



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)

«СОГЛАСОВАНО»

Руководитель ОП

Артемьева И.Л.

«УТВЕРЖДАЮ»

И.о. директора департамента

Смагин С.В.

«23» марта 2022 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Разработка объектно-ориентированных приложений

**Направление подготовки 02.03.03 Математическое обеспечение и администрирование
информационных систем
(Технология программирования)**

Форма подготовки очная

курс 2 семестр 4
лекции 00 час.
практические занятия 00 час.
лабораторные работы 54 час.
в том числе с использованием МАО лек. 0 / пр. 0 / лаб. 54 час.
всего часов аудиторной нагрузки 54 час.
в том числе с использованием МАО 54 час.
самостоятельная работа 90 час.
в том числе на подготовку к экзамену 36 час.
контрольные работы (количество) не предусмотрены
курсовая работа / курсовой проект не предусмотрены
зачет 0 семестр
экзамен 4 семестр

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта по направлению 02.03.03 Математическое обеспечение и администрирование информационных систем, утвержденного приказом Министерства образования и науки РФ от 23 августа 2017 г. № 809 (с изменениями и дополнениями).

Рабочая программа обсуждена на заседании департамента программной инженерии и искусственного интеллекта, протокол № 3.0 от «23» марта 2022 г.

И.о. директора департамента программной инженерии и искусственного интеллекта к.т.н. Смагин С.В.

Составители: И.Л.Артемьева д.т.н., профессор

Владивосток
2022

Оборотная сторона титульного листа РПД

I. Рабочая программа пересмотрена на заседании департамента:

Протокол от « ____ » _____ 20__ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

II. Рабочая программа пересмотрена на заседании департамента:

Протокол от « ____ » _____ 20__ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

III. Рабочая программа пересмотрена на заседании департамента:

Протокол от « ____ » _____ 20__ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

IV. Рабочая программа пересмотрена на заседании департамента:

Протокол от « ____ » _____ 20__ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель дисциплины - усвоение и закрепление методов создания объектно-ориентированных программ на C++, знакомство с основными понятиями в рамках данного подхода.

Задачи:

- Изучить основные концептуальные положения объектно-ориентированного программирования, а также механизмы, методы и средства разработки приложений в рамках данного направления

- Изучить язык программирования C++, научиться грамотно его использовать.

- Научиться использовать методы разработки объектно-ориентированных программ.

Планируемые результаты обучения по данной дисциплине (знания, умения, владения), соотнесенные с планируемыми результатами освоения образовательной программы.

В результате изучения данной дисциплины у обучающихся формируются следующие профессиональные компетенции:

Тип задачи профессиональной деятельности	Код и наименование профессиональной компетенции	Код и наименование индикатора достижения компетенции
производственно-технологический	ПК-6. Способность использовать навыки моделирования, анализа и формальных методов конструирования программного обеспечения	ПК-6.1 демонстрирует знание современных приемов работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов, их сопровождения и администрирования ПК-6.2. использует современные инструментальные средства в практической деятельности ПК-6.3. применяет современные инструментальные средства при разработке программных продуктов
производственно-технологический	ПК-8. Способен использовать основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки	ПК-8.1. демонстрирует знание основных концептуальных положений функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений ПК-8.2. использует основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования

	программ в рамках этих направлений	ПК-8.3. применяет основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений при создании программных систем
--	------------------------------------	---

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
ПК-6.1 демонстрирует знание современных приемов работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов, их сопровождения и администрирования	Знает современные приемы работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов, их сопровождения и администрирования
	Умеет применять современные приемы работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов, их сопровождения и администрирования
	Владеет приемами работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов, их сопровождения и администрирования
ПК-6.2. использует современные инструментальные средства в практической деятельности	Знает методы проектирования объектно-ориентированных приложений; методы оценки качества проекта объектно-ориентированных приложений
	Умеет проектировать требуемый набор классов и методов работы с ними при создании объектно-ориентированных приложений; проектировать требуемый набор тестов для оценки качества объектно-ориентированных приложений
	Владеет навыками разработки объектно-ориентированных программных средств по проекту; навыками создания пакета тестов для оценки качества объектно-ориентированных приложений
ПК-6.3. применяет современные инструментальные средства при разработке программных продуктов	Знает методы проектирования программных средств
	Умеет определять структуру программных средств
	Владеет приемами работы с инструментальными средствами при разработке объектно-ориентированных приложений
ПК-8.1. демонстрирует знание основных концептуальных положений функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений	Знает основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений
	Умеет использовать структурное и объектно-ориентированное проектирование при разработке ПО
	Владеет современными технологиями разработки ПО
ПК-8.2. использует основные концептуальные положения функционального, логического,	Знает современные технологии разработки ПО (структурное, объектно-ориентированное), принципы наследования, инкапсуляции и полиморфизма,

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
объектно-ориентированного и визуального направлений программирования	положенные в разработку объектно-ориентированных языков
	Умеет использовать методы объектно-ориентированного программирования при создании программных систем
	Владеет методами, способами и программными средствами для разработки объектно-ориентированных программ
ПК-8.3. применяет основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений при создании программных систем	Знает основные положения и концепции объектно-ориентированного программирования; принципы наследования, инкапсуляции и полиморфизма, положенные в разработку объектно-ориентированных языков
	Умеет использовать принципы наследования, инкапсуляции и полиморфизма при создании объектно-ориентированных приложений
	Владеет методами проектирования и разработки программ, используя полиморфные функции и операции, разные типы наследования и создавая методы доступа к объектам разных классов (инкапсуляция)

Для формирования вышеуказанных компетенций в рамках дисциплины «Разработка объектно-ориентированных приложений» применяются следующие методы активного/ интерактивного обучения: метод проектов.

Трудоёмкость дисциплины и видов учебных занятий по дисциплине

Общая трудоёмкость дисциплины составляет 4 зачётных единицы (144 часа) (1 зачетная единица соответствует 36 академическим часам).

Видами учебных занятий и работы обучающегося по дисциплине являются:

Обозначение	Виды учебных занятий и работы обучающегося
Лек	Лекции
Лаб	Лабораторные работы
СР	Самостоятельная работа обучающегося в период теоретического обучения
Контроль	Самостоятельная работа обучающегося и контактная работа обучающегося с преподавателем в период промежуточной аттестации

Структура дисциплины:

Форма обучения – очная.

№	Наименование раздела дисциплины	Семестр	Количество часов по видам учебных занятий и работы обучающегося						Формы промежуточной аттестации
			Лек	Лаб	Пр	ОК	СР	Контроль	
1	Раздел I. Основные принципы объектно-ориентированного программирования	4	-	-	-	-	54	36	УО-1; УО-3; ПР-2; ПР-6;
2	Раздел 2 Основы программирования на языке C++	4	-	54	-	-			
Итого:			-	54	-	-	54	36	

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Лекционные занятия по дисциплине не предусмотрены учебным планом. Студенты изучают материал самостоятельно.

Раздел I. Основные принципы объектно-ориентированного программирования

Тема 1. Основные понятия объектно-ориентированного подхода в программировании.

Основные принципы ООП: наследование инкапсуляция, полиморфизм. Понятие класс, экземпляр класса, метод класса, интерфейс. История развития концепции ООП, примеры объектно-ориентированных языков.

Раздел I. Основы программирования на языке C++

Тема 2. Основные элементы языка C++

Типы данных, их размерность, реализация управляющих конструкций в C++, организация ввода-вывода данных, манипуляторы потока вывода. Основные библиотеки для работы со стандартными типами данных.

Тема 3. Функции в C++ и реализация простых структур

Синтаксис описания функций, рекурсивные функции, перегрузка. Функции с переменным числом параметров, прототипы функций. Передача параметров функции по значению, по ссылке по указателю. Массив, индексация, организация работы с массивом.

Тема 4. Модульное программирование

Многофайловые программы – принцип организации и доступа к отдельным файлам, файлы исходного кода и заголовочные файлы. Область видимости переменных, описание глобальных переменных.

Тема 5. Работа с памятью – динамическое и статическое выделение памяти. Указатели.

Способы выделения динамической памяти, исключение утечек памяти. Указатели – синтаксис и назначение, арифметика указателей. Работа с массивами в динамической памяти, организация линейных списков, двухсвязных списков и бинарных деревьев.

Тема 6. Пользовательские типы данных

Структура, объединение, перечислимый тип данных – синтаксис, назначение, основные принципы разработки указанных типов.

Тема 7. Классы, наследование, специальные методы классов

Синтаксис описания класса, реализация методов, тип доступа, специфика наследования с различным типом доступа. Одиночное и множественное наследование. Статические и константные методы класса. Конструктор и деструктор.

Тема 8. Перегрузка операторов. Виртуальные функции.

Дружественные функции. Дружественные классы

Специфика перегрузки операторов, наследование перегруженных операций. Абстрактный класс, виртуальная функция и чистая виртуальная функция. Переопределение виртуальных методов.

Тема 9. Шаблоны функций и классов. Обработка исключительных ситуаций.

Синтаксис, назначение и принцип разработки шаблонов функций и классов. Конструкция try...catch с параметрами и без.

Тема 10. Библиотека STL, основные элементы. Итераторы и аллокаторы. Контейнерные классы.

Элементы библиотеки STL с прямым и ассоциативным доступом.

Тема 11. Организация многопоточных программ. Мьютексы.

