

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Смирнов Сергей Николаевич
Должность: врио ректора
Дата подписания: 24.07.2025 17:15:44
Уникальный программный ключ:
69e375c64f7e975d4e8830e7b4fcc2ad1bf35f08

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Тверской государственный университет»

«Утверждаю»

Руководитель ООП

 Шаров Г.С.

«14» 05 2025 г.


Рабочая программа дисциплины (с аннотацией)

Основы программирования

Направление подготовки

01.03.01 Математика

Профиль подготовки

Математическое обеспечение экономической деятельности

Для студентов 1, 2 курсов

Форма обучения очная

Тверь, 2025

I. Аннотация

1. Цель и задачи дисциплины

Целями освоения дисциплины являются:

- получение базовых знаний, связанных с разработкой алгоритмов и программ;
- приобретение навыков работы с современными системами программирования;
- подготовка к осуществлению деятельности по созданию и применению средств математического обеспечения информационных систем;
- подготовка к работе, связанной с разработкой программного обеспечения и способов администрирования информационных систем и сетей;
- подготовка к работе по программно-информационному обеспечению научно-исследовательской, проектно-конструкторской и эксплуатационно-управленческой деятельности.

2. Место дисциплины в структуре ООП

Дисциплина относится к обязательной части блока 1 учебного плана – к дисциплинам, формирующим универсальные и общепрофессиональные компетенции.

Предварительные знания, необходимые для освоения дисциплины – знания, полученные при изучении школьной программы по информатике и ИКТ. Освоение данной дисциплины необходимо как предшествующее для следующих дисциплин: методы вычислений, информационно-коммуникационные технологии, история и перспективы развития компьютерной техники, математика в истории мировой культуры, программные средства математических вычислений, компьютерная математика и др.

Дисциплина изучается на 1, 2 курсах (1–4-й семестры).

3. Объём дисциплины: 13 зачётных единиц, 468 академических часов, в том числе:

контактная аудиторная работа: лекции 140 часов, лабораторные работы 70 часов, практические занятия 70 часов;

самостоятельная работа: 188 часов, в том числе контроль 27 часов.

4. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Планируемые результаты освоения образовательной программы (формируемые компетенции)	Планируемые результаты обучения по дисциплине
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1 Анализирует задачу, выделяя ее базовые составляющие УК-1.2 Определяет, интерпретирует и ранжирует информацию, требуемую для

	решения поставленной задачи УК-1.3 Осуществляет поиск информации для решения поставленной задачи по различным типам запросов УК-1.4 При обработке информации отличает факты от мнений, интерпретаций, оценок, формирует собственные мнения и суждения, аргументирует свои выводы и точку зрения УК-1.5 Рассматривает и предлагает возможные варианты решения поставленной задачи, оценивая их достоинства и недостатки
ОПК-1 Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности	ОПК-1.1 Осуществляет отбор теоретического и практического материала ОПК-1.2 Решает типовые задачи в рамках профессиональной деятельности ОПК-1.3 Использует различные методы и приемы решения задач профессиональной деятельности
ОПК-4 Способен решать задачи профессиональной деятельности с использованием существующих информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	ОПК-4.1 Отбирает теоретический материал в области ИКТ для решения профессиональных задач ОПК-4.2 Применяет современные методы сбора и представления данных ОПК-4.3 Учитывает требования информационной безопасности при решении профессиональных задач
ОПК-5 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-5.1 Анализирует существующие алгоритмы и методы программирования для решения типовых задач прикладного характера ОПК-5.2 Применяет известные алгоритмы и методы программирования для решения типовых задач прикладного характера ОПК-5.3 Используя различные методы и приемы, разрабатывает алгоритмы и компьютерные программы, пригодные для практического применения

5. Форма промежуточной аттестации и семестр прохождения

зачёт (2 семестр), экзамен (4 семестр).

6. Язык преподавания: русский.

II. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

№	Учебная программа – наименование разделов и тем	Всего (час.)	Контактная работа (час.)			Самостоятельная работа (час.)
			Лекции	Лаборатор- ные занятия	Практичес- кие занятия	
	Раздел 1. Структурное программирование					
1.	Тема 1.1. Общие принципы построения и использования языков программирования	12	4	1	1	6
2.	Тема 1.2. Лексические основы с++. Типы и объявления	24	8	5	5	6
3.	Тема 1.3. Операторы и инструкции с++	26	10	5	5	6
4.	Тема 1.4. Указатели и массивы	36	12	6	6	12
5.	Тема 1.5. Функции	44	18	9	9	8
6.	Тема 1.6. Файлы	30	10	5	5	10
7.	Тема 1.7. Обработка исключений	28	10	5	5	8
	Раздел 2. Основы объектно-ориентированной парадигмы Программирования на с++					
8.	Тема 2.1. Объектно- ориентированное программирование и с++	25	6	3	3	13
9.	Тема 2.2. Класс – абстрактный тип данных	62	16	8	8	30
10.	Тема 2.3. Перегрузка операторов, копирование и преобразование	32	10	5	5	12
11.	Тема 2.4. Наследование. Производные классы	45	14	7	7	17
	Раздел 3. Основы					

	обобщенной парадигмы программирования на с++					
12.	Тема 3.1. Шаблоны как поддержка обобщенного программирования	32	6	3	3	20
13.	Тема 3.2. Использование параметризованных классов	22	6	3	3	10
14.	Тема 3.3. Организация стандартной библиотеки с++	50	10	5	5	30
	ИТОГО	468	140	70	70	188

III. Образовательные технологии

Преподавание учебной дисциплины строится на сочетании аудиторных занятий и различных форм самостоятельной работы студентов.

Также на занятиях практикуется выполнение заданий в малых группах, письменные работы, моделирование дискуссионных ситуаций, работа с раздаточным материалом, привлекаются ресурсы сети INTERNET. Курс предусматривает выполнение контрольных и самостоятельных работ, письменных домашних заданий. В качестве форм контроля используются различные варианты взаимопроверки и взаимоконтроля.

Интерактивное взаимодействие студентов с одной стороны и преподавателя с другой, а также студентов между собой и с преподавателем во время практических занятий.

Образовательные технологии

1. Дискуссионные технологии
2. Информационные (цифровые)
3. Технологии развития критического мышления

Современные методы обучения

1. Активное слушание
2. Лекция (традиционная)

IV. Оценочные материалы для проведения текущей и промежуточной аттестации

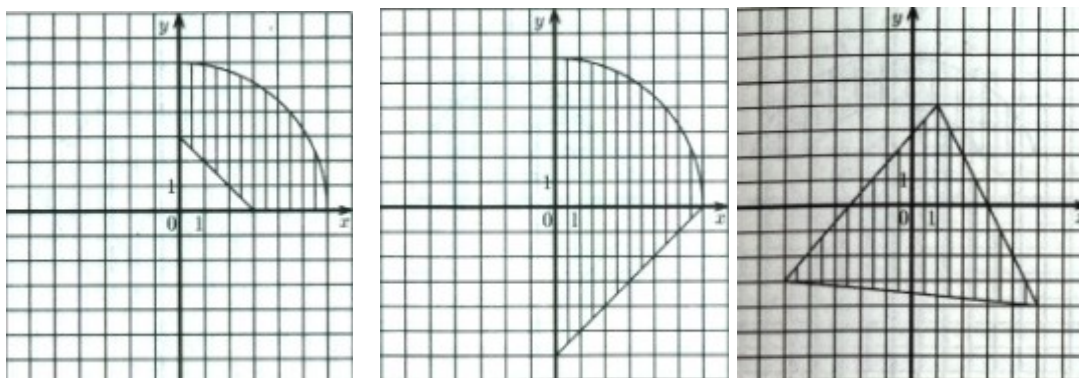
1. Оценочные материалы для проведения текущей аттестации

Практические задания

1. Найти расстояние между двумя точками с заданными координатами (x_1, y_1) , (x_2, y_2) . На плоскости. Расстояние вычисляется по формуле. $l = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

2. Поменять местами содержимое переменных А и В и вывести новые значения А и В.
3. С клавиатуры вводится число. После этого на экран выводится следующее сообщение. Например если введено число 173, то сообщение будет «Следующее число после 173 это 174, а предыдущее 172».
4. Составьте программу вычисления площади треугольника по формуле Герона $S = \sqrt{p(p-a)(p-b)(p-c)}$, где полупериметр вычисляется по формуле $p = \frac{a+b+c}{2}$, если даны длины сторон треугольника.
5. Даны три точки А, В, С на числовой оси. Найти длины отрезков АС и ВС и их сумму.
6. Даны переменные А, В, С. Изменить их значения, переместив содержимое А в В, В в С, С в А, и вывести новые значения переменных А, В, С.
7. Составить программу вычисления: $y = \frac{\sqrt{1+|2+a|}}{3,28}$, где $a = \sqrt{2x+12}$ и x водится с клавиатуры.
8. Дано число а. Не используя никаких операций, кроме умножения, и никаких функций получите: a^6 за три операции, a^7 за четыре операции, a^8 за три операции, a^9 за четыре операции, a^{10} за четыре операции.
9. Даны координаты двух противоположных вершин прямоугольника: (x_1, y_1) , (x_2, y_2) . Стороны прямоугольника параллельны осям координат. Найти периметр и площадь данного прямоугольника.
10. С клавиатуры вводится двузначное число. Вывести на экран его последнюю цифру. (Разряд единиц.)
11. Дано натуральное число. Найдите число десятков в его записи (т.е. вторую цифру с права в его записи).
12. Найти сумму цифр заданного четырехзначного числа.
13. Найти площадь треугольника, две стороны которого равны a и b . А угол между ними γ . (Воспользоваться теоремой косинусов и формулой Герона.)
14. Вычислить площадь кольца, ширина которого равна Н, а отношение радиуса большей окружности к радиусу меньшей окружности D.
15. Дано целое число, лежащее в диапазоне 1-999. Вывести его строку описание вида «четное двузначное число», «нечетное трехзначное число», и т.д.
16. Даны три целых числа, одно из которых отлично от двух других, равных между собой. Определить порядковый номер числа, отличного от остальных.
17. . Даны три целых числа. Если они упорядочены по возрастанию или убыванию, то удвоить их. В противном случае оставить без изменений. Вывести эти числа на экран.
18. Даны три числа. Найти наименьшее из них.
19. Даны три числа. Найти среднее из них (то есть число, расположенное между наименьшим и наибольшим).
20. С клавиатуры вводятся два числа А и В. Меньшее из этих чисел заменить их полусуммой, а большее удвоенным произведением

21. Составить программу определяющую, принадлежит ли точка с координатами X и Y, введенными с клавиатуры, фигуре на плоскости.



