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

Утверждаю:
Руководитель ООП
Н.А. Семыкина
«4» 09 20 г.

# Рабочая программа дисциплины (с аннотацией)

**Языки программирования**

Специальность

**10.05.01 Компьютерная безопасность**

Специализация

**«Математические методы защиты информации»**

Для студентов очной формы обучения

СПЕЦИАЛИТЕТ

Для студентов 1, 2, 3 курсов ОФО

Составитель:
Шаповалова И. А.,

Тверь 2023

## I. Аннотация

### 1. Цель и задачи дисциплины

Целью освоения дисциплины (модуля) является:

формирование базы для развития профессиональных компетенций, связанных с готовностью студента к деятельности в области проектирования и разработки информационных моделей, предназначенных для решения различных профессиональных, исследовательских и прикладных задач.

Задачами освоения дисциплины (модуля) являются:

– получение базовых знаний и умений, связанных с разработкой алгоритмов и формирование алгоритмического мышления;

– получение теоретических знаний о роли и назначении различных языков программирования высокого и низкого уровня;

– обучения студентов общим принципам использования языков программирования; средствам описания данных; средствам описания действий; абстрактным типам данных;

– развитие навыков программирования на языках C/C++ с использованием современных интегрированных сред разработки (IDE) и инструментальных средств;

– формирование навыков мышления программиста и создания ПО для решения различных профессиональных, исследовательских и прикладных задач, а также содействовать фундаментализации образования и развитию системного мышления.

### 2. Место дисциплины в структуре ООП

Данная дисциплина входит в обязательную часть учебного плана. Для освоения дисциплины студент должен владеть современными методами и средствами информационных технологий. Необходимы знания, умения и компетенции, полученные обучающимися на занятиях по предмету «Информатика и ИКТ» в средней общеобразовательной школе и необходимы компетенции, сформированные в процессе обучения дисциплин

«Информатика», «Анализ алгоритмов и структур». Дисциплина «Языки программирования» является базовой для изучения дисциплин «Методы программирования», «Объектно-ориентированное программирование», «Параллельное программирование», «Технологии разработки программного обеспечения», «Тестирование программного обеспечения», «Операционные системы», «Системы управления базами данных». Знания и практические навыки, полученные из курса, используются студентами при изучении научных дисциплин, при прохождения производственной и преддипломной практики, а также при разработке курсовых и дипломных работ.

**3. Объем дисциплины:** 18 зачетных единиц, 648 академических часов, **в том числе:**

**контактная аудиторная работа:** лекции 140 часов, в т.ч. практическая подготовка 0 часов, лабораторные занятия 210 часов, в т.ч. практическая подготовка 0 часов*;*

**самостоятельная работа:** 298 часа.

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

**ОПК-7.1; ОПК-7.2; ОПК-7.3; ОПК-13.2**

| Планируемые результаты освоения образовательной программы (формируемые компетенции) | Планируемые результаты обучения по дисциплине |
|---|---|
| **ОПК-7.** Способен создавать программы на языках высокого и низкого уровня, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ; | **ОПК-7.1.** Разрабатывает и применяет на языке высокого уровня алгоритмы решения типовых профессиональных задач; |
| | **ОПК-7.2.** Применяет известные методы программирования и возможности базово го языка программирования для решения типовых профессиональных задач; |
| | **ОПК-7.3.** Использует основные принципы разработки, документирования, тестирования и отладки программ; |

| | |
|---|---|
| **ОПК-13.** Способен разрабатывать компоненты программных и программно-аппаратных средств защиты информации в компьютерных системах и проводить анализ их безопасности; | **ОПК-13.2.** Работает с интегрированными средами разработки программного обеспечения. |