Параллельная обработка данных с использованием потоков, мьютексы – синтаксис, назначение, принцип работы.

II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА

Практические занятия (0 час.)

Практические занятия учебным планом не предусмотрены.

Лабораторные работы (54 час.)

Лабораторная работа 1. Объекты и классы C++. (9 часов)

Получение базовых знаний в области объектно-ориентированного программирования, изучение основных понятий и парадигм: определение класса для заданной задачи, программирование методов класса, типы конструкторов, прототипы функций.

Лабораторная работа 2. Работа с классами в C++. (9 часов).

Получение практических навыков в области перегрузки операторов.

Разработка классов для работы с данными разных типов: стек, список, вектор, комплексные числа и т.д. Перегрузка операций. Дружественные функции.

Лабораторная работа 3. Классы и динамическое выделение памяти. (9 часов)

Получение практических навыков в выделении динамической памяти под объекты классов.

Лабораторная работа 4. Наследование. (9 часов)

Получение практических навыков в области одиночного и множественного наследования классов. Базовый класс. Виртуальные методы.

Лабораторная работа 5. Шаблоны функций и классов. Обработка исключительных ситуаций. (9 часов)

Получение практических навыков работы в области разработки дружественных классов

Лабораторная работа 6. Организация многопоточных программ. Мьютексы. (9 часов)

Получение практических навыков работы с потоками ввода и вывода C++.

Задания для самостоятельной работы

Примеры задач для самостоятельного решения

Требования: Перед каждой самостоятельной работой обучающемуся необходимо изучить Методические указания по выполнению лабораторных работ по дисциплине «Объектно-ориентированное программирование».

Самостоятельная работа №1.

Описать структуру с именем STUDENT, содержащую следующие поля: NAME—фамилия и инициалы; GROUP—номер группы; SES—успеваемость (массив из пяти элементов).

Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в массив STUD1, состоящий из десяти структур типа STUDENT; записи должны быть упорядочены по возрастанию содержимого поля GROUP; вывод на дисплей фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4,0; если таких студентов нет, вывести соответствующее сообщение.

Самостоятельная работа №2.

Простой список можно описать следующим образом:

- простой список может содержать ноль или более элементов определенного типа;
- можно создавать пустой список;
- можно добавлять элемент в список;
- можно определять, пуст ли список;
- можно определять, полон ли список.
- можно посетить каждый элемент списка и выполнить над ним определенное действие.

Спроектируйте класс List для представления этого абстрактного типа. Вы должны подготовить заголовочный файл list.h с объявлением класса и файл list.cpp с реализацией его методов. Вы должны также написать короткую программу, которая будет использовать полученный класс.

Самостоятельная работа №3.

Описать класс «обыкновенная дробь», содержащий числитель и знаменатель (целые числа). В классе предусмотреть конструкторы с параметрами и без параметров. Предусмотреть метод info(), выводящий на экран информацию об объекте. Предусмотреть перегруженные операторы «==» и «!=», сравнивающие две дроби. Написать демонстрационную программу, создающую объекты и проверяющую работу перегруженных операторов.

Самостоятельная работа №4.

Описать класс «массив вещественных чисел». В классе предусмотреть методы удаления и добавления элементов (используя выделение динамической памяти). Предусмотреть метод info(), выводящий на экран информацию об объекте. Предусмотреть перегруженный оператор «/», выполняющий деление всех элементов массива на число. Написать

демонстрационную программу, создающую объекты и проверяющую работу перегруженного оператора.

Самостоятельная работа №5.

Описать класс «Автобус». Написать программу моделирования работы автобусного парка. Программа должна обеспечивать начальное формирование данных о всех автобусах в парке в виде списка (ввод с клавиатуры или из файла). Кроме того, программа должна обеспечивать имитацию выезда автобуса из парка: вводится номер автобуса, программа удаляет данные об этом автобусе из списка автобусов, находящихся в парке и записывает данные в список автобусов, находящихся на маршруте. Для представления необходимых списков использовать шаблонный класс `vector`.

Самостоятельная работа №6.

Напишите программу, запрашивающую у пользователя ввод двух целых чисел. Затем программа должна вычислить и выдать сумму всех целых чисел, лежащих между этими двумя целыми. Предполагается, что меньшее значение вводится первым. Например, если пользователь ввел 2 и 9, программа должна сообщить, что сумма всех целых чисел от 2 до 9 равна 44.

Самостоятельная работа №7.

Предположим, что вы продаете книгу по программированию на языке C++ для начинающих. Напишите программу, которая позволит ввести ежемесячные объемы продаж в течение года (в количестве книг, а не в деньгах). Программа должна использовать цикл, в котором выводится приглашение с названием месяца, применяя массив указателей на `char` (или массив объектов `string`, если вы предпочитаете его), инициализированный строками — названиями месяцев, и сохраняя введенные значения в массиве `int`. Затем программа должна найти сумму содержимого массива и выдать общий объем продаж за год.

Самостоятельная работа №8.

Определить шаблон функции, осуществляющей сортировку элементов массива по убыванию. Продемонстрировать работу шаблона на массивах объектов как встроенного типа, так и собственного класса. Для этого создать собственный класс и перегрузить в нём нужные функции-операции.

Самостоятельная работа №9.

Написать программу «Англо-русский и русско-английский словарь». Программа должна обеспечивать загрузку в контейнер типа `map` базы англо-русского перевода и формирование по нему другого контейнера типа `map` – базы русско-английского перевода. Реализовать функции вывода вариантов перевода заданного слова на альтернативный язык.

Самостоятельная работа №10.

Найти количество элементов матрицы равных нулю. Количество нулевых элементов по каждой строке матрицы рассчитывается отдельным потоком.

Самостоятельная работа №11.

Организовать программу, позволяющую имитировать работу банкоматов (работающих только на выдачу денег) отделения банка. На одном банкомате может обслуживаться одновременно только один клиент, у каждого банкомата есть резерв денежных средств и лимит на единовременную выдачу. У каждого клиента есть счёт с определенным балансом, клиент может снимать деньги в количестве не превышающим лимит банкомата и размер остатка на счету с банкомата несколько раз. После каждого снятия деньги списываются со счета клиента. Программа должна выводить на экран фиио или номер клиента и данные количестве снимаемых денег. Работа программы заканчивается тогда, когда заканчиваются деньги в автомате.

III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Разработка объектно-ориентированных приложений» представлено в разделе VIII и включает в себя:

- план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;
- характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;
- требования к представлению и оформлению результатов самостоятельной работы;
- критерии оценки выполнения самостоятельной работы.

План-график выполнения самостоятельной работы по дисциплине

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1	1 неделя семестра	Изучить основные понятия объектно-ориентированного подхода в программировании	5	Защита индивидуального задания
2	2 неделя семестра	Освоить основные элементы языка C++	5	Защита индивидуального задания
3	3-4 неделя семестра	Изучить функции в C++ и реализация простых структур	5	Защита индивидуального задания
4	5-6 неделя семестра	Изучить модульное программирование	5	Защита индивидуального задания
5	7-8 неделя семестра	Изучить работу с памятью – динамическое и статическое выделение памяти. Указатели.	5	Защита индивидуального задания
6	9-10 неделя семестра	Изучить пользовательские типы данных	5	Защита индивидуального задания
7	11-12 неделя семестра	Изучить классы, наследование, специальные методы классов	5	Защита индивидуального задания
8	13 неделя семестра	Изучить что такое перегрузка операторов. Виртуальные функции. Дружественные функции. Дружественные классы	5	Защита индивидуального задания
9	14 неделя семестра	Изучить шаблоны функций и классов. Обработка исключительных ситуаций	5	Защита индивидуального задания
10	15 неделя семестра	Изучить библиотеку STL, основные элементы. Итераторы и аллокаторы. Контейнерные классы	5	Защита индивидуального задания
11	16 неделя семестра	Изучить организация	4	Защита

		многопоточных программ. Мьютексы		индивидуального задания
12	17-18 неделя	Подготовка к экзамену	36	Экзамен
Итого:			90	

Рекомендации по самостоятельной работе студентов

Планирование и организация времени, отведенного на выполнение заданий самостоятельной работы.

Изучив график выполнения самостоятельных работ, следует правильно её организовать. Рекомендуется изучить структуру каждого задания, обратить внимание на график выполнения работ, отчетность по каждому заданию предоставляется в последнюю неделю согласно графику. Обратите внимание, что итоги самостоятельной работы влияют на окончательную оценку по итогам освоения учебной дисциплины.

Работа с литературой.

При выполнении ряда заданий требуется работать с литературой. Рекомендуется использовать различные возможности работы с литературой: фонды научной библиотеки ДВФУ (<http://www.dvfu.ru/library/>) и других ведущих вузов страны, а также доступных для использования научно-библиотечных систем.

В процессе выполнения самостоятельной работы, в том числе при написании эссе рекомендуется работать со следующими видами изданий:

а) Научные издания, предназначенные для научной работы и содержащие теоретические, экспериментальные сведения об исследованиях. Они могут публиковаться в форме: монографий, научных статей в журналах или в научных сборниках;

б) Учебная литература подразделяется на:

- учебные издания (учебники, учебные пособия, тексты лекций), в которых содержится наиболее полное системное изложение дисциплины или какого-то ее раздела;

- справочники, словари и энциклопедии – издания, содержащие краткие сведения научного или прикладного характера, не предназначенные для сплошного чтения. Их цель – возможность быстрого получения самых общих представлений о предмете.