22. Найдите сумму всех двузначных чисел.

23. Дано целое число N ($N > 0$). Используя один цикл найти сумму $1 + 1/2 + 1/3 + \dots + 1/N$.

24. Дано целое число N ($N > 0$). Найти произведение N сомножителей $1, 1*1, 2*1, 3*1, \dots$

25. Дано целое число N ($N > 0$). Найти значение выражения из N слагаемых (знаки чередуются). Условный оператор не использовать.

$$1, 1 - 1, 2 + 1, 3 - \dots$$

26. Дано вещественное число A и целое число N ($N > 0$). Используя один цикл, вывести все целые степени числа A от 1 до N.

27. Дано вещественное число A и целое число N ($N > 0$). Используя один цикл найти сумму:

$$1 + A + A^2 + A^3 + \dots + A^N$$

28. Дано целое число N ($N > 0$). Найти факториал числа N. $N! = 1*2*3*\dots*N$

29. Дано целое число N ($N > 0$). Используя один цикл найти сумму: $1! + 2! + 3! + \dots + N!$

30. Практическая работа «Расчет конечных сумм»

Вид суммы	Контрольное значение
$1 + 2 + 3 + 4 + \dots + N$	$K = \frac{N(N+1)}{2}$
$1 + 3 + 5 + 7 + \dots + (2N-1)$	$K = N^2$
$2 + 4 + 6 + 8 + \dots + 2N$	$K = N(N+1)$
$1^2 + 2^2 + 3^2 + 4^2 + \dots + N^2$	$K = \frac{N(N+1)(2N+1)}{6}$
$1^2 + 3^2 + 5^2 + \dots + (2N-1)^2$	$K = \frac{N(4N^2-1)}{3}$
$1^3 + 2^3 + 3^3 + 4^3 + \dots + N^3$	$K = \frac{N^2(N+1)^2}{4}$
$1^3 + 3^3 + 5^3 + \dots + (2N-1)^3$	$K = N^2(2N^2-1)$

$1^4 + 2^4 + 3^4 + 4^4 + \dots + N^4$	$K = \frac{(N^2 + N)(2N + 1)(3N^2 + 3N - 1)}{30}$
---------------------------------------	---

31. Найти все четырехзначные числа для цифр из которых состоит это число выполняется равенство $AB - CD = A+B+C+D$. (Где A,B,C,D – цифры из которых состоит число.)

32. Найти все трехзначные числа, которые при делении на 2 дают остаток 1, при делении на 3 остаток 2, при делении на 4 остаток 3, а само число делится на 5 (без остатка).

33. Число Армстронга - такое число из k цифр, для которого сумма k-х степеней его цифр равна самому числу. Например, число 153 является числом Армстронга (k=3) т.к. $153 = 1^3 + 5^3 + 3^3$. Требуется написать программу для нахождения всех трёхзначных чисел Армстронга.

34. Написать программу, осуществляющую заполнение массива из 10 элементов случайными числами из интервала от -5 до 5 и вывода их в строчку.

35. Написать программу, заполняющую массив из 10 элементов по следующему правилу: $A(i) = (2*i - 1) / \sin(i)$.

36. Найдите 6-ой член последовательности $A(i) = A(i-1) * A(i-1) + 1$, если $A(1) = 5$.

37. Заполнить массив из 10 элементов случайными числами из интервала от -10 до 10, вывести на экран, а затем увеличить каждый элемент массива на 1 и повторно вывести на экран.

38. Заполнить двумерный массив $N \times N$ случайными числами из интервала $[-10; 10]$ и найти сумму элементов в каждой строке.

39. Заполнить двумерный массив $N \times N$ случайными числами из интервала $[-5; 5]$ и найти произведение элементов в каждом столбце.

40. Заполнить двумерный массив $N \times N$ случайными числами из интервала $[-10; 10]$ и найти сумму положительных элементов в каждой строке.

41. Элемент матрицы назовем седловой точкой, если он является наименьшим в своей строке и одновременно наибольшим в своем столбце. Или, наоборот, является наибольшим в своей строке и наименьшим в своем столбце. Для матрицы $N \times M$ введенной с клавиатуры, вывести на экран индексы всех седловых точек.

42. Заполнить двумерный массив $N \times N$ случайными числами из интервала $[-10; 10]$, найти максимальный элемент, стоящий на главной и побочной диагоналях. Поменять его местами с элементом стоящим на пересечении этих диагоналей.

43. Заполнить двумерный массив $N \times N$ случайными числами из интервала $[-10; 10]$, найти максимальный элемент в массиве и удалить строку, в которой находится этот элемент.

44. Заполнить двумерный массив $N \times N$ случайными числами из интервала $[-10; 10]$, найти минимальный элемент в массиве и удалить столбец, в которой находится этот элемент.

45. С клавиатуры вводится строка символов, вывести на экран символ, стоящий на k – ом месте. k – вводится с клавиатуры.

46. С клавиатуры вводится слово, верно ли, что оно начинается и заканчивается на одну и ту же букву.

47. Дано два слова. Верно ли, что первое слово заканчивается на ту же букву, на которую начинается второе.

48. Описать структуру с именем STUDENT , содержащую следующие поля: NAME – фамилия и инициалы; номер группы; успеваемость (массив из пяти элементов). Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT; записи должны быть упорядочены по возрастанию номера группы; 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.

49. Описать класс, реализующий стек. Написать программу, использующую этот класс для моделирования Т-образного сортировочного узла на железной дороге. Программа должна разделять на два направления состав, состоящий из вагонов двух типов (на каждое направление формируется состав из вагонов одного типа). Предусмотреть возможность формирования состава из файла и с клавиатуры.

50. Описать класс, реализующий бинарное дерево, обладающее возможностью добавления новых элементов, удаления существующих, поиска элемента по ключу, а также последовательного доступа ко всем элементам. Написать программу, использующую этот класс для представления англо-русского словаря. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса. Предусмотреть возможность формирования словаря из файла и с клавиатуры.

51. Построить систему классов для описания плоских геометрических фигур: круга, квадрата, прямоугольника. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и вращения на заданный угол. Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Контрольные задания

1. Циклические вычислительные процессы

Вариант 1

Вычислить и вывести на экран в виде таблицы значения функции F на интервале от X нач. до X кон. С шагом dX .

$$F = \begin{cases} \frac{1}{ax-b} & \text{при } x+5 < 0 \text{ и } c = 0 \\ \frac{x-a}{x-c} & \text{при } x+5 > 0 \text{ и } c \neq 0 \\ \frac{x}{c} & \text{в остальных случаях} \end{cases}$$

где a, b, c – действительные числа.

Функция F должна принимать действительное значение, если выражение (Ац И Вц) ИЛИ (Вц И Сц) не равно нулю, и целое значение в противном случае. Через Ац, Вц и Сц обозначены целые части значений a, b, c операции И и ИЛИ – поразрядные. Значения a, b, c, X нач., X кон., dX ввести с клавиатуры.

Вариант 2

Вычислить и вывести на экран в виде таблицы значения функции F на интервале от X нач. до X кон. С шагом dX .

$$F = \begin{cases} ax^2 + b & \text{при } x+5 < 0 \text{ и } c = 0 \\ \frac{x-a}{x} & \text{при } x+5 > 0 \text{ и } c \neq 0 \\ \frac{10x}{c-4} & \text{в остальных случаях} \end{cases}$$

где a, b, c – действительные числа.

Функция F должна принимать действительное значение, если выражение (Ац ИЛИ Вц) И (Ац И Сц) не равно нулю, и целое значение в противном случае. Через Ац, Вц и Сц обозначены целые части значений a, b, c операции И и ИЛИ – поразрядные. Значения a, b, c, X нач., X кон., dX ввести с клавиатуры.

Вариант 3

Вычислить и вывести на экран в виде таблицы значения функции F на интервале от X нач. до X кон. С шагом dX .

$$F = \begin{cases} ax^2 + bx + c & \text{при } a < 0 \text{ и } c \neq 0 \\ \frac{-a}{x-c} & \text{при } a > 0 \text{ и } c = 0 \\ a(x+c) & \text{в остальных случаях} \end{cases}$$

где a, b, c – действительные числа.

Функция F должна принимать действительное значение, если выражение Ац И (Вц ИЛИ Сц) не равно нулю, и целое значение в противном случае. Через Ац, Вц и Сц обозначены целые части значений a, b, c операции И и ИЛИ – поразрядные. Значения a, b, c, X нач., X кон., dX ввести с клавиатуры.

2. Одномерные массивы и указатели

Размерности массивов задаются именованными константами.

Вариант 1

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

- 1) сумму отрицательных элементов массива;
- 2) произведение элементов массива, расположенных между максимальным и минимальным элементами.

Упорядочить элементы массива по возрастанию.

Вариант 2

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

- 1) сумму положительных элементов массива;
- 2) произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.

Упорядочить элементы массива по убыванию.

Вариант 3

В одномерном массиве, состоящем из n целых элементов, вычислить:

- 1) произведение элементов массива с четными номерами;
- 2) сумму элементов массива, расположенных между первым и последним нулевыми элементами.

Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом – все отрицательные.

3. Двумерные массивы

Вариант 1

Дана целочисленная прямоугольная матрица. Определить:

- 1) количество столбцов, содержащих хотя бы один нулевой элемент;
- 2) номер строки, в которой находится самая длинная серия одинаковых элементов.

Вариант 2

Дана целочисленная прямоугольная матрица. Определить:

- 2) количество строк, не содержащих ни одного нулевого элемента;
- 3) максимальное из чисел, встречающихся в заданной матрице более одного раза.