**5. Форма промежуточной аттестации и семестр прохождения** зачет – 2, 4 семестры, экзамен – 3, 5 семестры, курсовая работа – 5 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, в том числе Контроль (час.) |
|---|---|---|---|---|---|---|---|
| | | Лекции | | Практические занятия | | Контроль самостоятельной работы (в том числе курсовая работа) | |
| | | всего | в т.ч. практическая подготовка | всего | в т.ч. практическая подготовка | | |
| Раздел 1. Структурное программирование | | | | | | | |
| Тема 1.1. Общие принципы построения и использования языков программирования | 16 | 2 | | 4 | | | 10 |
| Тема 1.2. Лексические основы C/C++. Типы и объявления | 26 | 4 | | 6 | | | 16 |
| Тема 1.3. Операторы и инструкции C/C++ | 44 | 8 | | 16 | | | 20 |
| Тема 1.4. Указатели и массивы | 50 | 10 | | 20 | | | 20 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Тема 1.5. Функции | 44 | 8 | | 16 | | | 20 |
| Тема 1.6. Файлы | 40 | 8 | | 12 | | | 20 |
| Тема 1.7. Обработка исключений | 38 | 8 | | 10 | | | 20 |
| Тема 1.8. Динамические структуры | 48 | 8 | | 16 | | | 24 |
| Раздел 2. Основы объектно-ориентированной парадигмы Программирования на С++ | | | | | | | |
| Тема 2.1. Объектно-ориентированное программирование и С++ | 34 | 8 | | 10 | | | 16 |
| Тема 2.2. Класс – абстрактный тип данных | 72 | 12 | | 20 | | | 40 |
| Тема 2.3. Перегрузка операторов, копирование и преобразование | 60 | 12 | | 18 | | | 30 |
| Тема 2.4. Производные классы | 48 | 12 | | 16 | | | 20 |
| Раздел 3. Основы обобщенной парадигмы программирования на С++ | 26 | 8 | | 8 | | | 10 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Тема 3.1. Шаблоны как поддержка обобщенного программирования | 26 | 8 | | 12 | | | 6 |
| Тема 3.2. Использование параметризованных классов | 18 | 6 | | 6 | | | 6 |
| Тема 3.3. Организация стандартной библиотеки C++ | 58 | 18 | | 20 | | | 20 |
| **ИТОГО** | 648 | 140 | 0 | 210 | 0 | 0 | 298 |

## III. Образовательные технологии

| Учебная программа – наименование разделов и тем *(в строгом соответствии с разделом II РПД)* | Вид занятия | Образовательные технологии |
|---|---|---|
| Тема 1.1. Общие принципы построения и использования языков программирования | Лекция<br>Лабораторное занятие | Традиционная лекция.<br>Цифровые технологии. |
| Тема 1.2. Лексические основы C/C++. Типы и объявления | Лекция<br>Лабораторное занятие | Традиционная лекция.<br>Цифровые технологии. |
| Тема 1.3. Операторы и инструкции C/C++ | Лекция<br>Лабораторное занятие | Традиционная лекция.<br>Цифровые технологии. |
| Тема 1.4. Указатели и массивы | Лекция<br>Лабораторное занятие | Традиционная лекция.<br>Цифровые технологии. |
| Тема 1.5. Функции | Лекция<br><br><br>Лабораторное занятие | Проблемная лекция, дискуссионные технологии.<br>Проектные технологии, методы группового решения творческих задач, цифровые технологии |

| | | |
|---|---|---|
| Тема 1.6. Файлы | Лекция<br>Лабораторное занятие | Традиционная лекция.<br>Цифровые технологии. |
| Тема 1.7. Обработка исключений | Лекция<br><br><br>Лабораторное занятие | Проблемная лекция, дискуссионные технологии.<br>Проектные технологии, методы группового решения творческих задач, цифровые технологии |
| Тема 1.8. Динамические структуры | Лекция<br>Лабораторное занятие | Традиционная лекция.<br>Проектные технологии, методы группового решения творческих задач, цифровые технологии |
| Тема 2.1. Объектно-ориентированное программирование и C++ | Лекция<br><br><br>Лабораторное занятие | Проблемная лекция, дискуссионные технологии.<br>Цифровые технологии |
| Тема 2.2. Класс – абстрактный тип данных | Лекция<br>Лабораторное занятие | Традиционная лекция.<br>Проектные технологии, методы группового решения творческих задач, цифровые технологии. |
| Тема 2.3. Перегрузка операторов, копирование и преобразование | Лекция<br>Лабораторное занятие | Традиционная лекция.<br>Проектные технологии, методы группового решения творческих задач, цифровые технологии. |
| Тема 2.4. Производные классы | Лекция<br><br><br>Лабораторное занятие | Проблемная лекция, дискуссионные технологии.<br>Проектные технологии, методы группового решения творческих задач, цифровые технологии. |