Существуют два метода работы над источниками:

– сплошное чтение обязательно при изучении учебника, глав монографии или статьи, то есть того, что имеет учебное значение. Как

правило, здесь требуется повторное чтение, для того чтобы понять написанное. Старайтесь при сплошном чтении не пропускать комментарии, сноски, справочные материалы, так как они предназначены для пояснений и помощи. Анализируйте рисунки (карты, диаграммы, графики), старайтесь понять, какие тенденции и закономерности они отражают;

– метод выборочного чтения дополняет сплошное чтение; он применяется для поисков дополнительных, уточняющих необходимых сведений в словарях, энциклопедиях, иных справочных изданиях. Этот метод крайне важен для повторения изученного и его закрепления, особенно при подготовке к зачету.

Для того чтобы каждый метод принес наибольший эффект, необходимо фиксировать все важные моменты, связанные с интересующей Вас темой.

Конспект – это способ самостоятельно изложить содержание книги или статьи в логической последовательности. Конспектируя какой-либо источник, надо стремиться к тому, чтобы немногими словами сказать о многом. В тексте конспекта желательно поместить не только выводы или положения, но и их аргументированные доказательства (факты, цифры, цитаты).

Писать конспект можно и по мере изучения произведения, например, если прорабатывается монография или несколько журнальных статей.

Методические рекомендации по выполнению заданий для самостоятельной работы и критерии оценки.

Собеседование (устный опрос) позволяет оценить знания и кругозор студента, умение логически построить ответ, владение монологической речью и иные коммуникативные навыки.

Опрос – важнейшее средство развития мышления и речи. Обучающая функция опроса состоит в выявлении деталей, которые по каким-то причинам оказались недостаточно осмысленными в ходе учебных занятий и при подготовке задания по самостоятельной работе.

Критерии оценки. Используется зачетная система. Во время опроса допускается не более 1-й ошибки или неточности по названию периода, его времени и длительности.

Оценка	Требования
«зачтено»	Студент владеет навыками самостоятельной работы по теме исследования, реферировать литературные источники; методами анализа теоретических и/или практических аспектов изучаемой области. Эссе характеризуется смысловой цельностью, связностью и

	последовательностью изложения. Студент умеет обобщать фактический материал, делать самостоятельные выводы. Работа соответствует требованиям и выполнена в установленные сроки.
«не зачтено»	Не раскрыта структура и теоретическая составляющая темы. Студент не умеет обобщать фактический материал, делать самостоятельные выводы, не владеет навыком реферировать литературные источники. Эссе не выполнено.

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые модули/ разделы / темы дисциплины	Код индикатора достижения компетенции	Результаты обучения	Оценочные средства – наименование		
				текущий контроль	промежуточная аттестация	
1	<p>Объекты и классы С++.</p> <p>Работа с классами в С++.</p> <p>Классы и динамическое выделение памяти.</p> <p>Наследование.</p> <p>Шаблоны функций и классов.</p> <p>Обработка исключительных ситуаций.</p> <p>Организация многопоточных программ.</p> <p>Мьютексы.</p>	<p>ПК-6. Способность использовать навыки моделирования, анализа и формальных методов конструирования программного обеспечения</p>	демонстрирует знание современных приемов работы с инструментальными средствами, поддерживающим и создание программных продуктов и программных комплексов, их сопровождения и администрирования	<p>УО1 собеседование</p> <p>ПР2 контрольное задание</p> <p>ПР-6 лабораторная работа 1</p>	<p>Экзамен, вопросы к экзамену № 1-15</p>	
			использует современные инструментальные средства в практической деятельности			<p>УО1 собеседование</p> <p>ПР2 контрольное задание</p> <p>ПР-6 лабораторная работа 1</p>
			применяет современные инструментальные средства при разработке программных продуктов			<p>УО1 собеседование</p> <p>ПР2 контрольное задание</p> <p>ПР-6 лабораторная работа 1</p>
			ПК-8. Способен использовать основные	<p>демонстрирует знание основных концептуальных</p> <p>УО1 собеседование</p> <p>ПР2</p>		

		концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений	положений функционального, логического, объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках этих направлений	контрольное задание ПР-6 лабораторная работа 2	
			использует основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений программирования	УО1 собеседование ПР2 контрольное задание ПР-6 лабораторная работа 2	
			применяет основные концептуальные положения функционального, логического, объектно-ориентированного и визуального направлений при создании программных систем	УО1 собеседование ПР2 контрольное задание ПР-6 лабораторная работа 2	

Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в разделе IX.

У. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература

1. Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс]/ Мейер Б.— Электрон. текстовые данные.— Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019.— 285 с.— Режим доступа: <http://www.iprbookshop.ru/79706.html>.
2. Новиков П.В. Объектно-ориентированное программирование [Электронный ресурс]: учебно-методическое пособие к лабораторным работам/ Новиков П.В.— Электрон. текстовые данные.— Саратов: Вузовское образование, 2017.— 124 с.— Режим доступа: <http://www.iprbookshop.ru/64650.html>.
3. Хорев, П. Б. Объектно-ориентированное программирование с примерами на С#: Учебное пособие / Хорев П.Б. - Москва Форум, НИЦ ИНФРА-М, 2016. - 200 с. (Высшее образование: Бакалавриат) ISBN 978-5-00091-144-0. - Текст: электронный. - URL: <https://znanium.com/catalog/product/529350>
4. Павловская, Т. А. С/С++. Процедурное и объектно-ориентированное программирование: учебник для вузов / Татьяна Павловская. - Санкт-Петербург: Питер, 2015. - 495 с. - Режим доступа: <http://lib.dvfu.ru:8080/lib/item?id=chamo:780731&theme=FEFU> Язык программирования С++. Специальное издание: пер. с англ. / Бьерн Страуструп. - М.: Вильямс, 2011. — 1135с. <https://lib.dvfu.ru:8443/lib/item?id=chamo:418460&theme=FEFU>
5. С/С++. Программирование на языке высокого уровня: учебник для вузов для магистров и бакалавров / Т. А. Павловская. СПб.: Питер, 2011.-460 с. <https://lib.dvfu.ru:8443/lib/item?id=chamo:307692&theme=FEFU>
6. Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; под ред. Л.Г. Гагариной. — Москва: ИД «ФОРУМ» : ИНФРА-М, 2019. — 512 с. — URL: <https://znanium.com/catalog/product/1000008>
7. Объектно-ориентированное программирование в С++: лекции и упражнения [Электронный ресурс]: Учебное пособие для вузов / Ашарина И.В. - М.: Горячая линия - Телеком, 2012. - <http://www.studentlibrary.ru/book/ISBN9785991270014.html>

Дополнительная литература

(печатные и электронные издания)

1. C/C++. Структурное и объектно-ориентированное программирование: практикум / Т.А. Павловская, Ю.А. Щупак.- СПб.: Питер, 2011.- 352 с.
<http://lib.dvfu.ru:8080/lib/item?id=chamo:418970&theme=FEFU>
2. Павловская Т.А. C/C++. Программирование на языке высокого уровня- СПб.: Питер, 2021. - 461 с.
3. Царев, Р. Ю. Программирование на языке Си [Электронный ресурс]: учеб. пособие / Р. Ю. Царев. – Красноярск: Сиб. федер. ун-т, 2014. – 108 с.
<http://znanium.com/catalog.php?bookinfo=510946>
4. Комлев, Н. Ю. Объектно-ориентированное программирование. Хорошая книга для хороших Людей / Комлев Н.Ю. - Москва: СОЛОН-Пр., 2015. - 298 с.:ISBN 978-5-91359-138-8. - Текст: электронный. - URL:
<https://znanium.com/catalog/product/884394>
5. Керниган Б., Ритчи Д. Язык программирования С. Пер. с англ. — М.: Издательский дом "Вильямс", 2012. — 289 с.
<http://lib.dvfu.ru:8080/lib/item?id=chamo:666721&theme=FEFU>
6. Программирование на языке C++ в среде Qt Creator/ Е. Р. Алексеев, Г. Г. Злобин, Д. А. Костюк, О. В. Чеснокова, А. С. Чмыхало - М. : ALT Linux, 2015. — 448 с.
7. Объектно-ориентированное программирование в C++ / Р. Лафоре ; [Пер. с англ. А. Кузнецов и др.]. - 4. изд. - М. [и др.] : Питер, 2018 (ГПП Печ. Двор). - 923 с

Профессиональные базы данных и информационные справочные системы

1. База данных Scopus <http://www.scopus.com/home.url>
2. База данных Web of Science <http://apps.webofknowledge.com/>
3. Электронная библиотека диссертаций Российской государственной библиотеки <http://diss.rsl.ru/>
4. Электронные базы данных EBSCO <http://search.ebscohost.com/>

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. <https://code-live.ru/tag/cpp-manual/> Портал о программировании
2. <http://cppstudio.com/cat/274/> Язык программирования C++
3. <http://www.c-cpp.ru/> Программирование на C и C++. Онлайн справочник программиста

4. <http://www.proklondike.com/books/cpp.html> Программирование
5. <http://purecodecpp.com/> основы программирования на С++ для начинающих
6. <http://progbook.ru/c/> Книги по С++
7. <http://www.programmersclub.ru/main/> Клуб программистов
8. <http://info-comp.ru/> Информационный портал. Все о компьютере и программировании для начинающих

Перечень информационных технологий и программного обеспечения

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используется следующее программное обеспечение:

1. Microsoft Office (Access, Excel, PowerPoint, Word и т. д).
2. Open Office.
3. Skype.

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используются следующие информационно-справочные системы:

1. Научная электронная библиотека eLIBRARY.
2. Электронно-библиотечная система издательства «Лань».
3. Электронная библиотека "Консультант студента".
4. Электронно-библиотечная система IPRbooks.
5. Информационная система "ЕДИНОЕ ОКНО доступа к образовательным ресурсам".
6. Доступ к электронному заказу книг в библиотеке ДВФУ, доступ к нормативным документам ДВФУ, расписанию, рассылке писем.