Вариант 3

Дана целочисленная квадратная матрица. Определить:

- 4) сумму элементов в тех столбцах, которые не содержат отрицательных;
- 5) минимум среди сумм модулей элементов диагоналей, параллельных побочной диагонали матрицы.

4. Структуры

Вариант 1

1. Описать структуру с именем STUDENT, содержащую следующие поля:

- фамилия и инициалы;
- номер группы;
- успеваемость (массив из пяти элементов).

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT; записи должны быть упорядочены по возрастанию среднего балла;
- вывод на дисплей фамилий и номеров групп для всех студентов, имеющих оценки 4 и 5;
- если таких студентов нет, вывести соответствующее сообщение.

Вариант 3

1. Описать структуру с именем STUDENT, содержащую следующие поля:

- фамилия и инициалы;
- номер группы;
- успеваемость (массив из пяти элементов).

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT; записи должны быть упорядочены по алфавиту;
- вывод на дисплей фамилий и номеров групп для всех студентов, имеющих хотя бы одну оценку 2;
- если таких студентов нет, вывести соответствующее сообщение.

5. Строки и файлы

Вариант 1

Написать программу, которая считывает из текстового файла три предложения и выводит их в обратном порядке.

Вариант 2

Написать программу, которая считывает текст из файла и выводит на экран только предложения, содержащие введенное с клавиатуры слово.

Вариант 3

Написать программу, которая считывает английский текст из файла и выводит на экран слова, начинающиеся с гласных букв.

6. Функции

Вариант 1

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

- 1) сумму отрицательных элементов массива;
- 2) произведение элементов массива, расположенных между максимальным и минимальным элементами.

Упорядочить элементы массива по возрастанию.

Оформить каждый пункт задания в виде функции. Все необходимые данные для функции должны передаваться им в качестве параметров. Использование глобальных переменных в функциях не допускается.

Вариант 2

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

- 1) сумму положительных элементов массива;
- 2) произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.

Упорядочить элементы массива по убыванию.

Оформить каждый пункт задания в виде функции. Все необходимые данные для функции должны передаваться им в качестве параметров. Использование глобальных переменных в функциях не допускается.

Вариант 3

В одномерном массиве, состоящем из n целых элементов, вычислить:

- 1) произведение элементов массива с четными номерами;
- 2) сумму элементов массива, расположенных между первым и последним нулевыми элементами.

Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом – все отрицательные.

Оформить каждый пункт задания в виде функции. Все необходимые данные для функции должны передаваться им в качестве параметров. Использование глобальных переменных в функциях не допускается.

7. Динамические структуры данных

Вариант 1

Составить программу, которая содержит динамическую информацию о наличии автобусов в автобусном парке.

Сведения о каждом автобусе включают:

1. номер автобуса;
2. фамилию и инициалы водителя;
3. номер маршрута.
4. Программа должна обеспечивать:
5. начальное формирование данных обо всех автобусах в парке в виде списка;
6. при выезде каждого автобуса из парка вводится номер автобуса, и программа удаляет данные об этом автобусе из списка автобусов, находящихся в парке, и записывает эти данные в список автобусов, находящихся на маршруте;
7. при въезде каждого автобуса в парк вводится номер автобуса, и программа удаляет данные об этом автобусе из списка автобусов, находящихся на маршруте, и записывает эти данные в список автобусов, находящихся в парке;
8. по запросу выдаются сведения об автобусах, находящихся в парке, или об автобусах, находящихся на маршруте.

Вариант 2

Составить программу, которая содержит текущую информацию о книгах в библиотеке.

Сведения о книгах включают:

- номер УДК;
- фамилию и инициалы автора;
- название;
- год издания;
- количество экземпляров данной книги в библиотеке;
- Программа должна обеспечивать:
- начальное формирование данных обо всех книгах в библиотеке в виде двоичного дерева;
- добавление данных о книгах, вновь поступающих в библиотеку;
- удаление данных о списываемых книгах;
- по запросу выдаются сведения о наличии книг в библиотеке, упорядоченные по годам издания.

Вариант 3

Составить программу, которая содержит текущую информацию о заявках на авиабилеты.

Каждая заявка включает:

- пункт назначения;
- номер рейса;
- фамилию и инициалы пассажира;
- желаемую дату вылета.
- Программа должна обеспечивать:
- хранение всех заявок в виде списка;
- добавление заявок в список;
- удаление заявок;
- вывод заявок по заданному номеру рейса и дате вылета;
- вывод всех заявок.

8. Знакомство со стандартной библиотекой STL

vector

1. Прибавить к каждому элементу массива среднее арифметическое его положительных элементов
2. Каждый элемент массива должен быть умножен на минимальный элемент исходного массива
3. Умножить элементы массива, делящиеся на 3 без остатка, на среднее арифметическое элементов массива, делящихся на 2 без остатка
4. Разделить элементы массива на половину максимального элемента
5. Отсортировать элементы массива в порядке возрастания суммы составляющих их цифр

6. Умножить каждый отрицательный элемент массива на произведение максимального и минимального элементов исходного массива

7. Умножить каждый элемент массива на максимальный элемент исходного массива и разделить на минимальный элемент исходного массива

8. Прибавить к каждому элементу массива сумму трех минимальных элементов массива

9. Элементы, стоящие на четных позициях массива умножить на 2, а из элементов, стоящих на нечетных позициях, вычесть сумму всех неотрицательных элементов

10. Каждую четверку элементов массива (либо оставшуюся часть массива меньшей длины) отсортировать в порядке возрастания

string

1. Разработайте функцию `&`, удаляющую из строки, переданной в параметре, лишние пробелы. Лишними считаются все пробелы в начале и конце строки, а также дополнительные пробелы между словами.

2. Разработайте с ее использованием функцию, выполняющую удаление лишних пробелов из каждой входного потока символов и вывод результирующих строк в выходной поток.

3. Разработайте функцию `std::string HtmlEncode(std::string const& text)`, выполняющую кодирование специальных символов строки `text` соответствующими сущностями HTML:

- “ (двойная кавычка) заменяется на `"`;
- ‘ (апостроф) заменяется на `'`;
- < (знак меньше) заменяется на `<`;
- > (знак больше) заменяется на `>`;
- & (амперсанд) заменяется на `&`;

Разработайте на ее основе программу, выполняющую html-кодирование текста, поступающего со стандартного потока ввода, и выводящую результат в стандартный поток вывода.

4. Разработайте функцию, выполняющую декодирование html-сущностей строки, перечисленных в варианте 3, обратно в их символьное представление.

Разработайте на ее основе программу, выполняющую декодирование html-сущностей текста, поступающего со стандартного потока ввода, и выводящую результат в стандартный поток вывода.

map

5. Разработайте программу-словарь, осуществляющую перевод **слов и словосочетаний**, поступающих со стандартного потока ввода, **с английского языка на русский** с использованием заданного файла словаря и выводящую результат перевода в стандартный поток вывода. Если вводимое слово или словосочетание, отсутствует в словаре, программа должна попросить пользователя ввести перевод и запомнить его, в случае, если пользователь ввел

непустую строку. Для выхода из диалога с программой пользователь должен ввести строку, состоящую из трех точек. Перед выходом программа спрашивает о необходимости сохранить изменения в файле словаря, в том случае, если в словарь были добавлены фразы во время текущей сессии работы с программой. Имя файла словаря передается программе с помощью параметра командной строки.

Stack

Обратная польская запись

1. Преобразовать выражение в ОПЗ с использованием стека
2. Реализовать алгоритм вычисления выражения, записанного в ОПЗ с использованием стека

9. ООП

Цель работы – освоение и закрепление основных навыков объектно-ориентированного программирования, связанных с проектирование спецификаций класса, его структуры и операций интерфейса с учетом свойства инкапсуляции.

Вариант 1

Описать класс, реализующий стек. Написать программу, использующую этот класс для моделирования Т-образного сортировочного узла на железной дороге. Программа должна разделять на два направления состав, состоящий из вагонов двух типов (на каждое направление формируется состав из вагонов одного типа). Предусмотреть возможность формирования состава из файла и с клавиатуры.

Вариант 2

Описать класс, реализующий бинарное дерево, обладающее возможностью добавления новых элементов, удаления существующих, поиска элемента по ключу, а также последовательного доступа ко всем элементам. Написать программу, использующую этот класс для представления англо-русского словаря. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса. Предусмотреть возможность формирования словаря из файла и с клавиатуры.

Вариант 3

Построить систему классов для описания плоских геометрических фигур: круга, квадрата, прямоугольника. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и вращения на заданный угол. Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 4

Построить описание класса, содержащего информацию о почтовом адресе организации. Предусмотреть возможность отдельного изменения

составных частей адреса, создания и уничтожения объектов этого класса. Написать программу, демонстрирующую работу с этим классом. Программа должна

Вариант 5

Составить описание класса для представления комплексных чисел. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

Вариант 6

Составить описание класса для объектов-векторов, задаваемых координатами концов в трехмерном пространстве. Обеспечить операции сложения и вычитания векторов с получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, косинуса угла между векторами. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

Вариант 7

1. Создать класс *Employee*.
 - **поля:** *имя, фамилия, должность, дата рождения, дата найма на работу, зарплата.*
 - для полей *дата рождения, дата найма* использовать класс *Data*.
2. поля *firstName, lastName, position* разместить в динамической памяти (*new, delete*).
3. Реализовать конструкторы (3 шт) и деструктор
4. **Реализовать методы:** перевод сотрудника на другую должность, повысить/понижить зарплату, вычислить возраст сотрудника, подсчитать стаж работы, в зависимости от стажа повысить зарплату (15 лет – 3%, 20 лет – 5%, 25 лет – 10%, ≥ 30 - 15%).
5. Создать бригаду рабочих (Массив из 5 объектов *Employee*).
6. + свое добавление.