| | | |
|---|---|---|
| Раздел 3. Основы обобщенной парадигмы программирования на C++ | Лекция<br>Лабораторное занятие | Традиционная лекция.<br>Цифровые технологии. |
| Тема 3.1. Шаблоны как поддержка обобщенного программирования | Лекция<br><br><br>Лабораторное занятие | Проблемная лекция, дискуссионные технологии.<br>Цифровые технологии. |
| Тема 3.2. Использование параметризованных классов | Лекция<br>Лабораторное занятие | Традиционная лекция.<br>Проектные технологии, методы группового решения творческих задач, цифровые технологии |
| Тема 3.3. Организация стандартной библиотеки C++ | Лекция<br>Лабораторное занятие | Традиционная лекция.<br>Цифровые технологии. |

## IV. Оценочные материалы для проведения текущей и промежуточной аттестации

***Оценочные материалы для проведения <u>текущей аттестации</u>***

<u>Задания для лабораторных занятий</u>

**Задание 1 (ОПК-7.3, ОПК-13.2).** Организуйте вывод в таблицу аргумента x, значения суммы функционального ряда и числа членов, вошедших в сумму, а также укажите значение стандартной функции f(x).

Аргумент x изменяется на интервале [a; b] с заданным шагом Δx. В программе предусмотрите контроль вводимых значений a, b, Δx по условиям $b > a$, $\Delta x > 0$.

Организуйте вывод значений аргумента, вычисленных значений функции y(x) и значений соответствующей стандартной функции в виде таблицы:

ТАБЛИЦА ФУНКЦИИ Y(X)

X | Y | количество членов | например, SIN(X)

. . . | . . . | . . . | …

. . . | . . . | . . . | …

Значения параметров – начальное и конечное значения аргумента a, b и шаг Δx выберите сами в соответствии с вашим вариантом функции y(x).

Например для y = sin (x) a = 0, b = 3, Δx = 0,15;

для y = cos(x) a = 0, b = 7, Δx = 0,5.

Для вычисления членов ряда получите и используйте рекуррентные соотношения вида $u_n = k_n \cdot u_{n-1}$, связывающие последовательные элементы ряда. Считайте, что требуемая точность достигнута, если очередное слагаемое по модулю будет меньше ε. Обеспечьте возможность ввода любого желаемого значения ε > 0 (рекомендуемое значение порядка 0,001).

**Задание 2 (ОПК-7.3, ОПК-13.2).** Пусть дано уравнение $f(x) = 0$, где $f(x)$ – функция, определенная и непрерывная на некотором промежутке. В некоторых случаях на функцию $f(x)$ могут быть наложены дополнительные ограничения, например, непрерывность первой и второй производных, что специально оговаривается. Функция $f(x)$ может быть задана в виде алгебраического многочлена или трансцендентной функции.

Требуется найти корни нелинейного уравнения $f(x) = 0$, т.е. числа $x_{*1}, x_{*2}, \ldots$, которые путем подстановки их в уравнение превращают его в верное числовое равенство.

Во всех индивидуальных заданиях необходимо отделить и найти **ВСЕ** действительные корни уравнений с заданной точностью $\varepsilon$ указанным методом (метод хорд, метод простых итераций, метод Ньютона, метод касательных, метод секущих, метод половинного деления). При решении:

- описать функцию $f(x)$ и при необходимости функцию $f'(x)$;
- описать через функцию метод нахождения корня;
- проверить является ли найденное значение корнем уравнения;
- точность вычислений вводить интерактивно с клавиатуры;
- для отделения корней можно использовать сайт

https://www.geogebra.org/calculator

**Задание 3 (ОПК-7.3, ОПК-13.2).** 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, ж потом – все отрицательные в том же порядке, что и в исходном массиве.

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.