Лабораторные занятия проводятся в специализированном компьютерном классе.

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Дисциплина «Разработка объектно-ориентированных приложений» изучается в следующих организационных формах: лабораторное занятие; самостоятельное изучение теоретического материала; индивидуальные и групповые консультации.

Контроль за выполнением самостоятельной работы студента производится в виде контроля каждого этапа работы, отраженного в документации и защиты проекта.

Студент должен планировать график самостоятельной работы по дисциплине и придерживаться его.

Лабораторные работы

В результате выполнения лабораторных работ студент должен научиться формализовать поставленную задачу, составлять и оформлять программы на языке программирования C++, тестировать и отлаживать программы, использовать для разработки программ современные интегрированные среды разработки программ.

При выполнении лабораторной работы необходимо следовать методическим рекомендациям по ее выполнению. Результатом лабораторной работы является программа и отчет (документация), которые демонстрируются преподавателю в конце работы. Студент должен уметь отвечать на вопросы преподавателя, поясняя процесс создания программ и документа и выполнения работы.

Индивидуальные задания необходимо выполнять согласно рассмотренной технологии:

1. изучить словесную постановку задачи;
2. сформулировать математическую постановку задачи;
3. выбрать метод решения задачи, если это необходимо;
4. разработать схему алгоритма и спроектировать требуемые классы;
5. записать разработанный алгоритм на языке C++;
6. разработать контрольные тесты программы;
7. отладить программу;
8. написать отчет.

Содержание отчета

1. Титульный лист.
2. Неформальная постановка задачи (НПЗ).
3. Формальная постановка задачи (ФПЗ).
4. Проекты классов
5. Текст программы с комментариями.
6. Тестовые условия.

Работа с литературой

В процессе подготовки к лабораторным работам студентам необходимо обратить особое внимание на поиск и на самостоятельное изучение рекомендованной учебно-методической (а также научной и популярной) литературы. Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами

периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов свое отношение к конкретной проблеме. Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем по каждой теме практического занятия, что позволяет студентам проявить свою индивидуальность в рамках выполнения индивидуального проекта, выявить широкий спектр мнений по изучаемой проблеме.

Самостоятельная работа студента

Основными формами самостоятельной работы студента являются:

- подготовка к лабораторным занятиям, зачету,
- изучение обязательной и дополнительной литературы,
- поиск информации по изучаемым темам в периодических изданиях и Интернете,
- изучение в рамках программы курса тем, не выносимых на лабораторные работы,
- оформление отчетов по лабораторным работам.

Контроль за выполнением работы студента производится в виде контроля каждого этапа работы.

Студент должен планировать график самостоятельной работы по дисциплине и придерживаться его.

Критерии оценки отчетов по самостоятельной работе

– 100-86 баллов выставляется, если содержание и составляющие части соответствуют выданному заданию. Продемонстрировано владение навыками подготовки документа по теме. Фактических ошибок, связанных с пониманием структуры и содержания задания нет.

– 85-76 - баллов выставляется, если при выполнении задания допущено не более одной ошибки. Продемонстрировано владение навыками подготовки документа по теме. Фактических ошибок, связанных с пониманием структуры и содержания задания нет.

– 75-61 балл выставляется, если при выполнении задания допущено не более двух ошибок. Продемонстрировано навыки подготовки документа по теме. Допущено не более 2 ошибок, связанных с пониманием структуры и содержания задания.

60-50 баллов - если структура и содержание задания не соответствуют требуемым

Шкала оценивания

Менее 60 баллов	Не зачтено	неудовлетворительно
От 61 до 75 баллов	зачтено	удовлетворительно
От 76 до 85 баллов	зачтено	хорошо
От 86 до 100 баллов	зачтено	отлично

Рекомендации по работе с литературой

Для более эффективного освоения и усвоения материала рекомендуется ознакомиться с теоретическим материалом по той или иной теме до проведения лабораторного занятия. Всю учебную литературу желательно изучать «под конспект».

Цель написания конспекта по дисциплине – сформировать навыки по поиску, отбору, анализу и формулированию учебного материала.

Работу с теоретическим материалом по теме можно проводить по следующей схеме:

- название темы;
- цели и задачи изучения темы;
- основные вопросы темы;
- характеристика основных понятий и определений, необходимых для усвоения данной темы;
- краткие выводы, ориентирующие на определенную совокупность сведений, основных идей, ключевых положений, систему доказательств, которые необходимо усвоить.

При работе над конспектом обязательно выявляются и отмечаются трудные для самостоятельного изучения вопросы, с которыми уместно обратиться к преподавателю при посещении консультаций, либо в индивидуальном порядке.

Подготовка к лабораторным работам

Подготовку к каждой лабораторной работе каждый студент должен начать с изучения теоретического материала и ознакомления с планом, который отражает содержание предложенной темы. Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса. Результат такой работы должен проявиться в способности студента свободно ответить на теоретические вопросы по теме задания, правильном выполнении лабораторной работы.

В процессе выполнения лабораторной работы студент должен создать

требуемый документ (отчет) с помощью предлагаемого программного средства и выполнить требуемые в задании операции. Задание по лабораторной работе содержит методические указания по подготовке документа, который должен быть получен в результате выполнения работы. При подготовке к лабораторной работе следует их внимательно прочесть.

При формировании индивидуальных заданий по самостоятельной работе используется материал из следующих источников: Программирование: метод. указания/ сост. Л.И.Прудникова - Владивосток: Изд-во ДВФУ, 2012. - 24 с., C/C++. Структурное и объектно-ориентированное программирование: практикум / Т.А. Павловская, Ю.А. Щупак.- СПб.: Питер, 2011.- 352 с., Программирование на языке C++ в среде Qt Creator/ Е. Р. Алексеев, Г. Г. Злобин, Д. А. Костюк, О. В. Чеснокова, А. С. Чмыхало - М. : ALT Linux, 2015. — 448 с.

Содержание отчета по лабораторной работе

1. Титульный лист.
2. Постановка задачи.
3. Текст программы с комментариями.
4. Тестовые условия.

Критерии оценки отчетов по лабораторным работам (проектов)

– 100-86 баллов выставляется, если содержание и составляющие части соответствуют выданному заданию. Продемонстрировано владение навыками подготовки документа по теме. Фактических ошибок, связанных с пониманием структуры и содержания задания нет.

– 85-76 - баллов выставляется, если при выполнении задания допущено не более одной ошибки. Продемонстрировано владение навыками подготовки документа по теме. Фактических ошибок, связанных с пониманием структуры и содержания задания нет.

– 75-61 балл выставляется, если при выполнении задания допущено не более двух ошибок. Продемонстрировано навыки подготовки документа по теме. Допущено не более 2 ошибок, связанных с пониманием структуры и содержания задания.

60-50 баллов - если структура и содержание задания не соответствуют требуемым

Шкала оценивания

Менее 60 баллов	незачтено	неудовлетворительно
От 61 до 75 баллов	зачтено	удовлетворительно

От 76 до 85 баллов	зачтено	хорошо
От 86 до 100 баллов	зачтено	отлично

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Перечень материально-технического и программного обеспечения дисциплины приведен в таблице.

Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы	Перечень лицензионного программного обеспечения. Реквизиты подтверждающего документа
690922, Приморский край, г. Владивосток, остров Русский, полуостров Саперный, поселок Аякс, 10, Корпус D, аудитория 733/733а Учебная аудитория для проведения занятий лекционного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	Помещение укомплектовано специализированной учебной мебелью (посадочных мест – 13) Оборудование: ЖК-панель 47", Full HD, LG M4716 CСВА – 1 шт. Доска аудиторная, Моноблок Lenovo C360G- i34164G500UDK с лицензионными программами Microsoft Office 2013(13 шт.) и аудиовизуальными средствами проектор Panasonic DLPPjectorPT- D2110XE	1С Предприятия8 (8.2), 7-Zip, ABBYY Lingvo12, Alice 3, Anaconda3, Autodesk, CodeBlocks, CorelDRAW X7, Dia ,Directum4.8 ,DosBox-0.74, Farmanager, Firebird 2.5, FlameRobin, Foxit Reader, Free Pascal, Geany, Ghostscript, Git version 1.9.5-preview20141217, Greenfoot, gsview, Inscapе0.91, Java 8 Update 60 (64-bit), Java development Kit, Kaspersky, Lazarus, LibreOffice4.4, MatLab R2017b, Maxima 5.37.2, Microsoft Expression, Microsoft Office 2013, Microsoft Silverlight, Microsoft Silverlight 5SDK-русский, Microsoft System Center, Microsoft Visual Studio 2012, MikTeX2.9, MySQL, NetBeans, Notepad++, Oracle VM VirtualBox, PascalABC.NET, PostgreSQL 9.4, PTC Mathcad, Putty, PyQt GPL v5.4.1 for Pythonv 3.4, Python2.7(3.4,3.6), QGIS Brighton, RStudio, SAM CoDeC Pack, SharePoint, Strawberry Perl, Tecnomatix, TeXnicCenter, TortoiseSVN, Unity2017.3.1f1, Veusz, Vim8.1, Visual Paradigm CE, Visual Studio 2013, Microsoft Visual C++, Windows Kits, Windows Phone SDK8.1, Xilinx Design Tools, Acrobat Reader DC, Adobe Bridge CS3, Adobe Device Central, CS3,Adobe Extend Script Toolkit 2,Adobe Photoshope CS3,DVD-студия Windows, Google Chrome, Internet Explorer, ITMOproctor, Mozilla Firefox 39.0 (x86 ru), Visual Studio Installer, Windows Media Center, WinSCP, Xming 6.9.0.31, ESET Endpoint Security, Firebird 2.5.3.26780 (x64), 7-Zip 9.20 (x64 edition), Adobe Flash Player, Adobe Shockwave Player, ESET Endpoint Antivirus, FOG Service, Hao Zip
690922, Приморский край, г.Владивосток, остров Русский, полуостров Саперный, поселок Аякс, 10, Корпус D, аудитория 733/733а. Аудитория для самостоятельной работы	Помещение укомплектовано специализированной учебной мебелью (посадочных мест – 13) Оборудование: ЖК-панель 47", Full HD, LG M4716 CСВА – 1 шт. Доска аудиторная, Моноблок Lenovo C360G- i34164G500UDK с лицензионными	1С Предприятия8 (8.2), 7-Zip, ABBYY Lingvo12, Alice 3, Anaconda3, Autodesk, CodeBlocks, CorelDRAW X7, Dia ,Directum4.8 ,DosBox-0.74, Farmanager, Firebird 2.5, FlameRobin, Foxit Reader, Free Pascal, Geany, Ghostscript, Git version 1.9.5-preview20141217, Greenfoot, gsview, Inscapе0.91, Java 8 Update 60 (64-bit), Java development Kit, Kaspersky, Lazarus, LibreOffice4.4, MatLab R2017b, Maxima 5.37.2, Microsoft Expression, Microsoft Office 2013, Microsoft Silverlight, Microsoft Silverlight 5SDK-русский, Microsoft System Center, Microsoft Visual Studio 2012, MikTeX2.9, MySQL, NetBeans, Notepad++, Oracle VM VirtualBox, PascalABC.NET, PostgreSQL 9.4, PTC Mathcad, Putty, PyQt GPL v5.4.1 for Pythonv 3.4, Python2.7(3.4,3.6), QGIS Brighton, RStudio, SAM CoDeC Pack, SharePoint, Strawberry Perl, Tecnomatix, TeXnicCenter, TortoiseSVN, Unity2017.3.1f1,

	<p>программами Microsoft Office 2013(13 шт.) и аудиовизуальными средствами проектор Panasonic DLPProjectorPT-D2110XE</p>	<p>Veusz, Vim8.1, Visual Paradigm CE, Visual Studio 2013, Microsoft Visual C++, Windows Kits, Windows Phone SDK8.1, Xilinx Design Tools, Acrobat Reader DC, Adobe Bridge CS3, Adobe Device Central, CS3, Adobe Extend Script Toolkit 2, Adobe Photosope CS3, DVD-студия Windows, Google Chrome, Internet Explorer, ITMOproctor, Mozilla Firefox 39.0 (x86 ru), Visual Studio Installer, Windows Media Center, WinSCP, Xming 6.9.0.31, ESET Endpoint Security, Firebird 2.5.3.26780 (x64), 7-Zip 9.20 (x64 edition), Adobe Flash Player, Adobe Shockwave Player, ESET Endpoint Antivirus, FOG Service, Hao Zip</p>
--	--	---

Для проведения учебных занятий по дисциплине, а также для организации самостоятельной работы студентам доступно следующее лабораторное оборудование и специализированные кабинеты, соответствующие действующим санитарным и противопожарным нормам, а также требованиям техники безопасности при проведении учебных и научно-производственных работ.

В целях обеспечения специальных условий обучения инвалидов и лиц с ограниченными возможностями здоровья в ДВФУ все здания оборудованы пандусами, лифтами, подъемниками, специализированными местами, оснащенными туалетными комнатами, табличками информационно-навигационной поддержки.

VIII. ФОНДЫ ОЦЕНОЧНЫХ СРЕДСТВ

Для дисциплины «Объектно-ориентированное программирование» используются следующие оценочные средства:

Устный опрос:

1. Собеседование (УО-1)
2. Презентация (УО-3)

Письменные работы:

1. Лабораторная работа (ПР-6)
2. Контрольное задание (ПР-2)

Устный опрос

Устный опрос позволяет оценить знания и кругозор студента, умение логически построить ответ, владение монологической речью и иные коммуникативные навыки.

Обучающая функция состоит в выявлении деталей, которые по каким-то причинам оказались недостаточно осмысленными в ходе учебных занятий и при подготовке к зачёту.

Устный опрос (УО-1) – средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п.

Презентация (УО-3) – продукт самостоятельной работы обучающегося, представляющий собой публичное выступление по представлению полученных результатов решения определенной учебно-практической, учебно-исследовательской или научной темы.

Письменные работы

Письменный ответ приучает к точности, лаконичности, связности изложения мысли. Письменная проверка используется во всех видах контроля и осуществляется как в аудиторной, так и во внеаудиторной работе.

Лабораторная работа (ПР-6) – средство для закрепления и практического освоения материала по определенному разделу.

Контрольное задание (ПР-2) – средство для материала по определенному разделу.

Методические рекомендации, определяющие процедуры оценивания результатов освоения дисциплины

Оценочные средства для промежуточной аттестации

Промежуточная аттестация студентов по дисциплине «Разработка объектно-ориентированных приложений» проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной. Форма отчётности по дисциплине – экзамен (четвертый семестр). Экзамен по дисциплине включает ответы на 2 вопроса. Один из вопросов носит общий характер. Он направлен на раскрытие студентом знаний по «сквозным» вопросам и проблемам. Второй вопрос включает практическое задание.

Методические указания по сдаче зачета/экзамена

Экзамен принимается ведущим преподавателем. При большом количестве групп у одного преподавателя или при большой численности потока по распоряжению заведующего кафедрой (заместителя директора по учебной и воспитательной работе) допускается привлечение в помощь ведущему преподавателю других преподавателей. В первую очередь привлекаются преподаватели, которые проводили лабораторные занятия по дисциплине в группах.

В исключительных случаях, по согласованию с заместителем директора Школы по учебной и воспитательной работе, заведующий кафедрой имеет право принять экзамен в отсутствие ведущего преподавателя.

Форма проведения экзамена утверждается на заседании кафедры по согласованию с руководителем в соответствии с рабочей программой дисциплины.

Во время проведения зачета студенты могут пользоваться рабочей программой дисциплины, а также с разрешения преподавателя, справочной литературой и другими пособиями (учебниками, учебными пособиями, рекомендованной литературой и т.п.).

Время, предоставляемое студенту на прохождение теста на экзамене, должно составлять не более 60 минут.

Присутствие на зачете посторонних лиц (кроме лиц, осуществляющих проверку) без разрешения соответствующих лиц (ректора либо проректора по учебной и воспитательной работе, директора Школы, руководителя ОПОП или заведующего кафедрой), не допускается. Инвалиды и лица с ограниченными возможностями здоровья, не имеющие возможности самостоятельного передвижения, допускаются зачет с сопровождающими.

Оценочные средства для промежуточной аттестации

Вопросы к экзамену

1. Модель памяти и структура программы. Классы памяти. Ссылки.
2. Средства абстракции C++. Структура класса. Статические члены и их инициализация
3. Средства инкапсуляции C++. Инкапсуляция и наследование. Друзья.
4. Модульность, раздельная компиляция, пространства имен, using директива.
5. Представление иерархических отношений. Наследование.
6. Представление иерархических отношений. Агрегация. Зависимость по времени жизни.
7. Правила преобразования типов в C++. Параметрический и виртуальный полиморфизм.
8. C++: средства реализации состояния объектов; реализация поведения.
9. Перегрузка операторов.

10. Жизненный цикл объекта. Инициализация массивов. Конструкторы и деструкторы. Порядок вызова конструкторов и деструкторов при наследовании.
11. Исключения в C++. Обработка исключений. Умные указатели.
12. Шаблоны классов и шаблоны функций. Специализация.
13. Основы STL. Структура и назначение. Контейнеры.
14. Основы STL. Аллокаторы и итераторы.
15. Общие сведения об организации многопоточного программирования, потоки и мьютексы.

Критерии выставления оценки студенту на экзамене

К экзамену допускаются обучающиеся, выполнившие программу обучения по дисциплине, прошедшие все этапы текущей аттестации.

Баллы	Оценка зачета/ экзамена	Требования к сформированным компетенциям
86-100	«зачтено»/ «отлично»	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.
76-85	«зачтено»/ «хорошо»	Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения.
61-75	«зачтено»/ «удовлетворительно»	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ.

0-60	«не зачтено»/ «неудовлетворительно»	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.
------	--	---

Оценочные средства для текущей аттестации

Текущая аттестация студентов по дисциплине проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

Текущая аттестация проводится в форме контрольных мероприятий (устного опроса, презентации, лабораторных работ) по оцениванию фактических результатов обучения студентов и осуществляется ведущим преподавателем.

Объектами оценивания выступают:

- учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость всех видов занятий по аттестуемой дисциплине);
- степень усвоения теоретических знаний;
- уровень овладения практическими умениями и навыками по всем видам учебной работы;
- результаты самостоятельной работы.

Составляется календарный план контрольных мероприятий по дисциплине. Оценка посещаемости, активности обучающихся на занятиях, своевременность выполнения различных видов заданий ведётся на основе журнала, который ведёт преподаватель в течение учебного семестра.

Вопросы для устного опроса

Раздел 1. Основные принципы объектно-ориентированного программирования

1. Эволюция методологий программирования. Парадигмы программирования.
2. Основные принципы объектного подхода. Абстрагирование.
3. Основные принципы объектного подхода. Инкапсуляция.
4. Основные принципы объектного подхода. Модульность.
5. Основные принципы объектного подхода. Иерархия.
6. Основные принципы объектного подхода. Типизация.

7. Объект с точки зрения ООП. Состояние. Поведение.
 8. Объект с точки зрения ООП. Идентичность и жизненный цикл объектов.
 9. Объект с точки зрения ООП. Взаимоотношения между объектами.
 10. Классы. Природа классов. Мета модель. Инстанцирование.
 11. Классы. Структура класса. Абстрактные классы и интерфейсы.
 12. Классы. Принцип подстановки Лисковской. Принцип разделения интерфейсов.
 13. Классы. Средства UML для построения диаграмм классов.
 14. Классы. Отношения между классами. Ассоциация и агрегация.
- Классы. Иерархии классов. Зависимость.

Раздел 2. Основы программирования на языке C++

1. Модель памяти и структура программы. Классы памяти. Ссылки.
2. Средства абстракции C++. Структура класса. Статические члены и их инициализация
3. Средства инкапсуляции C++. Инкапсуляция и наследование. Друзья.
4. Модульность, раздельная компиляция, пространства имен, using директива.
5. Представление иерархических отношений. Наследование.
6. Представление иерархических отношений. Агрегация. Зависимость по времени жизни.
7. Правила преобразования типов в C++. Параметрический и виртуальный полиморфизм.
8. C++: средства реализации состояния объектов; реализация поведения.
9. Перегрузка операторов.
10. Жизненный цикл объекта. Инициализация массивов. Конструкторы и деструкторы. Порядок вызова конструкторов и деструкторов при наследовании.
11. Исключения в C++. Обработка исключений. Умные указатели.
12. Шаблоны классов и шаблоны функций. Специализация.
13. Основы STL. Структура и назначение. Контейнеры.
14. Основы STL. Аллокаторы и итераторы.
15. Общие сведения об организации многопоточного программирования, потоки и мьютексы.

Критерии оценки устного ответа

– **100-85 баллов** - если ответ показывает прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой

раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа; умение приводить примеры современных проблем изучаемой области.

– **85-76 баллов** - ответ, обнаруживающий прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа. Однако допускаются одна - две неточности в ответе.

– **75-61 балл** - оценивается ответ, свидетельствующий в основном о знании процессов изучаемой предметной области, отличающийся недостаточной глубиной и полнотой раскрытия темы; знанием основных вопросов теории; слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры; недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа; неумение привести пример развития ситуации, провести связь с другими аспектами изучаемой области.

– **60-50 баллов** - ответ, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы; незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов; неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Допускаются серьезные ошибки в содержании ответа; незнание современной проблематики изучаемой области.

Шкала оценивания

Менее 60 баллов	незачтено	неудовлетворительно
От 61 до 75 баллов	зачтено	удовлетворительно
От 76 до 85 баллов	зачтено	хорошо
От 86 до 100 баллов	зачтено	отлично

Тематика презентаций

1. Стандарты языка C++
2. Описание структур в различных стандартах языка C++
3. Лямбда-функции – назначение, синтаксис, специфика применения
4. Синхронизация потоков данных в C++.

Критерии оценки презентации

Оценка	2 балла (неудовлетворительно)	3 балла (удовлетворительно)	4 балла (хорошо)	5 баллов (отлично)
Критерии	Содержание критериев			
Раскрытие Проблемы	Проблема не раскрыта. Отсутствуют выводы	Проблема раскрыта не полностью. Выводы не сделаны и/или выводы не обоснованы	Проблема раскрыта. Проведен анализ проблемы без привлечения дополнительной литературы. Не все выводы сделаны и/или обоснованы	Проблема раскрыта полностью. Проведен анализ проблемы с привлечением дополнительной литературы. Выводы обоснованы
Представление	Представляемая информация логически не связана. Не использованы профессиональные термины. Отсутствует иллюстративный материал в виде блок-диаграмм, профилей	Представляемая информация не систематизирована и/или не последовательна. Использовано 1-2 профессиональных термина. Иллюстративный материал в виде блок-диаграмм, профилей заимствован	Представляемая информация не систематизирована и последовательна. Использовано более 2 профессиональных терминов. Представлен иллюстративный материал в виде блок-диаграмм, профилей	Представляемая информация систематизирована, последовательна и логически связана. Использовано более 5 профессиональных терминов. Представлен самостоятельно сделанный иллюстративный материал в виде блок-диаграмм, профилей
Оформление	Не использованы технологии Power Point. Больше 4 ошибок в представляемой информации	Использованы технологии Power Point частично. 3-4 ошибки в представляемой информации	Использованы технологии Power Point. Не более 2 ошибок в представляемой информации	Широко использованы технологии (Power Point и др.). Отсутствуют ошибки в представляемой информации

Ответы на вопросы	Нет ответов на вопросы	Только ответы на элементарные вопросы	Ответы на вопросы полные и/или частично полные	Ответы на вопросы полные, с приведением примеров и/или пояснений
--------------------------	------------------------	---------------------------------------	--	--

Тематика лабораторных работ

1. Объекты и классы C++.
2. Работа с классами в C++.
3. Классы и динамическое выделение памяти.
4. Наследование в C++.
5. Шаблоны функций и классов. Обработка исключительных ситуаций
6. Организация многопоточных программ. Мьютексы.

Критерии оценки программы по лабораторным работам (проектов)

– 100-86 баллов выставляется, если содержание и составляющие части соответствуют выданному заданию. Продемонстрировано владение навыками разработки, тестирования программ на языке программирования C++. Программа правильно работает на всех наборах входных данных. Текст программы содержит комментарии.

– 85-76 - баллов выставляется, если при выполнении задания допущено не более одной ошибки. Продемонстрировано владение навыками разработки программ на языке программирования C++. Программа правильно работает не на всех наборах входных данных (90%). Текст программы содержит комментарии.

– 75-61 балл выставляется, если при выполнении задания допущено не более двух ошибок. Продемонстрировано знание методов разработки программ на языке программирования C++. Программа правильно работает не на всех наборах входных данных (70%). В тексте программы комментарии отсутствуют.

60-50 баллов - если структура и содержание задания не соответствуют требуемым

Шкала оценивания

Менее 60 баллов	незачтено	неудовлетворительно
От 61 до 75 баллов	зачтено	удовлетворительно
От 76 до 85 баллов	зачтено	хорошо
От 86 до 100 баллов	зачтено	отлично

Примерные задания к лабораторным работам

ЛАБОРАТОРНАЯ РАБОТА 1. ОБЪЕКТЫ И КЛАССЫ C++

Получение базовых знаний в области объектно-ориентированного программирования, изучение основных понятий и парадигм: определение класса для заданной задачи, программирование методов класса, типы конструкторов, прототипы функций.

Задание 1. Описать класс в соответствии с предметной областью «Книга». В классе предусмотреть несколько свойств и методов. Предусмотреть метод info(), выводящий на экран информацию об объекте. Предусмотреть конструкторы, инициализирующие свойства объектов переданными данными и значениями по умолчанию. Программу снабдить комментариями. Написать демонстрационную программу, создающую 4-5 объектов и выводящую на экран информацию о них.

Задание 2. Пусть имеется определение следующего простого класса:

```
#pragma once
using namespace std;
class cargo
{
public:
    char modelNumber[20];
    char color[10];
    double weight;
    double price;

    cargo();
    cargo(char* modelNumber, char* color, double weight, double
price);
    void info();
    void print(char* page);
};
```

Напишите программу, которая дополнит реализацию за счет предоставления кода для пока еще не определенных методов. В программе, использующей класс, должны также присутствовать вызовы всех возможных конструкторов (без аргументов, с аргументами) и всех методов.

Ниже приведен пример применения этих конструкторов и методов:

```
cargo prn1, prn2((char*)"HP2000 ",(char*)"GREY ", 2.1, 5200);
prn1.info();
prn2.info();

prn2.print((char*)"Hello!!!");

return 0;
```

Задание 3. Пусть имеется следующее объявление структуры:

```
struct customer {
    char fullname[35]; double payment;
};
```

Напишите программу, которая будет добавлять структуры заказчиков в стек и удалять их из стека, представленного объявлением класса `Stack`. Всякий раз, когда заказчик удаляется из стека, его зарплата должна добавляться к промежуточной сумме и по этой сумме выдаваться отчет. На заметку: вы должны иметь возможность пользоваться классом `Stack` без изменений; просто поменяйте объявление `typedef`, чтобы `Item` был типом `customer` вместо `unsigned long`.

Задание 4. Пусть имеется следующее объявление класса:

```
class Move {
private:
    double x;
    double y;
public:
    Move (double a = 0, double b = 0) ; // устанавливает x, y в a, b
    showmove() const; // отображает текущие значения x, y
    Move add (const Move & m) const;
    // Эта функция добавляет x из m к x вызывающего объекта,
    // чтобы получить новое значение x,
    reset (double a = 0, double b = 0) ; // сбрасывает x, y в a, b
};
```

Создайте определения функций-членов и напишите программу, которая использует этот класс.

ЛАБОРАТОРНАЯ РАБОТА 2. РАБОТА С КЛАССАМИ В C++

Получение практических навыков в области перегрузки операторов.

Разработка классов для работы с данными разных типов: стек, список, вектор, комплексные числа и т.д. Перегрузка операций. Дружественные функции.

Задание 1. Разработайте класс `Vector` (для хранения вектора в прямоугольной системе координат) так, чтобы модуль и направление вектора не хранились в виде компонентов а вычислялись вычисляться по требованию при вызове методов `magval ()` и `angval ()`. Предусмотрите необходимые конструкторы. Разработайте дружественную функцию для сложения двух векторов.

Задание 2. Ниже представлено описание класса `Time`:

```
class Time {
private:
    int hours; int minutes;
public:
    Time();
    Time (int h, int m = 0) ;
    void AddMin (int m) ;
    void AddHr(int h) ;
    void Reset (int h = 0, int m = 0) ;
    Time operator+(const Time & t) const;
    Time operator-(const Time & t) const;
    Time operator*(double n) const;
    friend Time operator*(double m, const Time & t)
}
```

Напишите реализацию для перегрузки операторов и дружественной функции.

Задание 3. Комплексное число состоит из двух частей — вещественной и мнимой. Один из способов записи такого числа выглядит как $(3.0, 4.0)$. Здесь 3.0 — вещественная часть, а 4.0 — мнимая. Предположим, что $a = (A, Bi)$ и $c = (C, Di)$. Ниже представлены некоторые операции с комплексными числами:

сложение: $a + c = (A + C, (B + D)i)$

вычитание: $a - c = (A - C, (B - D)i)$

умножение: $a * c = (Ax C - BxD, (Ax D + BxC)i)$

умножение (x — вещественное число): $xxc = (xxC, xxDi)$

сопряжение: $\sim a = (A, -Bi)$

Определите класс `complex` так, чтобы следующая программа могла использовать его с корректными результатами:

```
#include <iostream>
using namespace std;
#include "complexO.h" // во избежание конфликта с complex.h
int main()
{
    complex a (3.0, 4.0);    // инициализация значением (3,4i)
    complex c;
    cout << "Enter a complex number (q to quit) :\n";
    // Ввод комплексного числа (q для завершения) while (cin >> c)
    {
        cout << "c is " << c << "\n";    // значение c
        cout << "complex conjugate is " << ~c << "\n";
        // значение сопряженного числа cout << "a is " << a << "\n" ; 9 // значение
        a
        cout << "a + c is " << a + c << "\n"; // значение a + c
        cout << "a — c is " << a — c << "\n"; // значение a — c
        cout << "a * c is " << a * c << "\n"; // значение a * c
        cout << "2 •* c is " << 2 * c << '\n '; // значение 2 * c
        cout << "Enter a complex number (q to quit) :\n";
    }
    cout << "Done!\n"; return 0;
}
```

Не забывайте, что вы должны перегрузить операции `<<` и `>>`. В стандарте C++ уже присутствует поддержка комплексных чисел — и намного более развитая, чем в этом примере — в заголовочном файле `complex`, поэтому во избежание конфликтов назовите свой файл `complex.h`. Используйте `const` там, где это оправдано.

Ниже показан пример выполнения этой программы:

```
Enter a complex number (q to quit) : real: 10 imaginary: 12 c is (10,12i)
complex conjugate is (10,-12i) a is (3,4i) a + c is (13,16i) a - c is (-7,-8i) a *
c is (-18, 76i)
2 * c is (20, 24i)
```

```
Enter a complex number (q to quit) : real: q Done!
```

Обратите внимание, что благодаря перегрузке, `cin >> c` теперь запрашивает ввод вещественной и мнимой частей комплексного числа.

ЛАБОРАТОРНАЯ РАБОТА 3. КЛАССЫ И ДИНАМИЧЕСКОЕ ВЫДЕЛЕНИЕ ПАМЯТИ

Получение практических навыков в выделении динамической памяти под объекты классов.

Задание 1. Имеется следующее объявление класса:

```
class Cow {
char name[20]; char * hobby; double weight; public:
Cow();
Cow (const char * nm, const char * ho, double wt) ;
Cow(const Cow c&) ;
Cow & operator= (const Cow & c) ;
void ShowCow() const; // отображение всех данных cow
};
```

Напишите реализацию для этого класса и короткую программу, использующую все функции-члены.

Задание 2. Имеется следующий вариант класса Stack:

```
class Stack
private:
enum {MAX =10}; // константа, специфичная для класса
Item * pitems; // хранит элементы стека
int size; // количество элементов в стеке
int top; // индекс для верхнего элемента стека
public:
Stack (int n = 10); // создает стек с n элементами
Stack(const Stack & st) ;
Stack ();
bool isempty() const;
bool isfull() const;
// push() возвращает значение false, если стек уже полный,
//и true в противном случае
bool push(const Item & item);
// добавление элемента в стек //
bool pop () возвращает значение false, если стек уже пустой,
// и true в противном случае
bool pop(Item & item); // извлечение элемента из стека
```

```
Stack & operator= (const Stack & st) ;
```

Как понятно из закрытых членов, данный класс использует динамически выделенный массив для хранения элементов стека. Перепишите методы для соответствия новому представлению и напишите программу, которая демонстрирует работу всех методов, включая конструктор копирования и операцию присваивания.

Задание 3. Описать класс «Пациент поликлиники». При этом считать, что некоторые пациенты обладают большим приоритетом по отношению к остальным (ветераны, инвалиды): в классе определить два уровня приоритета. Определить для класса операцию сравнения $<$. Смоделировать поведение очереди в регистратуру с помощью шаблонного класса `priority_queue`. Программа должна считывать данные о приходящих пациентах с консоли или из файла, помещать их в очередь и выдавать на консоль информацию о последовательности работы с пациентами.

Задание 4. Описать класс «Автобус». Написать программу моделирования работы автобусного парка. Программа должна обеспечивать начальное формирование данных о всех автобусах в парке в виде списка (ввод с клавиатуры или из файла). Кроме того, программа должна обеспечивать имитацию выезда автобуса из парка: вводится номер автобуса, программа удаляет данные об этом автобусе из списка автобусов, находящихся в парке и записывает данные в список автобусов, находящихся на маршруте. Для представления необходимых списков использовать шаблонный класс `list`.

Задание 5. Написать программу «Телефонная книга». Программа должна обеспечивать загрузку в контейнер типа `map` базы телефонов и соответствующих им фамилий и формирование по нему другого контейнера типа `map` – базы фамилий и соответствующих им телефонов. Реализовать функции вывода телефона по заданной фамилии и фамилии по заданному телефону.

ЛАБОРАТОРНАЯ РАБОТА 4. НАСЛЕДОВАНИЕ

Получение практических навыков в области одиночного и множественного наследования классов. Базовый класс. Виртуальные методы.

Задание 1. Определить два класса с иерархией наследования в соответствии с предметной областью «Транспортное средство», «Автобус». Каждый класс снабдить свойствами и методами, определяющими их специфику. Предусмотреть метод `info()`, выводящий информацию об объекте. Предусмотреть конструкторы с параметрами и без параметров. Написать демонстрационную программу, создающую 4-5 объектов и демонстрирующую вызов их методов.

Задание 2. Определить класс «Электронное устройство» и два его класса-потомка «Мобильный телефон» и «Фотоаппарат». Определить конструкторы класса и метод `info()`, выводящий данные об объекте. В базовом классе объявить чистые виртуальные функции для выдачи сообщения при сохранении фотографии и начале записи видеофайла. Реализовать эти функции в классах-потомках. Они должны быть разными для разных устройств. Продемонстрировать возможности их вызова как виртуальных функций для указателя на базовый класс.

Задание 3. Напишите программу, которая приглашает пользователя вводить числа. После каждого введенного значения программа должна выдавать накопленную сумму введенных значений. Программа должна завершаться при вводе 0.

Задание 4. Дарья инвестировала 1000 руб под простые 10%. Другими словами, ежегодно инвестиция должна приносить 10% инвестированной суммы, т.е. 100 руб каждый год:

$$\text{прибыль} = 0,10 \times \text{исходный баланс}$$

В то же время Гликерья инвестировала 1000 руб под сложные 5%. Это значит, что прибыль составит 5% от текущего баланса, включая предыдущую накопленную прибыль:

$$\text{прибыль} = 0,05 \times \text{текущий баланс}$$

Клео зарабатывает 5% от 1000 руб в первый год, что дает ей 1050. На следующий год она зарабатывает 5% от 1050, что составляет 52,5 руб, и т.д. Напишите программу, которая вычислит, сколько лет понадобится для того, чтобы сумма баланса Гликерьи превысила сумму баланса Дарьи, с отображением значений обоих балансов за каждый год.

Задание 5. Выполните задание 4, но используя двумерный массив для сохранения данных о месячных продажах за 3 года. Выдайте общую сумму продаж за каждый год и за все годы вместе.

Задание 6. Разработайте структуру по имени `car`, которая будет хранить следующую информацию об автомобиле: название производителя в виде строки в символьном массиве или в объекте `string`, а также год выпуска автомобиля в виде целого числа. Напишите программу, которая запросит пользователя, сколько автомобилей необходимо включить в каталог. Затем программа должна применить `new` для создания динамического массива структур `car` указанного пользователем размера. Далее она должна пригласить пользователя ввести название производителя и год выпуска для наполнения данными каждой структуры в массиве. И, наконец, она должна отобразить содержимое каждой структуры.

Пример запуска программы должен выглядеть подобно следующему:

Сколько автомобилей поместить в каталог? 2

Автомобиль #1:

Введите производителя: Toyota

Укажите год выпуска: 2007

Автомобиль #2:

Введите производителя: Lexus

Укажите год выпуска: 2010

Вот ваша коллекция:

2007 Toyota 2010 Lexus

ЛАБОРАТОРНАЯ РАБОТА 5. ШАБЛОНЫ ФУНКЦИЙ И КЛАССОВ. ОБРАБОТКА ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦИЙ.

Получение практических навыков работы в области разработки шаблонов функций, обработки исключительных ситуаций.

Задание 1. Определить шаблон функции, осуществляющей поиск элемента в массиве. Функция возвращает 0, если объект не найден, и 1, если объект найден. Продемонстрировать работу шаблона на массивах объектов как встроенного типа, так и собственного класса. Для этого создать собственный класс и перегрузить в нём нужные функции-операции.

Задание 2. Описать шаблон класса для хранения данных. Данные должны храниться, например, в виде массива. Шаблон должен включать в себя функции добавления данных, удаления данных, вывода данных на консоль (`info()`). При этом добавление данных должно осуществляться так, чтобы каждый элемент встречался не более одного раза (то есть не добавлять

данное, если оно уже есть в массиве). Продемонстрировать работу шаблона на объектах как встроенного типа, так и собственного класса. Для этого создать собственный класс и перегрузить в нём нужные функции-операции.

Задание 3. Описать шаблон класса для хранения данных. Данные должны храниться, например, в виде массива. Шаблон должен включать в себя функции добавления данных, удаления данных, вывода данных на консоль (`info()`). При этом добавление данных должно осуществляться так, чтобы элемент добавлялся только в том случае, если он не более чем в 2 раза больше уже имеющегося максимального элемента (то есть не добавлять данное, если оно слишком велико). Продемонстрировать работу шаблона на объектах как встроенного типа, так и собственного класса. Для этого создать собственный класс и перегрузить в нём нужные функции-операции.

Задание 4. Описать шаблон класса для хранения данных. Данные должны храниться, например, в виде массива. Шаблон должен включать в себя функции добавления данных, удаления данных, вывода данных на консоль (`info()`) и метод, осуществляющий сортировку элементов по возрастанию. Предусмотреть обработку следующих исключений: добавление данных в заполненный массив, удаление данных из пустого массива, сортировка массива, где все элементы равны. Продемонстрировать работу шаблона на объектах как встроенного типа, так и собственного класса. Для этого создать собственный класс и перегрузить в нём нужные функции-операции.

Задание 5. Описать шаблон класса для хранения данных (контейнер). Данные должны храниться, например, в виде массива. Шаблон должен включать в себя функции добавления данных, удаления данных, вывода данных на консоль (`info()`). Реализовать итератор для перебора данных контейнера так, чтобы он перебирал элементы в порядке их возрастания. Продемонстрировать работу шаблона на объектах как встроенного типа, так и собственного класса. Для этого создать собственный класс и перегрузить в нём нужные функции-операции.

ЛАБОРАТОРНАЯ РАБОТА 6.

ОРГАНИЗАЦИЯ МНОГОПОТОЧНЫХ ПРОГРАММ. МЬЮТЕКСЫ.

Получение практических навыков работы с потоками.

Задание 1. Отсортировать элементы строк матрицы по убыванию. Сортировка элементов каждой строки выполняется отдельным потоком. Первые 10 элементов первой строки вывести на консоль для проверки.

Задание 2. Организовать программу, позволяющую имитировать работу с игровыми автоматами. Одновременно на одном автомате может играть только один человек. В процессе игры каждый автомат генерирует 3 числа в диапазоне от 1 до 3, если эти три числа совпали, то игрок получает выигрыш и автомат прекращает свою работу, если выпали разные числа, то автомат продолжает работу. Один игрок может сыграть несколько раз на одном автомате. Сумма выигрыша для разных автоматов задаётся произвольно, в случае победы игрок получает всю сумму которая есть в автомате. Программа должна выводить на экран фио или номер игроков и данные о результатах их игры. Работа программы заканчивается, когда не остаётся работающих автоматов.

Задание 3. Организовать программу, позволяющую имитировать работу регистратуры. В одном окне регистратуры могут одновременно обслуживать только одного пациента, у каждого окна есть определенное количество рабочего времени. Каждый пациент может занимать некоторое количество времени в окошке (например, от 5 до 20 минут), один пациент может обратиться в одно окно несколько раз (только не подряд). Программа должна выводить на экран фио или номер пациента и данные о времени, проведенном в окошке и в регистратуре вообще. Работа программы заканчивается тогда, когда заканчивается время работы окошек регистратуры.

Критерии оценки лабораторных работ

Оценка	Требования
<i>«зачтено»</i>	Оценка «зачтено» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.
<i>«не зачтено»</i>	Оценка «не зачтено» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ.

Контрольные задания

1. Разработать класс Прямоугольник со сторонами, параллельными осям координат.

Предусмотреть конструкторы по умолчанию и с параметром и функции-методы для реализации следующих возможностей:

- перемещения прямоугольников на плоскости,
- изменение размеров,

Перегрузить операторы сравнения для вывода наибольшего прямоугольника.

Написать программу, демонстрирующую работу с этим классом.

Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

2. Описать класс «массив вещественных чисел». В классе предусмотреть методы удаления и добавления элементов, конструкторы и деструктор. Предусмотреть метод info(), выводящий на экран информацию об объекте. Предусмотреть перегруженный оператор «==», выполняющий сравнение двух массивов с точки зрения их равенства. Перегрузить оператор [] для доступа к элементам массива. Написать демонстрационную программу, создающую объекты и проверяющую работу перегруженных операторов.

3. Описать базовый класс СТРОКА. Обязательные поля класса:

- указатель на char —хранит адрес динамически выделенной памяти для размещения символов строки;
- значение типа int —хранит длину строки в байтах.

Обязательные методы должны выполнять следующие действия:

- конструктор без параметров;
- конструктор, принимающий в качестве параметра Си-строку (заканчивается нулевым байтом);
- конструктор, принимающий в качестве параметра символ;
- конструктор копирования;
- получение длины строки;
- очистка строки (сделать строку пустой);
- деструктор.

Описать производный от СТРОКА класс СТРОКА_ИДЕНТИФИКАТОР. Строки данного класса строятся по правилам записи идентификаторов в языке Си могут включать в себя только те символы, которые могут входить в состав С-идентификаторов. Если исходные данные противоречат правилам записи идентификатора, то создается пустая СТРОКА_ИДЕНТИФИКАТОР.

Обязательные методы:

- конструктор без параметров;
- конструктор, принимающий в качестве параметра Си-строку (заканчивается нулевым байтом);
- конструктор, принимающий в качестве параметра символ;
- конструктор копирования;
- перевод всех символов строки в верхний регистр;
- перевод всех символов строки в нижний регистр;
- поиск первого вхождения символа в строку;
- деструктор.

Переопределить следующие операции:

- присваивание (=);
- сложение (+)
- операция конкатенации строк;
- вычитание (—)-из строки (первый операнд) удаляются все символы, входящие в строку, —второй операнд, при этом может получиться пустая строка;

-оператор — проверка на больше. Строка считается больше другой, если код символа первой строки в i -й позиции (i изменяется от 0 до $n-1$, где n —длина более короткой строки) больше кода символа в той же позиции во второй строке, длины строк могут не совпадать.

-оператор < — проверка на меньше. Строка считается меньше другой, если код символа первой строки в i -й позиции (i изменяется от 0 до $n-1$, где n —длина более короткой строки) меньше кода символа в той же позиции кода символа в той же позиции во второй строке, длины строк могут не совпадать

.

Разработчик вправе вводить любое (с обоснованием необходимости) число дополнительных полей и методов.

Написать тестовую программу, которая:

- динамически выделяет массив указателей на базовый класс (4–6);
- в режиме диалога заполняет этот массив указателями на производные классы, при этом экземпляры производных классов создаются динамически с заданием начальных значений;
- для созданных экземпляров производных классов выполняет проверку всех разработанных методов с выводом исходных данных и результатов на экран.

4. Описать шаблонный класс List для работы с однонаправленными списками в динамической памяти. Для объектов класса List определить операции проверки списка на пустоту, добавления элемента в начало списка,

в конец списка, подсчет числа вхождений элемента в список, удаление элемента из списка. Продемонстрировать работу с шаблонным классом для списка с целыми элементами и с элементами-строками.

5. Определить класс «Летательный аппарат» и два его класса-потомка «Вертолёт» и «Дирижабль». Определить конструкторы класса и метод `info()`, выводящий данные об объекте. В базовом классе объявить чистые виртуальные функции для взлёта и посадки аппарата. Реализовать эти функции в классах-потомках. Они должны быть разными для разных аппаратов (например – выводить разное сообщение). Продемонстрировать возможности их вызова как виртуальных функций для указателя на базовый класс.