Вариант 8

Создание класса *Date*. (начало времён 1901 год)

```
main(){
...
Date d1, d2, d3, ... ;

Date d4(1, 1, 2015);

d.setDate(1999, 1, 1);
...
Date datesArray[N];
datesArray[0] = d1;
...
vector <Date> dates;
dates.Add(d1);
...
}
```

```

d1.Print();

d1.futureDate(10);
d1.next();
d1.pastDate(10);
int k = d1.countDaysBetweenTwoDates(d2);
d1.compareDate(dates[2]);
sort(datesArray, N); // можно использовать библиотечную сортировку
sort(dates, dates.count); // можно использовать библиотечную сортировку
...
}

```

1. Правила оформления кода (для C++) -> отчёт в файле Word
 2. Программа -> документация по классу (назначение каждого члена класса) в файле Word
- в программе:
3. Комментарии к каждому методу
 4. Продумать спецификаторы доступа.
 5. "Дробить" функции в классе.
 6. (*) + к программе добавить свой код

10. ООП. Наследование и полиморфизм

Вариант 7

Описать класс Circle, соответствующую окружностям. Окружность задаётся координатами центра и радиусом. Определить функцию, проверяющую, пересекаются ли две окружности. Написать программу, использующую эту функцию. Предусмотреть обработку исключительных ситуаций.

Контрольная работа № 1

- В массиве Z(N) определить количество положительных чисел и чисел, принадлежащих интервалу [-7;5.2], все отрицательные числа переписать в массив P.
- Написать функцию, которая определяет, является ли заданная целочисленная квадратная матрица: ортонормированной, т.е. такой, в которой скалярное произведение каждой пары различных строк равно 0, а скалярное произведение каждой строки на себя равно 1.

Контрольная работа № 2

Вводится текст, написанный заглавными и строчными латинскими и русскими буквами и, возможно, содержащий цифры, а также другие символы. *Текст состоит из слов, разделяемых пробелами и другими общепринятыми знаками препинания.* Выведите на экран:

- Эту же последовательность слов, удалив из нее *повторные* вхождения слов и заменив все строчные буквы заглавными.
- Все *различные* слова, указав после каждого из них (в скобках) *число* его вхождений в текст.

Контрольная работа № 3

Описать структуру с именем STUDENT, содержащую следующие поля:

- фамилия и инициалы;
- номер группы;
- успеваемость (массив из пяти элементов).

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT; записи должны быть упорядочены по возрастанию среднего балла;
- вывод на дисплей фамилий и номеров групп для всех студентов, имеющих оценки 4 и 5; если таких студентов нет, вывести соответствующее сообщение.

Вопросы для контрольных тестов и самоконтроля

1. Что определяет класс? Чем обличается класс от объекта?

Класс определяет пользовательский тип: описание данных и операций над ними. Объект - это конкретный экземпляр класса со своим состоянием.

2. Можно ли объявлять массив объектов? А массив классов?

Да. Нет. Только в метапрограммировании. `boost::tuple`

3. Разрешается ли объявлять указатель на объект? А указатель на класс?

Да. Нет.

4. Допускается ли передавать объекты в качестве параметров, и какими способами? А возвращать как результат?

По ссылке, по указателю. Возвращать также. По значению передаётся копия объекта.

5. Как называется использование объекта одного класса в качестве поля другого класса?

Композиция.

6. Является ли структура классом? Чем класс отличается от структуры?

Да. По умолчанию в классе все поля `private`

7. Какие ключевые слова в C++ обозначают класс?

`class`, `struct`

8. Объясните принцип инкапсуляции.

Соккрытие деталей реализации.

9. Что такое композиция?

Использование объекта одного класса в качестве поля другого.

10. Для чего используются ключевые слова `public` и `private`?

Для определения области видимости.

11. Можно ли использовать ключевые слова `public` и `private` в структуре?

Да.

12. Существуют ли ограничения на использование `public` и `private` в классе? А в структуре?

Нет. Нет.

13. Обязательно ли делать поля класса приватными?
Нет.

14. Что такое метод? Как вызывается метод?
Метод - это функция, определённая в классе.
Если статический, то вызывается для класса, если обычный - то для объекта класса.

15. Может ли метод быть приватный?
Да.

16. Как определить метод непосредственно внутри класса? А вне класса? Чем эти определения отличаются?
`class A { void method x{} };`

17. Можно в методах присваивать параметрам значения по умолчанию?
Да. Существуют дополнительные ограничения.

18. Что обозначается ключевым словом `this`?
Указатель на объект, который вызвал метод.

19. Зачем нужны константные методы? Чем отличается определение константного метода от обычного?
Чтобы предотвратить случайное изменение данных внутри метода и показать, что метод не меняет состояние класса.

20. Может ли константный метод вызываться для объектов-переменных? А обычный метод — для объектов-констант?
Да. Нет.

21. Объясните принцип полиморфизма.
Возможность работать с объектами разных классов одинаковым образом.
Поддержка различного поведения родственных классов, предоставляемого через единый интерфейс базового класса.
Есть статический - на этапе компиляции. Шаблоны, перегрузка функций и операторов.
Есть динамический - виртуальные функции. Точное значение операции определяется объектом для которого она вызывается.

22. Что будет напечатано в результате работы следующей программы на Си++?

```
class X {
public:
    virtual void f() {cout << "X::f\n"; g(); }
    void g() { cout << "X::g\n";}
};
class Y : public X {
public:
    void f() { cout << "Y::f\n"; }
    void g() { cout << "Y::g\n"; f();}
};
class Z : public Y {
public:
    void f() { cout << "Z::f\n"; }
```

```

        void g() { cout << "Z::g\n"; f();}
    };
    void P(X*px,Y*py) {
        px->f(); px->g(); py->f(); py->g();
        delete px; delete py;
    }

    int main() {
        P(new X, new Y);
        P(new Y, new Z);
        return 0;
    }

```

2. Оценочные материалы для проведения промежуточной аттестации

Планируемый образовательный результат (компетенция, индикатор)	Типовые контрольные задания	Критерии оценивания и шкала оценивания
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач УК-1.1 Анализирует задачу, выделяя ее базовые составляющие УК-1.2 Определяет, интерпретирует и ранжирует информацию, требуемую для решения поставленной задачи УК-1.3 Осуществляет поиск информации для решения поставленной задачи по различным типам запросов УК-1.4 При обработке информации отличает факты от мнений, интерпретаций, оценок, формирует собственные мнения и суждения, аргументирует свои выводы и точку зрения УК-1.5 Рассматривает и предлагает возможные варианты решения поставленной задачи, оценивая их достоинства и недостатки	1. Парадигмы программирования, их особенности. Структурное программирование. Процедурное программирование. Модульное программирование. Объектно-ориентированное программирование. Функциональное программирование. Логическое программирование. Обобщенное программирование. (УК-1.2, УК-1.3, УК-1.4) 2. Типы данных в языке C++. Размеры данных. Примеры. Переменные. Константы. Примеры. Явное и неявное преобразование типов. Объявление переменных и констант в языке Си. Область видимости переменных и время жизни объектов. Локальные и глобальные переменные. Переменные, классы памяти. Статическое, автоматическое и динамическое размещение данных. Особенности и различия. (УК-1.1, УК-1.2, УК-1.3, УК-1.5) 3. Спроектируйте и реализуйте класс Data.	• <i>Имеется полный анализ задачи, предложен верный алгоритм решения задачи, приведено полное верное решение задачи, включающее правильный ответ</i> – 10 – 12 баллов • <i>Имеется полный анализ задачи, предложен верный алгоритм решения задачи, приведено решение задачи, но получен неправильный ответ из-за арифметической / решение недостаточно обосновано / в решении имеются лишние или неверные записи, не отделенные от решения</i> – 8 - 9 балла • <i>Неполный анализ задачи, предложен верный алгоритм решения задачи, но имеется верное решение лишь части задачи из-за логической ошибки</i> – 5 – 7 баллов • <i>Не соответствует требованиям, изложенным выше</i> – 0 - 4 баллов

	(УК-1.1, УК-1.5) 4. Строки в стиле С. Понятие строки. Инициализация строк. Функции для работы со строками. Примеры. (УК-1.4)	
ОПК-1 Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности <i>ОПК-1.1 Осуществляет отбор теоретического и практического материала</i> <i>ОПК-1.2 Решает типовые задачи в рамках профессиональной деятельности</i> <i>ОПК-1.3 Использует различные методы и приемы решения задач профессиональной деятельности</i>	1. Парадигмы программирования, их особенности. Структурное программирование. Процедурное программирование. Модульное программирование. Объектно-ориентированное программирование. Функциональное программирование. Логическое программирование. Обобщенное программирование. (ОПК-1.1) 2. Типы данных в языке C++. Размеры данных. Примеры. Переменные. Константы. Примеры. Явное и неявное преобразование типов. Объявление переменных и констант в языке Си. Область видимости переменных и время жизни объектов. Локальные и глобальные переменные. Переменные, классы памяти. Статическое, автоматическое и динамическое размещение данных. Особенности и различия. (ОПК-1.1) 3. Спроектируйте и реализуйте класс Data.	• <i>Имеется полный анализ задачи, предложен верный алгоритм решения задачи, приведено полное верное решение задачи, включающее правильный ответ: 12 – 14 баллов</i> • <i>Имеется полный анализ задачи, предложен верный алгоритм решения задачи, приведено решение задачи, но получен неправильный ответ из-за арифметической / решение недостаточно обосновано / в решении имеются лишние или неверные записи, не отделенные от решения – 9 – 11 балла</i> • <i>Неполный анализ задачи, предложен верный алгоритм решения задачи, но имеется верное решение лишь части задачи из-за логической ошибки – 5 – 8 баллов</i> • <i>Не соответствует требованиям, изложенным выше – 0 - 4 баллов</i>

	(ОПК-1.2, ОПК-1.3) 4. Строки в стиле С. Понятие строки. Инициализация строк. Функции для работы со строками. Примеры. (ОПК-1.1, ОПК-1.2, ОПК-1.3)	
ОПК-4 Способен решать задачи профессиональной деятельности с использованием существующих информационно-коммуникационных технологий и с учетом основных требований информационной безопасности ОПК-4.1 Отбирает теоретический материал в области ИКТ для решения профессиональных задач ОПК-4.2 Применяет современные методы сбора и представления данных ОПК-4.3 Учитывает требования информационной безопасности при решении профессиональных задач	1. Парадигмы программирования, их особенности. Структурное программирование. Процедурное программирование. Модульное программирование. Объектно-ориентированное программирование. Функциональное программирование. Логическое программирование. Обобщенное программирование. (ОПК-4.2) 2. Типы данных в языке С++. Размеры данных. Примеры. Переменные. Константы. Примеры. Явное и неявное преобразование типов. Объявление переменных и констант в языке Си. Область видимости переменных и время жизни объектов. Локальные и глобальные переменные. Переменные, классы памяти. Статическое, автоматическое и динамическое размещение данных. Особенности и различия. (ОПК-4.1, ОПК-4.2) 3. Спроектируйте и реализуйте класс Data. (ОПК-4.1, ОПК-4.3) 4. Строки в стиле С. Понятие строки. Инициализация строк. Функции для работы со	• Полно и правильно даны ответы на все поставленные вопросы, приведены необходимые примеры; студент показывает понимание излагаемого материала – 12 – 14 баллов • Полно и правильно даны ответы на все поставленные вопросы, приведены примеры, однако имеются неточности; в целом студент показывает понимание изученного материала – 8 – 11 балла • Ответ дан в основном правильно, но недостаточно аргументированы выводы, приведены не все необходимые примеры – 4 – 7 баллов • Даны неверные ответы на поставленные вопросы – 0 - 3 баллов

	строками. Примеры.	
ОПК-5 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения <i>ОПК-5.1 Анализирует существующие алгоритмы и методы программирования для решения типовых задач прикладного характера</i> <i>ОПК-5.2 Применяет известные алгоритмы и методы программирования для решения типовых задач прикладного характера</i> <i>ОПК-5.3 Использует различные методы и приемы, разрабатывает алгоритмы и компьютерные программы, пригодные для практического применения</i>	1. Парадигмы программирования, их особенности. Структурное программирование. Процедурное программирование. Модульное программирование. Объектно-ориентированное программирование. Функциональное программирование. Логическое программирование. Обобщенное программирование. (ОПК-4.2) 2. Типы данных в языке C++. Размеры данных. Примеры. Переменные. Константы. Примеры. Явное и неявное преобразование типов. Объявление переменных и констант в языке Си. Область видимости переменных и время жизни объектов. Локальные и глобальные переменные. Переменные, классы памяти. Статическое, автоматическое и динамическое размещение данных. Особенности и различия. (ОПК-4.1, ОПК-4.2) 3. Спроектируйте и реализуйте класс Data. (ОПК-4.1, ОПК-4.3) 4. Строки в стиле C. Понятие строки. Инициализация строк. Функции для работы со строками. Примеры.	• Полно и правильно даны ответы на все поставленные вопросы, приведены необходимые примеры; студент показывает понимание излагаемого материала – 12 – 14 баллов • Полно и правильно даны ответы на все поставленные вопросы, приведены примеры, однако имеются неточности; в целом студент показывает понимание изученного материала – 8 – 11 балла • Ответ дан в основном правильно, но недостаточно аргументированы выводы, приведены не все необходимые примеры – 4 – 7 баллов • Даны неверные ответы на поставленные вопросы – 0 - 3 баллов

V. Учебно-методическое и информационное обеспечение дисциплины

1) Рекомендуемая литература

а) Основная литература:

1. Павловская, Т. А. Программирование на языке высокого уровня С# : учебное пособие / Т. А. Павловская. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 245 с. — ISBN 978-5-4497-0862-5. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/102051.html>
2. Объектно-ориентированное программирование на С++ : учебник / И. В. Баранова, С. Н. Баранов, И. В. Баженова [и др.]. — Красноярск : СФУ, 2019. — 288 с. — ISBN 978-5-7638-4034-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/157572>
3. Зоткин, С. П. Программирование на языке высокого уровня С/С++ : конспект лекций / С. П. Зоткин. — 3-е изд. — Москва : МИСИ-МГСУ, ЭБС АСВ, 2018. — 140 с. — ISBN 978-5-7264-1810-0. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/76390.html>

б) Дополнительная литература:

- 1.Скворцова, Л. А. Объектно-ориентированное программирование на языке С++ : учебное пособие / Л. А. Скворцова. — Москва : РТУ МИРЭА, 2020. — 246 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/163862>

2) Программное обеспечение

Google Chrome	бесплатное ПО
Яндекс Браузер	бесплатное ПО
Kaspersky Endpoint Security 10	акт на передачу прав ПК545 от 16.12.2022
Многофункциональный редактор ONLYOFFICE	бесплатное ПО
ОС Linux Ubuntu	бесплатное ПО

3) Современные профессиональные базы данных и информационные справочные системы

№ п/п	Вид информационного ресурса, наименование информационного ресурса	Адрес (URL)
1	ЭБС «ZNANIUM.COM»	https://znanium.com/
2	ЭБС «ЮРАИТ»	https://urait.ru/
3	ЭБС «Университетская библиотека онлайн»	https://biblioclub.ru/

4	ЭБС IPR SMART	http://www.iprbookshop.ru/
5	ЭБС «ЛАНЬ»	http://e.lanbook.com
6	ЭБС ТвГУ	http://megapro.tversu.ru/megapro/Web
7	Репозиторий ТвГУ	http://eprints.tversu.ru
8	Ресурсы издательства Springer Nature	http://link.springer.com/
9	СПС КонсультантПлюс (в сети ТвГУ)	

VI. Методические материалы для обучающихся по освоению дисциплины

Учебная программа

Раздел 1. СТРУКТУРНОЕ ПРОГРАММИРОВАНИЕ

Тема 1.1. ОБЩИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ И ИСПОЛЬЗОВАНИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Эволюция архитектуры программного обеспечения. Теоретические проблемы разработки языков программирования. История языков программирования. Общие принципы построения и использования языков программирования. Синтаксис и семантика языка программирования. Средства описания данных. Средства описания действий. Стандарты языков программирования. С++ и поддержка основных парадигм программирования – императивной, модульной, объектно-ориентированной и обобщенной. Обзор языков программирования С и С++. Структура программы на С и С++. Библиотеки программ и классов. Стандартные библиотеки С и С++. Современные интегрированные среды разработки программ. Интегрированная среда языка С++. Команды меню оболочки. С++. Установка режимов работы в среде С++. Графический интерфейс пользователя. Компиляция программы. Макрообработка программы. Препроцессор, директивы препроцессора (макросы). Использование макросов.

Тема 1.2. ЛЕКСИЧЕСКИЕ ОСНОВЫ С++. ТИПЫ И ОБЪЯВЛЕНИЯ

Алфавит и лексемы. Идентификаторы и ключевые слова. Пространство имен. Операторы и операции. Разделители. Ввод и вывод в С и С++. Математические вычисления в С и С++.

Средства описания данных. Фундаментальные типы – логический тип, символьные типы, целые типы, типы с плавающей точкой. Тип void. Типы, определяемые пользователем. Указатели. Массивы. Ссылки. Объявления и определения. Имена. Константы. Инициализация. Классы памяти, классификация переменных по классам памяти.

Тема 1.3. ОПЕРАТОРЫ И ИНСТРУКЦИИ С++

Средства описания действий. Обзор стандартных операторов. Инструкции языка С / С++. Оператор-выражение. Блоки, составной оператор. Управляющие операторы. Операторы цикла. Примеры и особенности использования. Составные типы языка С/С++: Массивы, Структуры (struct), Объединения (union), Перечисляемые типы (enum), Битовые поля. Оператор typedef.

Тема 1.4. УКАЗАТЕЛИ И МАССИВЫ

Память. Описание данных как инструмент распределения памяти и контроля обработки данных. Указатели и адресная арифметика. Классификация указателей. Указатели на объекты. Операции над указателями. Массивы и указатели. Указатели и константы. Указатели на функции. Отладчики. Отладчик интегрированной системы C++. Строки. Строковая константа, ввод строк с клавиатуры, стандартные функции для строк. Массивы строк, массив указателей.

Тема 1.5. ФУНКЦИИ

Оформление программы в виде функции. Объявления и определения функций. Имя функции, список формальных параметров функции, тело функции, область действия функции. Вызовы функций. Аргументы функции. Передача факт параметров в функцию, передача массивов в функцию. Аргументы по умолчанию. Указатели на функцию. Рекурсивные функции. Встраивание функций. Перегрузка функций.

Тема 1.6. ФАЙЛЫ

Файлы прямого доступа, режим работы в файлах прямого доступа. Файлы последовательного доступа, операции доступа к элементам файла. Поточковые файлы, ввод-вывод в файловых потоках. Обработка файлов. Директивы форматного ввода-вывода.

Тема 1.7. ОБРАБОТКА ИСКЛЮЧЕНИЙ

Обработка исключений. Обнаружение ошибок и обработка сбоев работы программ. Операторы try, catch, throw как средство обработки исключительных ситуаций.

Раздел 2. ОСНОВЫ ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ НА C++

Тема 2.1. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ И C++

Архитектура системы – структуры классов и объектов системы. Объектно-ориентированное программирование. Эволюция объектной модели. Объектно-ориентированная декомпозиция. Принципы объектной модели – абстрагирование, инкапсуляция, модульность, иерархичность, типизация. Абстрактные типы данных: инкапсуляция, спецификация, реализация, параметризация, классы и объекты. Принципы реализации абстрактных типов данных – инкапсуляция, наследование и полиморфизм.

Тема 2.2. КЛАСС – АБСТРАКТНЫЙ ТИП ДАННЫХ

Класс как расширение понятия структуры. Область действия класса и доступ к членам класса. Отделение интерфейса от реализации. Управление доступом к членам класса. Конструкторы, деструкторы. Спецификаторы доступа – собственный (закрытый), общедоступный (открытый) и защищенный. Компонентные данные и компонентные функции. Статические компоненты класса. Указатели на компоненты класса. Определение компонентных функций. Указатель this. Дружественные компоненты класса.

Тема 2.3. ПЕРЕГРУЗКА ОПЕРАТОРОВ, КОПИРОВАНИЕ И ПРЕОБРАЗОВАНИЕ

Общие принципы перегрузки операторов. Операторные функции. Бинарные и унарные операторы. Предопределенный смысл операторов. Операторы и типы, определяемые пользователем. Операторы в пространствах имен. Перегрузка конструктора. Конструктор по умолчанию. Конструктор копирования. Конструктор преобразования. Операторы преобразования. Друзья класса.