**Задание 4 (ОПК-7.1, ОПК-13.2).** Организовать данные в виде массива структур в соответствии с вариантом. В программе реализовать меню:

- ввод данных в массив структур с клавиатуры;
- чтение данных в массив структур из текстового файла;
- запись данных в текстовый файл из массива структур;
- добавление данных в массив структур;
- вставка записи в массив структур на i-ю позицию (нумерация с 1);
- сортировка массива структур по первому полю, входящему в структуру;
- сортировка массива структур по трем первым полям;
- поиск в массиве структур по заданному параметру;
- изменение заданной структуры;
- удаление i-ой структуры из массива (нумерация с 1);
- удаление структур из массива по заданному значению первого поля;
- вывод на экран массива структур.

Решение оформить в виде функций.

Пример данных: структура «Информация»: носитель; объем; название; автор.

**Задание 5 (ОПК-7.1, ОПК-13.2).** Описать класс, одним из свойств которого является динамический объект. В описании класса должны присутствовать конструкторы (в том числе конструктор копирования), деструктор, оператор присваивания, операторы потокового ввода-вывода, дружественные функции.

Для примера: описать класс для эффективной работы со строками, позволяющий форматировать и сравнивать строки, хранить в строках числовые значения и извлекать их. Для этого необходимо реализовать:

• перегруженные операции присваивания и конкатенации;

• операции сравнения и приведения типов;

• преобразование в число любого типа;

• форматный вывод строки.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса. Работу класса продемонстрировать на 5 -10 объектах класса.

**Задание 6 (ОПК-7.1, ОПК-13.2).** 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:

• конструктор без параметров;

• конструктор, принимающий в качестве параметра С-строку (заканчивается пулевым байтом);

• конструктор, принимающий в качестве параметра символ;

• конструктор копирования;

• получение длины строки;

• очистка строки (сделать строку пустой);

• деструктор.

Описать производный от класса СТРОКА класс СТРОКА__ИДЕНТИФИКАТОР.

Строки данного класса строятся по правилам записи идентификаторов в языке С и 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 С-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;

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

• операция (<) – 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 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;

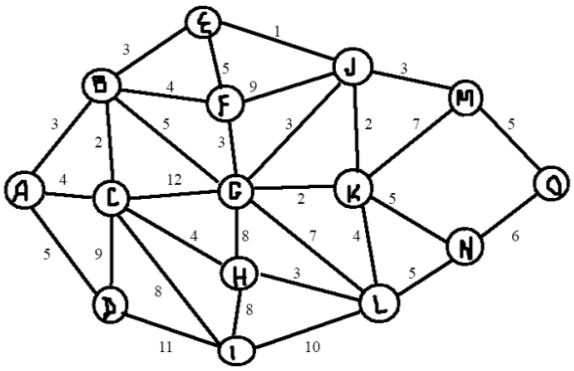
• для созданных экземпляров производных классов выполняет проверку всех разработанных методов с выводом исходных данных и результатов на дисплей.

Для конструкторов копирования каждого класса предусмотреть диагностическую печать количества его вызовов в определенное место дисплея (рекомендуется использовать статические члены класса).

Режим диалога обеспечивается с помощью иерархического меню.

**Задание 7 (ОПК-7.1, ОПК-13.2).** Описать алгоритм и составить программу для шифрования и дешифрования текста методом табличной маршрутной перестановки. Использовать различные варианты маршрута.

**Задание 8 (ОПК-7.1, ОПК-13.2).** С помощью алгоритма Дейкстры найти кратчайшее расстояние между вершиной A и всеми остальными. Заполнить массивы dist и prev




**Задание 9 (ОПК-7.1, ОПК-13.2).** Разбота с типом ***string*** .

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

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

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.

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.

***Оценочные материалы для проведения промежуточной аттестации***

Проверяемые индикаторы достижения компетенций: ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-13.2.

**Примеры заданий для проведения промежуточной аттестаций в форме зачета 1 курс, 2 семестр**

Обучающийся решает практические задания.

Способ проведения промежуточной аттестации: письменное решение задач.

**Пример 1.**

1. Описать функцию Inv(*A*, *N*), меняющую порядок следования элементов вещественного массива *A* размера *N* на обратный (*инвертирование* массива).
2. Описать функцию, которая из неупорядоченного массива A(m) получает упорядоченный по возрастанию массив D(n) (n<<m) следующим образом. Выбрать в A наименьший элемент и поместить его в массив D, затем – наименьший и оставшихся и т.д. Повторяющиеся элементы включать в массив D единожды; массив A сохранить.
3. Дана матрица *A(N, M)* . Описать функцию, формирующую вектор *B(N)* , элементы *Bi* которого равны единице, если элементы *i*-ой строки образуют упорядоченную по убыванию или по возрастанию последовательность, и нулю во всех остальных случаях.

**Пример 2.**

1. Описать функцию RemoveForInc(A, N), удаляющую из вещественного массива A размера N «лишние» элементы так, чтобы оставшиеся элементы оказались упорядоченными по возрастанию: первый элемент не удаляется, второй элемент удаляется, если он меньше первого, третий — если он меньше предыдущего элемента, оставленного в массиве, и т. д.
2. Описать функцию, которая переносит в начало массива A(n) все его отрицательные элементы, затем – нулевые и в конце – все положительные (с сохранением порядка следования в каждой группе).
3. Описать функцию RemoveCols(A, M, N, K1, K2), удаляющую из вещественной матрицы A размера M × N столбцы с номерами от K1 до K2 включительно (предполагается, что $1 < K1 \le K2$). Если K1 > N, то матрица не изменяется; если K2 > N, то удаляются столбцы матрицы с номерами от K1 до N.

**Критерии оценивания и шкала оценивания:**

Максимально возможное количество баллов за каждую задачу – 3 балла. Для получения зачета необходимо набрать не менее 5 балов.

3 балла:

Ответ демонстрирует знание и корректное использование теоретического материала и умение составлять алгоритмы. Имеется полное верное решение задачи, включающее правильный ответ.

2 балла:

Ответ демонстрирует знание и корректное использование теоретического материала и умение составлять алгоритмы. Решение не содержит фактических ошибок. В решении имеются неверные записи И/ИЛИ синтаксические ошибки.

1 балл:

Ответ демонстрирует знание и корректное использование теоретического материала. Решение содержит алгоритмические и синтаксические.

0 баллов:

Решение содержит существенные алгоритмические ошибки, искажающие смысл задания. Решение не дано ИЛИ дано неверное решение.

## Примеры заданий для проведения промежуточной аттестаций в форме экзамена, 2 курс 3 семестр

Обучающийся решает практическое задание и отвечает на теоретический вопрос.

Способ проведения промежуточной аттестации: письменное решение, устное пояснение.

**Пример 1.**

1. Динамическое распределение памяти. Выделение и освобождение динамической памяти. Операции new и delete. Утечка памяти.
2. Работа со структурами (Объявить структуру, создать массив элементов типа структура, заполнить его данными, изменить порядок

следования элементов, отсортировать, выбрать некоторые записи по условию). Все операции описать в виде функций.

**Пример 2.**

1. Подставляемые функции. Перезагрузка функций.
2. Вывести слова строки, начинающиеся и заканчивающиеся одной и той же буквой; которые содержат ровно три буквы «k».

**Критерии оценивания и шкала оценивания:**

Максимально возможное количество баллов – 40 баллов, при этом начисление баллов производится следующим образом:

Самостоятельно выполнено верно 85 - 100 % от всех заданий. Ответ на вопрос демонстрирует знание и корректное использование теоретического материала. Факты и примеры в полном объеме обосновывают выводы. Имеется полное верное решение задачи, включающее правильный ответ – 30 - 40 баллов;

Самостоятельно выполнено верно 75 - 84% от всех заданий. Ответ на вопрос демонстрирует знание и корректное использование теоретического материала. Ответ не содержит фактических ошибок. Дано верное решение задачи, но в решении имеются неверные записи И/ИЛИ арифметические ошибки– 20 - 30 баллов;

Самостоятельно выполнено верно 50 - 74% заданий. Ответ на вопрос демонстрирует знание теоретического материала. Решение содержит фактические ошибки, не искажающие общего смысла. – 10 - 20 баллов;

Выполнено верно менее 50% заданий. В ответе преобладают рассуждения общего характера И/ИЛИ содержит существенные фактические ошибки, искажающие смысл. Решение не дано ИЛИ дано неверное решение – 0 – 10 баллов.

**Примеры заданий для проведения промежуточной аттестаций в форме зачета 2 курс, 4 семестр**

**Тест №1**

Обучающийся отвечает на вопросы теста.

Способ проведения промежуточной аттестации: письменное решение.

I. Задания теоретические (каждое задание оценивается в 1 балл)

1. Что будет выведено на экран в результате вызова функции f1( )?

```
void  f1(int x1, int &x2) {
        if (x1>x2){
                x1=x1+x2;
                x2=x1-x2;
                x1=x1-x2;
        }
}
int main() {
   int a = 7, k = 1;
   f1(a, k);
   cout << a << " " << k;
}
```

2. Опишите прототип функции с именем alter ( ), которая принимает две переменные x и y типа int и присваивает этим переменным, соответственно, значения их суммы и разности.
3. Функция вычисления суммы цифр числа n имеет прототип int func1(int n), запишите определение функции, используя рекурсию:

| | | | |
|---|---|---|---|
| 1 | {if (n > 10) return 1;<br>else return 1 + func(n % 10);} | 3 | {if (n < 10) return n;<br>else return n/10 + func(n % 10);} |
| 2 | {if (n < 10) return n;<br>else return n%10 + func(n/10);} | 4 | {if (n > 10) return n;<br>else return n%10 + func(n/10);} |

4. Типом возвращаемого функцией значения может быть любым, кроме …?

| | | | |
|---|---|---|---|
| 1 | структуры | 3 | массива |
| 2 | указателя на функцию | 4 | типа определенного пользователем |

5. Какой будет результат выполнения следующей программы?

```
int sump ( int *start, int *end ){
   int total = 0 ;
   while ( start < end){
      total += *start ;
      start++ ;
   }
   return total ;
}
int main() {
   #define Bmax 6
   int B[6] = {1, 2, 3, 4, 5, 6};
   std::cout << sump(B + 1, B+5) << std::endl;
}
```

6. Что лучше выбрать: вектор, список или двустороннюю очередь, если из файла считывается неизвестное количество слов, слова добавляются в конец контейнера, а удаляются всегда из начала?
7. Как создать список размера 5 и инициализировать его элементы значением 10?

| | | | |
|---|---|---|---|
| | list <int> spisok.resize(5, 10 ); | 3 | list <int> spisok.insert(5, 10 ); |
| 2 | list <int> spisok (5, 10 ); | | |

8. Алгоритм … копирует элементы из заданного диапазона, удаляя из него элементы, имеющие значение val.

| | |
|---|---|
| 1 | copy_backward() |
| 2 | erase () |
| 3 | remove_copy() |
| 4 | replace_copy() |

9. Что будет выведено на экран?

```
char x[5] = {'1', '2', '3', '4', '5'};
vector<int> Arr (&x[0], &x[5]);
Arr.push_back('A');
Arr. push_front('A');
vector< char >::iterator where = find(Arr.begin(), Arr.end(), '3');
for (;where!=Arr.end(); where++)
    cout<< *where<<endl;
```

10. Напишите фрагмент программы, которая в бинарном файле меняет местами 2-ю и 7-ю записи. Записи имеют пользовательский тип My_Type.
11. Запишите модификаторы потока, открывающие файл для чтения, записи, добавления в конец файла, работы с бинарным файлом
12. Опишите структуру, которая содержит название месяца, трехбуквенную аббревиатуру месяца, количество дней в месяце и номер месяца. Определите массив из 12 структур и инициализируйте первые его 3 элемента для года, отличного от високосного.
13. Что возвращает функция, прототип которой объявлен следующим образом:

```
struct A {char *x; int y;};
A fun(void);
```

| | | | |
|---|---|---|---|
| 1 | структуру как результат | 3 | ссылку на структуру |
| 2 | указатель на структуру | | |

14. Напишите фрагмент программы, которая с помощью функций-членов assign() и append() из строк

```
string quote1( "When lilacs last in the dooryard bloom'd" );
string quote2( "The child "is father of the man" );
составит предложение "The child is in the dooryard"
```

15. Функция strstr ( s1 , s2)
    - Возвращает указатель на позицию первого вхождения символа s2 в строку s1

- Возвращает указатель на позицию первого вхождения символа s1 в строку s2
- Возвращает указатель на позицию первого вхождения строки s2 в строку s1
- Возвращает указатель на позицию первого вхождения строки s1 в строку s2

16. Какая функция не отображает вводимый символ на экран?

| | | | |
|---|---|---|---|
| 1 | getche() | 3 | getchar() |
| 2 | getch() | 4 | gettch() |

II. Задания практические (каждое задание оценивается в 3 балла)

1. Опишите функцию, которая из однонаправленного динамического списка структур типа Успеваемость (поля: ID, название дисциплины, оценка, сложность дисциплины) удаляет все элементы, в которых значение поля сложность дисциплины больше 5.
2. Опишите функцию, которая из контейнера vector, хранящего данные типа Успеваемость (поля: ID, название дисциплины, оценка, сложность дисциплины) формирует новый vector, в котором все элементы имеют оценку «5» и уровень сложности больше «4».
3. Опишите функцию, которая бинарный файл структур типа Успеваемость (поля: ID, название дисциплины, оценка, сложность дисциплины) сортирует по убыванию значения поля сложность дисциплины.

Максимально возможное количество баллов за задачу II части – 3 балла. Для получения зачета необходимо набрать не менее 15 балов.

3 балла:

Ответ демонстрирует знание и корректное использование теоретического материала и умение составлять алгоритмы. Имеется полное верное решение задачи, включающее правильный ответ.

2 балла:

Ответ демонстрирует знание и корректное использование теоретического материала и умение составлять алгоритмы. Решение не содержит фактических ошибок. В решении имеются неверные записи И/ИЛИ синтаксические ошибки.

1 балл:

Ответ демонстрирует знание и корректное использование теоретического материала. Решение содержит алгоритмические и синтаксические.

0 баллов:

Решение содержит существенные алгоритмические ошибки, искажающие смысл задания. Решение не дано ИЛИ дано неверное решение.

**Примеры заданий для проведения промежуточной аттестаций в форме экзамена, 3 курс 5 семестр**

Обучающийся решает практическое задание и отвечает на теоретический вопрос.

Способ проведения промежуточной аттестации: письменное решение, устное пояснение.

**Пример 1.**

1. Однонаправленный линейный список: формирование списка, поиск по ключу, добавление и вставка элементов, удаление элементов по ключу.
2. Спроектируйте и реализуйте класс complex.

**Пример 2.**

1. Классы: функции-члены. Конструкторы и деструкторы. Перегрузка конструкторов. (Конструктор по умолчанию. Конструктор копирования. Конструктор с аргументами)
2. Создать шаблон класса «очередь». Написать программу, демонстрирующую работу с этим шаблоном для различных типов параметров шаблона.

**Критерии оценивания и шкала оценивания:**

Максимально возможное количество баллов – 40 баллов, при этом начисление баллов производится следующим образом:

Самостоятельно выполнено верно 85 - 100 % от всех заданий. Ответ на вопрос демонстрирует знание и корректное использование теоретического

материала. Факты и примеры в полном объеме обосновывают выводы. Имеется полное верное решение задачи, включающее правильный ответ – 30 - 40 баллов;

Самостоятельно выполнено верно 75 - 84% от всех заданий. Ответ на вопрос демонстрирует знание и корректное использование теоретического материала. Ответ не содержит фактических ошибок. Дано верное решение задачи, но в решении имеются неверные записи И/ИЛИ арифметические ошибки– 20 - 30 баллов;

Самостоятельно выполнено верно 50 - 74% заданий. Ответ на вопрос демонстрирует знание теоретического материала. Решение содержит фактические ошибки, не искажающие общего смысла. – 10 - 20 баллов;

Выполнено верно менее 50% заданий. В ответе преобладают рассуждения общего характера И/ИЛИ содержит существенные фактические ошибки, искажающие смысл. Решение не дано ИЛИ дано неверное решение – 0 – 10 баллов.

## V. Учебно-методическое и информационное обеспечение дисциплины

1) Рекомендуемая литература

а) Основная литература

1. Зоткин, С. П. Программирование на языке высокого уровня C/C++ : учебное пособие / С. П. