного обеспечения программно-аппаратных комплексов.
	ПК-4.3. Владеть: навыками разработки программного обеспечения технологических процессов обучающей организации.

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.	ПК-4	Выполнение лабораторных работ
Тема 2. Конструкторы и деструкторы.	ПК-4	Выполнение лабораторных работ
Тема 3. Иерархия классов. Наследование.	ПК-4	Выполнение лабораторных работ
Тема 4. Дружественные классы и функции.	ПК-4	Выполнение лабораторных работ
Тема 5. Виртуальные классы и функции.	ПК-4	Выполнение лабораторных работ
Тема 6. Абстрактные классы.	ПК-4	Выполнение лабораторных работ
Тема 7. Перегрузка операторов.	ПК-4	Выполнение лабораторных работ
Тема 8. Шаблоны классов.	ПК-4	Выполнение лабораторных работ
Тема 9. Диаграммы классов.	ПК-4	Выполнение лабораторных работ
Текущая аттестация	ПК-4	индивидуальная работа
Промежуточная аттестация	ПК-4	Экзамен

1.5. Описание показателей формирования компетенций

Код компетенции	Результаты сформированности
ПК-4. Способен разрабатывать требования и проектировать программное обеспечение, а также программно-аппаратные комплексы.	<p>ПК-4.1. Знает: методы и технологию анализа и проектирования требований к программному обеспечению процессов и систем с заданной структурой и функциональными свойствами.</p> <p>ПК-4.2. Умеет: осуществлять разработку требований и проектирование технических заданий на разработку программного обеспечения программно-аппаратных комплексов.</p> <p>ПК-4.3. Владеет: навыками разработки программного обеспечения технологических процессов обучающей организации.</p>

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	40		40
Самостоятельная работа	10		10
Иные виды учебной работы (подготовка презентации,	15		15

написание реферата, решение задач и др.)			
Экзамен	35		35
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	Д – удовлетворительно – теоретическое содержание дисциплины 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	Е – посредственно – теоретическое содержание курса освоено частично; некоторые практические навыки работы не сформированы, многие предусмотренные программой обучения учебные задания не выполнены либо качество выполнения некоторых из них оценено числом баллов, близким к минимальному	

Неудовлетворительно	21–49	FX – неудовлетворительно – теоретическое содержание курса освоено частично; необходимые практические навыки работы не сформированы; большинство предусмотренных программой обучения учебных заданий не выполнено либо качество их выполнения оценено числом баллов, близким к минимальному; при дополнительной 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

2.1. 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.

№ варианта	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.

2.2. Оценочные средства для промежуточной аттестации

3. Что такое объектно-ориентированное программирование (ООП)?
4. Какие основные принципы ООП существуют?
5. Объясните принципы инкапсуляции
6. Как инкапсуляция помогает защитить данные?
7. Как реализуется инкапсуляция в C++?
8. Что такое модификаторы доступа (private, protected, public)?
9. Как использовать геттеры и сеттеры для доступа к данным?
10. Какие преимущества дает инкапсуляция в программировании?
11. Как инкапсуляция способствует поддерживаемости кода?
12. Какие примеры инкапсуляции можно найти в стандартной библиотеке C++?
13. Как инкапсуляция связана с другими принципами ООП (например, наследованием и полиморфизмом)?
14. Объясните принципы наследования
15. Какие типы наследования существуют?
16. Как объявить класс-наследник в C++?
17. Что такое базовый (родительский) и производный (дочерний) классы?
18. Как использовать ключевые слова public, protected и private при наследовании?
19. Что такое множественное наследование и как его использовать?
20. Какие преимущества дает наследование в программировании?
21. Как работает конструктор и деструктор в контексте наследования?
22. Что такое виртуальное наследование и когда его использовать?
23. Объясните принципы полиморфизма
24. Какие виды полиморфизма существуют?

25. Как реализуется полиморфизм в C++?
26. Что такое перегрузка функций и операторов?
27. Что такое виртуальные функции и как они работают?
28. Как использовать абстрактные классы и интерфейсы для достижения полиморфизма?
29. Какие преимущества дает полиморфизм в программировании?
30. Как работает позднее связывание (late binding)?
31. Какие примеры полиморфизма можно найти в стандартной библиотеке C++?
32. Что такое класс и объект в C++?
33. Как объявить класс в C++?
34. Как создать объект класса в C++?
35. Как создать объект класса?
36. Что такое конструктор класса и как его объявить в C++?
37. Что такое конструктор по умолчанию?
38. Напишите класс для представления денег с операциями сложения и умножения.
39. Что дало ООП программистам?
40. Что такое деструктор класса и как его объявить в C++?
41. Что такое наследование в ООП и как его реализовать в C++?
42. Как объявить и использовать наследование в C++?
43. Зачем нужен указатель this?
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 виртуальная функция (virtual function) в C++?
49. Как объявить виртуальную функцию в абстрактном классе?
50. Что такое чисто виртуальная функция (pure virtual function) в C++?
51. Как объявить чисто виртуальную функцию в абстрактном классе?
52. Можно ли создать объект абстрактного класса в C++?
53. Как использовать абстрактные классы для создания интерфейсов в C++?
54. Как использовать виртуальные функции для достижения полиморфизма в C++?
55. Что такое виртуальный деструктор?
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. Что такое статические члены класса?
64. Что такое дружественные функции и классы?
65. Какие преимущества и ограничения связаны с использованием абстрактных классов и виртуальных функций в C++?
66. Как объявить и использовать абстрактный класс и интерфейс в C++?
67. Что такое инкапсуляция в ООП и как ее реализовать в C++?
68. Что такое модификаторы доступа в C++ и как они работают?
69. Как использовать наследование и полиморфизм для создания иерархии классов в C++?
70. Что такое композиция и агрегация в C++?
71. Как использовать композицию и агрегацию для создания связанных объектов в C++?
72. Какие модификаторы доступа существуют в C++ и как они работают?
73. Что такое конструкторы копирования класса в C++ и как они объявляются?
74. Что такое множественное наследование в C++ и как его использовать?

75. Создайте иерархию классов для представления различных типов транспортных средств (например, автомобиль, велосипед, самолет).
76. Как объявить и использовать множественное наследование в C++?
77. Что такое полиморфизм в C++ и как он реализуется?
78. Что такое исключения в C++ и как они обрабатываются?
79. Что такое перегрузка функций в C++?
80. Что такое перегрузка операторов в C++?
81. Как объявить и использовать перегрузку функций в C++?
82. Как объявить и использовать перегрузку операторов в C++?
83. Что такое статические члены класса в C++ и как они работают?
84. Как объявить и использовать статические члены класса в C++?
85. Что такое константные члены класса в C++ и как они работают?
86. Как объявить и использовать константные члены класса в C++?
87. Что такое шаблоны (templates) в C++ и как они работают?
88. Как объявить и использовать шаблоны (templates) в C++?
89. Как объявить и использовать шаблоны (templates) для создания обобщенных функций и классов в C++?
90. Что такое стандартные библиотеки (STL) в C++ и как они используются?
91. Как использовать контейнеры и алгоритмы из стандартных библиотек (STL) в C++?
92. Что такое файлы и потоки ввода-вывода в C++?
93. Как открыть, записать и закрыть файл в C++?
94. Как использовать потоки ввода-вывода для чтения и записи данных в C++?
95. Что такое вектор (vector) в C++ и как он отличается от обычного массива?
96. Как объявить и инициализировать вектор в C++?
97. Как добавить элемент в конец вектора в C++?
98. Как получить доступ к элементам вектора в C++?
99. Как удалить элемент из вектора в C++?
100. Что такое множество (set) в C++ и как оно работает?
101. Как объявить и инициализировать множество в C++?
102. Как добавить элемент в множество в C++?
103. Как удалить элемент из множества в C++?
104. Что такое ассоциативный массив (map) в C++ и как он работает?
105. Как объявить и инициализировать ассоциативный массив в C++?
106. Как добавить элемент в ассоциативный массив в C++?
107. Как получить доступ к элементам ассоциативного массива в C++?
108. Зачем нужны итераторы?
109. Что такое итераторы?
110. Какие бывают итераторы?
111. Что такое статическое связывание и динамическое связывание в C++?
112. Как работает статическое и динамическое связывание в C++?
113. Как объявить и использовать пространства имен (namespaces) в C++?
114. Что такое лямбда-выражения (lambda expressions) в C++ и как их использовать?
115. Как объявить и использовать лямбда-выражения в C++?
116. Что такое предикаты (predicates) в C++ и как их использовать?
117. Что такое раннее и позднее связывание (early binding vs. late binding) в программировании?
118. Что такое диаграмма классов (class diagram) в объектно-ориентированном программировании?
119. Какие элементы используются в диаграммах классов и как они представляются?
120. Какие отношения между классами могут быть представлены на диаграммах классов?
121. Как представляются классы, атрибуты и методы на диаграммах классов?
122. Какие виды отношений между классами могут быть представлены на диаграммах классов?
123. Что такое ассоциация (association) и как она представляется на диаграммах классов?
124. Что такое наследование (inheritance) и как оно представляется на диаграммах классов?

125. Что такое агрегация (aggregation) и композиция (composition) и как они представляются на диаграммах классов?
126. Какие дополнительные элементы могут быть представлены на диаграммах классов, например, стереотипы, шаблоны и т. д.?
127. Что такое диаграмма классов?
128. Какие основные элементы диаграммы классов?
129. Как обозначаются классы и их атрибуты на диаграмме?
130. Как обозначаются методы классов на диаграмме?
131. Какие типы отношений между классами существуют и как они обозначаются?
132. Что такое ассоциация, агрегация и композиция?
133. Как обозначаются наследование и реализация интерфейсов?
134. Как использовать диаграммы классов для проектирования системы?
135. Как интерпретировать и анализировать диаграммы классов?
136. Какие инструменты и программы можно использовать для создания диаграмм классов?
137. Как использование диаграмм классов может помочь в процессе проектирования и разработки программного обеспечения?
138. Какие языки программирования поддерживают использование диаграмм классов для описания структуры классов и их отношений?
139. Какие типичные ошибки можно совершить при создании диаграмм классов и как их избежать?
140. Как обозначаются зависимости между классами на диаграмме?
141. Что такое диаграмма объектов и как она связана с диаграммой классов?
142. Как использовать диаграммы классов для документирования существующей системы?
143. Какие существуют типы связей между классами и как они обозначаются?
144. Как обозначаются абстрактные классы и методы на диаграмме?
145. Как использовать диаграммы классов для планирования архитектуры системы?

Перечень практических заданий к экзамену по дисциплине «Технологии объектно-ориентированного программирования»:

№ п/п	Перечень практических заданий к экзамену
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 N 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.