Тема 2.4. ПРОИЗВОДНЫЕ КЛАССЫ

Наследование классов. Базовые и производные классы. Конструкторы производных классов. Иерархии классов и объектов. Одиночное наследование. Множественное наследование и виртуальные базовые классы. Полиморфизм времени выполнения (динамический полиморфизм). Виртуальные функции. Абстрактные классы. Иерархии классов и абстрактные классы. Вложенные и локальные классы. Интегрированная среда. Генераторы кода/приложений.

Раздел 3. ОСНОВЫ ОБОБЩЕННОЙ ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ НА C++

Тема 3.1. ШАБЛОНЫ КАК ПОДДЕРЖКА ОБОБЩЕННОГО ПРОГРАММИРОВАНИЯ

Шаблоны. Определение шаблонов функций. Параметры шаблонов функций. Выведение типа параметров шаблона по типам аргументов при вызове функции. Переопределение шаблонов функций. Определение шаблонов классов. Параметры шаблонов классов. Создание объектов по шаблонам. Включение конструкторов в шаблон функции. Параметризация и наследование. Полиморфизм времени компиляции (параметрический полиморфизм).

Тема 3.2. ИСПОЛЬЗОВАНИЕ ПАРАМЕТРИЗОВАННЫХ КЛАССОВ

Ограниченные (защищенные) массивы. Очереди. Стеки. Связные списки. Бинарные деревья.

Тема 3.3. ОРГАНИЗАЦИЯ СТАНДАРТНОЙ БИБЛИОТЕКИ C++

Основные концепции – контейнеры, итераторы и алгоритмы. Фундаментальные последовательности – вектора, списки, очереди с двумя концами (деки). Обзор операций с последовательностями. Адаптеры последовательностей – стеки, очереди, очереди с приоритетом. Ассоциативные контейнеры. Обзор операций с ассоциативными контейнерами. Алгоритмы и объекты-функции. Библиотеки программ и классов. Обзор алгоритмов стандартной библиотеки. Итераторы и распределители памяти.

Задания для самопроверки

1. Работа со структурами

- 1 - Ввод информации о рабочих.
- 2 - Просмотр информации о рабочих.
- 3 - Редактирование базы (добавление, удаление записей)
- 4 - Сохранение базы в файл. (два варианта: текстовый файл, двоичный файл)
- 5 - Загрузка базы из файла. (два варианта: текстовый файл, двоичный файл)
- 6 - Поиск рабочих по 3 критериям на выбор (фамилия, имя; возраст; дата устройства на работу)
- 7 - Сортировка по 4 критериям. (id; фамилия и имя; возраст; дата устройства на работу; ...)
- 8 - Обработка записей. (2 критерия: изменить оклад, должность)
- 9 - Выход из системы.

Дополнительно: + 4 балла переделать массив на однонаправленный список

- можно придерживаться данного шаблона
- меню изменить и добавить подменю
- добавлять свои функции, избегать повтора кода
- функции стараться делать как можно короче, разбивать на несколько
- имена функций и переменных на понятном английском

2. STL

1. Создать и протестировать функцию
`void parseString(const string &srcString, vector <set <string> > &result),`
 которая формирует массив множеств строк `result` из строки `srcString` по следующему правилу: `result[i]` – это множество слов одинаковой длины из строки `srcString`.

Пример:

```
srcString = "the quick brown fox jumps over the lazy dog"
result = {{ "quick", "brown", "jumps"}, {"over", "lazy"}, {"the", "fox", "dog" } }
```

2. Создать и протестировать функцию
`void parseString(const string &srcString, map <int, vector<string> > &result),`
 которая формирует `map` векторов строк `result` из строки `srcString` по следующему правилу: `result` – это `map`, где ключ число, означающее кол-во букв, а данные - слова одинаковой длины из строки `srcString`.

Пример:

```
srcString = "the quick brown fox jumps over the lazy dog"
result = { 5 - { "quick", "brown", "jumps" } , 4 - { "over", "lazy" },
3 - { "the", "fox", "dog" } }
```

3. Создать и протестировать функцию
`void parseString(const string &src, vector <set <string> > &result),`
 которая формирует массив множеств строк `result` из строки `src` по следующему правилу: `result[i]` – это множество слов из строки `src`, начинающихся на одну букву; множества отсортированы в алфавитном порядке букв, с которых начинаются их слова.

Пример:

```
src="this is the malt that lay in the house that jack built"
result = {{ "built"}, { "house"}, { "is", "in"}, { "jack"}, { "lay"}, { "malt"},
{ "this", "the", "that" } }
```

4. Создать и протестировать функцию
`void parseString(const string &src, map <char, vector<string> > &result),`
 которая формирует `map` векторов строк `result` из строки `src` по следующему правилу: `result` – это `map`, где ключ буква, а данные - слова начинающихся на эту букву; вектора отсортированы в алфавитном порядке.

Пример:

```
src= "this is the malt that lay in the house that jack built"
result = { b - { "built"}, h - { "house"}, i - { "is", "in"}, j - { "jack"}, l - { "lay"},
m - { "malt"}, t - { "this", "the", "that" } }
```

2. ООП

1. найти ошибку

```
#include <iostream>
using namespace std;

void f(int *, int *n);
void print(int *,int n);
void full(int*, int);

void main(){
    int n = 3;

    int * p = new int [n];
    full(p, n);
    print(p, n);

    f(p,&n);
    print(p, n);
}

void f(int * p, int *n){
    delete []p;
    *n = 5;
    p = new int [*n];
    full(p,*n);
}

void full(int*p, int n){
    for (int i= 0; i< n; i++)
        p[i] = n*i;
}

void print(int * p,int n){
    for (int i= 0; i< n; i++)
        cout<< p[i]<<" ";
    cout<< endl;
}
```

2. Симуляция: черепаха и заяц. В этой задаче вы воссоздадите один из поистине великих моментов в истории, а именно классическую гонку черепахи и зайца. Вы будете использовать генерацию случайных чисел для разработки симуляции этого запоминающегося события. Наши соперники начинают гонку на «квадрате 1» из 70 квадратов. Каждый квадрат представляет собой возможную позицию вдоль трассы. Финишная черта находится на квадрате 70. Соперник, который первым достигнет квадрат 70, награждается ведром свежей моркови и салата. Дорога ведёт свой путь вверх по склону скользкой горы, поэтому иногда соперники теряют позиции, скользя вниз. Есть часы, которые тикают раз в секунду. С каждым тиком часов ваша программа должна корректировать положение животных в соответствии с правилами.

3. Составить описание класса для представления даты. Предусмотреть возможности установки даты и изменения её отдельных полей (год, месяц, день) с проверкой допустимости вводимых значений. В случае недопустимых значений полей выбрасываются исключения. Создать методы изменения даты на заданное количество дней, месяцев и лет. Написать программу, демонстрирующую все разработанные элементы класса.

4. Составить описание класса для представления времени. Предусмотреть возможности установки времени и изменения его отдельных полей (час, минута, секунда) с проверкой допустимости вводимых значений. В случае недопустимых значений полей выбрасываются исключения. Создать методы

изменения времени на заданное количество часов, минут и секунд. Написать программу, демонстрирующую все разработанные элементы класса.

5. Описать структуру с именем TRAIN, содержащую следующие поля: название пункта назначения; номер поезда; время отправления. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в массив, состоящий из восьми элементов типа TRAIN (записи должны быть размещены в алфавитном порядке по названиям пунктов назначения); вывод на экран информации о поездах, отправляющихся после введённого с клавиатуры времени (если таких поездов нет, вывести соответствующее сообщение).

6. Описать структуру с именем NOTE, содержащую следующие поля: фамилия, имя; номер телефона; дата рождения (массив из трёх чисел). Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в массив, состоящий из восьми элементов типа NOTE (записи должны быть упорядочены по дате рождения); вывод на экран информации о человеке, номер телефона которого введён с клавиатуры (если такого нет, вывести соответствующее сообщение).

Список вопросов к зачёту (1 курс, 2 семестр)

1. Типы алгоритмов. Способы описания алгоритмов.
2. Базовые операции и базовые конструкции.
3. Структурное программирование.
4. Нисходящее и восходящее проектирование алгоритмов.
5. Структура программы на языке Си.
6. Выражения в языке Си.
7. Определение алгоритма и программы.
8. Простые типы данных языка Си.
9. Преобразование типов в языке Си.
10. Описание массивов на языке Си.
11. Описание функций на языке Си.
12. Стандартные функции для случайных генерации чисел.
13. Стандартные функции языка Си, предназначенные для обработки строк.
14. Стандартные функции языка Си для определения текущего времени и даты.
15. Составные типы данных в языке Си.
16. Указатели в языке Си.
17. Структуры данных в языке Си.
18. Производные типы: ссылки и указатели.
19. Виды указателей. Инициализация указателей. Адресная арифметика. Операции с указателями.
20. Динамическое распределение памяти. Функции malloc () и free ().
21. Динамическое распределение памяти.

22. Выделение и освобождение динамической памяти. Операции new и delete.
23. Утечка памяти
24. Примеры работы с массивами. Динамические массивы.
25. Связь массивов с указателями. Различия между массивами и указателями. Многомерные массивы. Динамические многомерные массивы.
26. Ввод и вывод массивов.
27. Основные приемы и методы обработки массивов.
28. Способы поиска данных в массиве.
29. Способы сортировки данных в массиве.
30. Массивы указателей, указатели на указатели, массив указателей на функции, сложные объявления.
31. Указатель на функцию. Callback (функция обратного вызова) в [программировании](#). Примеры использования в C/C++.
32. Строки в стиле C. Понятие строки. Инициализация строк. Функции для работы со строками.
33. Понятие функции. Средства организации подпрограмм и функций в языке Си. Локальные и статические переменные функций. Аргументы по умолчанию. Особенности передачи параметров в функцию. Вызов функций по значению. Вызов функций по ссылке и по указателю. Возврат из функций.
34. Перегрузка функций..
35. Понятие потока и файла. Двоичные и текстовые потоки. Работа с потоками средствами языка C. Функции для работы с файлами.
36. Понятие потока и файла. Двоичные и текстовые потоки. Работа с потоками средствами языка C++. Функции для работы с файлами.
37. Понятие и работа со списком. Однонаправленный и двунаправленный списки.
38. Понятие и работа с двоичным деревом.
39. Понятие и работа со стеком.
40. Понятие и работа с очередью

ВОПРОСЫ К ЭКЗАМЕНУ (2 курс, 4 семестр)

1 вопрос

1. Парадигмы программирования, их особенности. Структурное программирование. Процедурное программирование. Модульное программирование. Объектно-ориентированное программирование. Функциональное программирование. Логическое программирование. Обобщенное программирование.
2. Типы данных в языке C++. Размеры данных. Примеры. Переменные. Константы. Примеры. Явное и неявное преобразование типов. Объявление переменных и констант в языке Си. Область видимости переменных и время жизни объектов. Локальные и глобальные переменные. Переменные,

- классы памяти. Статическое, автоматическое и динамическое размещение данных. Особенности и различия.
3. Унарные, бинарные, тернарная операции в языке C++. Краткая характеристика. Операторы. приоритет и очередность вычислений (из таблицы). Примеры
 4. Вывод на экран и ввод с клавиатуры в языке C. Вывод на экран и ввод с клавиатуры в языке C++. Ввод и вывод данных (использование функций и потоковых классов). Примеры
 5. Типы данных Структуры в C++: объявление, инициализация, доступ к полям, функции-члены структуры. Структуры. Работа со структурами. Примеры
 6. Производные типы: ссылки и указатели. Ссылочный тип в C++. Понятие указателя. Виды указателей. Инициализация указателей. Адресная арифметика. Операции с указателями. Примеры.
 7. Динамическое распределение памяти. Функции `malloc ()` и `free ()`.
 8. Динамическое распределение памяти. Выделение и освобождение динамической памяти. Операции `new` и `delete`. Утечка памяти
 9. Массивы. Примеры работы с массивами. Динамические массивы. Связь массивов с указателями. Различия между массивами и указателями. Многомерные массивы. Динамические многомерные массивы. Примеры.
 10. Массивы, массивы указателей, указатели на указатели, массив указателей на функции, сложные объявления.
 11. Строки в стиле C. Понятие строки. Инициализация строк. Функции для работы со строками. Примеры.
 12. Понятие функции. Средства организации подпрограмм и функций в языке Си. Локальные и статические переменные функций. Аргументы по умолчанию. Особенности передачи параметров в функцию. Вызов функций по значению. Вызов функций по ссылке и по указателю. Возврат из функций. Перегрузка функций. Примеры.
 13. Понятие потока и файла. Двоичные и текстовые потоки. Работа с потоками средствами языка C. Функции для работы с файлами.
 14. Понятие потока и файла. Двоичные и текстовые потоки. Работа с потоками средствами языка C++. Функции для работы с файлами.
 15. Понятие и работа со списком. Однонаправленный и двунаправленный списки.
 16. Понятие и работа с двоичным деревом.
 17. Понятие и работа со стеком.
 18. Понятие и работа с очередью
 19. Указатель на функцию. Понятие и назначение функции обратного вызова. Callback в [программировании](#). Примеры использования в C/C++.
 20. Стандартная библиотека шаблонов STL: основные концепции. STL: последовательные и ассоциативные контейнеры. Итераторы. Алгоритмы.
 21. STL - Контейнеры STL вектор.
 22. STL - Контейнеры STL двусвязный список.
 23. STL - Контейнеры STL множество, карта.

24. STL - Контейнеры STL string.
25. STL - Контейнеры STL дек.
26. STL - Контейнеры STL. Стек.
27. STL - Контейнеры STL. Очередь. Очередь с приоритетом.
28. STL - Итераторы. Алгоритмы STL.
29. Основные концепции ООП. Понятия класса и объекта. Абстракция. Инкапсуляция. Наследование. Полиморфизм. Методы, данные и свойства. Ограничение доступа к полям классам.
30. Классы. Методы и поля (данные) классов. Объявление класса в C++.
31. Разграничение доступа к полям и методам класса (спецификаторы доступа). Интерфейс и реализация класса.
32. Классы: функции-члены. Конструкторы и деструкторы. Перегрузка конструкторов. (Конструктор по умолчанию. Конструктор копирования. Конструктор с аргументами)
33. Классы: Указатель this.
34. Друзья класса. Дружественные классы и функции.
35. Статические данные и методы. Особенности. Область применения. Пример.
36. Взаимодействия между объектами классов. Композиция. Агрегация. Ассоциация.

2 вопрос

1. Обработка одномерных и многомерных массивов (память под массивы выделять динамически, ввод, вывод и обработку элементов массива реализовать в виде функции, в функцию передавать указатель на массив).
 - a. Динамические массивы: создание, работа и удаление одномерного динамического массива.
 - b. Динамические массивы: создание, работа и удаление многомерного динамического массива.
 - c. Удаление элементов из динамического массива и вставка элементов в массив.
2. Обработка символьных данных (строк) – найти что-нибудь в строке, преобразовать строку, получить новую строку из исходной и т.п. Для обработки строк реализовать функцию, в функцию передать указатель на строку.
3. Работа со структурами (Объявить структуру, создать массив элементов типа структура, заполнить его данными, найти что-нибудь в этом массиве). Все делать через функции.
4. Работа со структурами (Объявить структуру, создать массив элементов типа структура, заполнить его данными, изменить порядок следования элементов, отсортировать, выбрать некоторые записи по условию). Все делать через функции.
5. Работа со структурами (Объявить структуру, создать массив элементов типа структура, заполнить его данными, найти что-нибудь в этом массиве,

- изменить порядок следования элементов, отсортировать, выбрать некоторые записи по условию). Все делать через функции.
6. Работа со списком. Примеры использования.
 7. Работа с двоичным деревом. Примеры.
 8. Работа со стеком
 - a. Организация стека.
 - b. Примеры использования стека.
 9. STL
 - a. STL вектор. Примеры использования
 - b. STL двусвязный список. Примеры использования
 - c. STL дек. Примеры использования
 - d. STL множество. Примеры использования
 - e. STL map. Примеры использования
 - f. STL string. Примеры использования
 - g. STL Стек. Примеры использования
 - h. STL Очередь. Очередь с приоритетом. Примеры использования
 10. Работа с файлами: файловые потоки, обработка текстовых файлов. Примеры
 11. Работа с файлами: файловые потоки, обработка двоичных файлов. Примеры
 12. ООП. Классы
 - a. Реализуйте класс String для работы со строками символов.
 - b. Спроектируйте и реализуйте класс дата.
 - c. Спроектируйте и реализуйте класс complex..
 - d. Спроектируйте и реализуйте класс rational.
 13. Тесты

Методические указания для обучающихся по освоению дисциплины

Организуя свою учебную работу, студенты должны:

Во-первых, выявить рекомендуемый режим и характер учебной работы по изучению теоретического курса, практическому применению изученного материала, по выполнению заданий для самостоятельной работы, по использованию информационных технологий и т.д.

Во-вторых, ознакомиться с указанным в методическом материале по дисциплине перечнем учебно-методических изданий, рекомендуемых студентам для подготовки к занятиям и выполнения самостоятельной работы, а также с методическими материалами на бумажных и/или электронных носителях, выпущенных кафедрой своими силами и предоставляемые студентам во время занятий.

Самостоятельная работа студентов, предусмотренная учебным планом должна соответствовать более глубокому усвоению изучаемого курса, формировать навыки исследовательской работы и ориентировать студентов на умение применять теоретические знания на практике.

1. Работа с учебными пособиями. Для полноценного усвоения курса студент должен, прежде всего, овладеть основными понятиями этой

дисциплины. Необходимо усвоить определения и понятия, уметь приводить их точные формулировки, приводить примеры объектов, удовлетворяющих этому определению. Кроме того, необходимо знать круг фактов, связанных с данным понятием. Требуется также знать связи между понятиями, уметь устанавливать соотношения между классами объектов, описываемых различными понятиями.

2. Самостоятельное изучение тем. Самостоятельная работа студента является важным видом деятельности, позволяющим хорошо усвоить изучаемый предмет и одним из условий достижения необходимого качества подготовки и профессиональной переподготовки специалистов. Она предполагает самостоятельное изучение студентом рекомендованной учебно-методической литературы, различных справочных материалов, написание рефератов, выступление с докладом, подготовку к лекционным и практическим занятиям, подготовку к зачёту и экзамену.

3. Подготовка к практическим занятиям. При подготовке к практическим занятиям студентам рекомендуется следовать методическим рекомендациям по работе с учебными пособиями, приведенным выше.

4. Составление глоссария. В глоссарий должны быть включены основные понятия, которые студенты изучают в ходе самостоятельной работы. Для полноты исследования рекомендуется вписывать в глоссарий и те термины, которые студентам будут раскрыты в ходе лекционных занятий.

5. Составление конспектов. В конспекте отражены основные понятия темы. Для наглядности и удобства запоминания использованы схемы и таблицы.

6. Подготовка к зачету и экзамену. При подготовке к зачету и экзамену студенты должны использовать как самостоятельно подготовленные конспекты, так и материалы, полученные в ходе занятий.

Качество усвоения студентом каждой дисциплины оценивается по 100-балльной шкале.

Интегральная рейтинговая оценка (балл) по каждому модулю (периоду обучения) складывается из оценки текущей работы обучающихся на занятиях семинарского типа (семинары, практические занятия, практикумы, лабораторные работы, коллоквиумы и иные аналогичные занятия), оценки индивидуальной работы обучающихся и оценки за выполнение заданий рейтингового контроля успеваемости. При этом доля баллов, выделенных на рейтинговый контроль не должна превышать 50% общей суммы баллов данного модуля (периода обучения).

Максимальная сумма рейтинговых баллов по учебной дисциплине, заканчивающейся экзаменом, по итогам семестра составляет 60.

Обучающемуся, набравшему 40-54 балла, при подведении итогов семестра (на последнем занятии по дисциплине) в рейтинговой ведомости учета успеваемости и зачетной книжке может быть выставлена оценка «удовлетворительно».

Обучающемуся, набравшему 55-57 баллов, при подведении итогов семестра (на последнем занятии по дисциплине) в графе рейтинговой

ведомости учета успеваемости «Премияльные баллы» может быть добавлено 15 баллов и выставлена экзаменационная оценка «хорошо».

Обучающемуся, набравшему 58-60 баллов, при подведении итогов семестра (на последнем занятии по дисциплине) в графе рейтинговой ведомости учета успеваемости «Премияльные баллы» может быть добавлено 27 баллов и выставлена экзаменационная оценка «отлично».

В каких-либо иных случаях добавление премиальных баллов не допускается.

Обучающийся, набравший до 39 баллов включительно, сдает экзамен. При наличии подтвержденных документально уважительных причин, по которым были пропущены занятия (длительная болезнь, обучение в другом вузе в рамках академической мобильности и др.), обучающийся имеет право отработать пропущенные занятия и получить дополнительные баллы в рамках установленных баллов за модуль. Сроки и порядок отработки определяет преподаватель. Баллы выставляются в графе «отработка».

Ответ обучающегося на экзамене оценивается суммой до 40 рейтинговых баллов. Итоговая оценка складывается из суммы баллов, полученных за семестр, и баллов, полученных на экзамене. Обучающемуся, который сдает экзамен, премиальные баллы не начисляются.

Максимальная сумма баллов по учебной дисциплине, заканчивающейся зачетом, по итогам семестра составляет 100 баллов (50 баллов – 1-й модуль и 50 баллов – 2-й модуль).

Студенту, набравший 40 баллов и выше по итогам работы в семестре, в экзаменационной ведомости и зачетной книжке выставляется оценка «зачтено». Студенту, набравшему до 39 баллов включительно, сдает зачет.

Согласно подходам балльно-рейтинговой системы в рамках оценки знаний, умений, владений (умений применять) и (или) опыта деятельности дисциплины установлены следующие аспекты:

- Содержание учебной дисциплины в рамках одного семестра делится на два модуля (периода обучения). По окончании модуля (периода обучения) осуществляется рейтинговый контроль успеваемости знаний студентов.
- Сроки проведения рейтингового контроля:
 - осенний семестр* – I рейтинговый контроль успеваемости проводится согласно графику учебного процесса, II рейтинговый контроль успеваемости - две последние недели фактического завершения семестра по графику учебного процесса;
 - весенний семестр* – I рейтинговый контроль успеваемости проводится согласно графику учебного процесса, II рейтинговый контроль успеваемости - две последние недели фактического завершения семестра по графику учебного процесса.

Методические указания для самостоятельной работы студентов

Самостоятельная работа студентов является неотъемлемой частью изучения дисциплины. Кроме того, в темах, изучаемых при контактной

работе со студентами, есть отдельные учебные вопросы, которые студенты должны изучить самостоятельно. Контроль знаний при самостоятельном изучении тем и вопросов дисциплины осуществляется при проведении текущего контроля в виде устных опросов, письменных контрольных работ и тестирования во время рейтинг-контроля. Вопросы для самостоятельной работы также включаются в темы рефератов, которые студенты защищают на семинарских занятиях, и в перечень вопросов для экзамена.

Записав лекцию или составив ее конспект, не следует оставлять работу над лекционным материалом до начала подготовки к экзамену. Нужно проделать как можно раньше ту работу, которая сопровождает конспектирование письменных источников и которую не удалось сделать во время записи лекции: прочесть свои записи, расшифровав отдельные сокращения, проанализировать текст, установить логические связи между его элементами, в ряде случаев показать их графически, выделить главные мысли, отметить вопросы, требующие дополнительной обработки, в частности, консультации преподавателя. При работе над текстом лекции студенту необходимо обратить особое внимание на проблемные вопросы, поставленные преподавателем при чтении лекции, а также на его задания и рекомендации. Работая над текстом лекции, необходимо иметь под рукой справочные издания: словарь-справочник, энциклопедический экономический словарь, в которых можно найти объяснение многим встречающимся в тексте терминам, содержание которых студент представляет себе весьма туманно, хотя они ему и знакомы.

В процессе организации самостоятельной работы большое значение имеют консультации с преподавателем, в ходе которых можно решить многие проблемы изучаемого курса, уяснить сложные вопросы.

VII. Материально-техническое обеспечение дисциплины

Наименование специальных* помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы	Перечень лицензионного программного обеспечения. Реквизиты подтверждающего документа
Помещение для самостоятельной работы, учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и	<i>Комплект учебной мебели, компьютер (системный блок, монитор, клавиатура, мышь) 8 шт., коммутатор.</i>	Adobe Acrobat Reader DC - Russian – бесплатно Cadence SPB/OrCAD 16.6 – Государственный контракт на поставку лицензионных программных продуктов 103 - ГК/09 от 15.06.2009 GIMP 2.8.20 – бесплатно Google Chrome – бесплатно Java SE Development Kit 8 Update 45 (64-bit) – бесплатно Kaspersky Endpoint Security 10 для

промежуточной аттестации, практики, компьютерный класс математического факультета: № 21 (170002 Тверская обл., г. Тверь, пер. Садовый, д. 35) Учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и	Комплект учебной мебели, CD-магнитола, компьютер: (системный блок + монитор), многофункциональный	Windows – Акт на передачу прав ПК545 от 16.12.2022 Lazarus 1.4.0 – бесплатно Mathcad 15 M010 – Акт предоставления прав ИС000000027 от 16.09.2011; MATLAB R2012b – Акт предоставления прав № Us000311 от 25.09.2012; Многофункциональный редактор ONLYOFFICE бесплатное ПО – бесплатно ОС Linux Ubuntu бесплатное ПО – бесплатно Microsoft SQL Server 2014 Express LocalDB – бесплатно Microsoft Visio Professional 2013 – Акт на передачу прав №785 от 06.08.2021 г. Microsoft Visual Studio Ultimate 2013 с обновлением 4 – Акт на передачу прав №785 от 06.08.2021 г. Microsoft Web Deploy 3.5 – бесплатно Microsoft Windows 10 Enterprise – Акт на передачу прав №785 от 06.08.2021 г. MiKTeX 2.9 – бесплатно MSXML 4.0 SP2 Parser and SDK – бесплатно MySQL Workbench 6.3 CE – бесплатно NetBeans IDE 8.0.2 – бесплатно Notepad++ – бесплатно Origin 8.1 Sr2 – договор №13918/M41 от 24.09.2009 с ЗАО «СофтЛайн Трейд»; PostgreSQL – бесплатно Python 3.4.3 – бесплатно Unity Web Player – бесплатно WCF RIA Services V1.0 SP2 – бесплатно WinDjView 2.1 – бесплатно. Google Chrome – бесплатно Kaspersky Endpoint Security 10 для Windows – Акт на передачу прав ПК545 от 16.12.2022 Lazarus – бесплатно OpenOffice – бесплатно Многофункциональный редактор ONLYOFFICE бесплатное ПО – бесплатно
---	--	--

индивидуальных консультаций, текущего контроля и промежуточной аттестации, <i>учебная аудитория: № 208 (170002 Тверская обл., г. Тверь, пер. Садовый, д. 35)</i> Лаборатория компьютерной безопасности кафедры компьютерной безопасности и математических методов управления: № 203а <i>(170002 Тверская обл., г. Тверь, пер. Садовый, д. 35)</i> Помещение для самостоятельной работы, учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, практики, <i>компьютерный класс математического факультета: № 16 (170002 Тверская обл., г.</i>	<i>лазер. копир/принтер/сканер, видеоплеер, телевизор, DVD плеер.</i> Комплект учебной мебели, источник бесперебойного питания 5 шт., коммутатор, компьютер (системный блок, монитор, клавиатура, мышь) 7 шт., копир-принтер-сканер 2 шт., плеер combo, телевизор, экран настенный, учебный лабораторный стенд для изучения основ криптографии, исполнение блочное BCRYPTO, учебно-лабораторный стенд для изучения блочного кодирования ЗИ-БК. <i>Комплект учебной мебели, компьютер (системный блок, монитор, клавиатура, мышь) 8 шт., коммутатор.</i>	ОС Linux Ubuntu бесплатное ПО – бесплатно Google Chrome – бесплатно Kaspersky Endpoint Security 10 для Windows – Акт на передачу прав ПК545 от 16.12.2022 Lazarus – бесплатно OpenOffice – бесплатно Многофункциональный редактор ONLYOFFICE бесплатное ПО – бесплатно ОС Linux Ubuntu бесплатное ПО – бесплатно Adobe Acrobat Reader DC – Russian – бесплатно Cadence SPB/OrCAD 16.6 – Государственный контракт на поставку лицензионных программных продуктов 103 - ГК/09 от 15.06.2009 Google Chrome – бесплатно Java SE Development Kit 8 Update 45 (64-bit) – бесплатно Kaspersky Endpoint Security 10 для Windows – Акт на передачу прав ПК545 от 16.12.2022 Lazarus 1.4.0 – бесплатно Mathcad 15 M010 – Акт предоставления прав ИС000000027 от 16.09.2011; MATLAB R2012b – Акт предоставления прав № Us000311 от
---	--	--

Тверь, пер. Садовый, д. 35)		25.09.2012; Mercurial 3.7.3 бесплатно Microsoft SQL Server 2012 Express LocalDB – бесплатно Microsoft Web Deploy 3.5 – бесплатно MiKTeX 2.9 – бесплатно MSXML 4.0 SP2 Parser and SDK – бесплатно MySQL Workbench 6.3 CE – бесплатно NetBeans IDE 8.0.2 – бесплатно Notepad++ – бесплатно Origin 8.1 Sr2 – договор №13918/M41 от 24.09.2009 с ЗАО «СофтЛайн Трейд»; Python 3.4.3 – бесплатно WinDjView 2.1 – бесплатно WCF RIA Services V1.0 SP2 – бесплатно Многофункциональный редактор ONLYOFFICE – бесплатное ПО бесплатно ОС Linux Ubuntu бесплатное ПО – бесплатно
--------------------------------	--	--

VIII. Сведения об обновлении рабочей программы дисциплины

№ п.п.	Обновленный раздел рабочей программы дисциплины	Описание внесенных изменений	Дата и № протокола заседания кафедры / методического совета факультета, утвердившего изменения
1.			
2.			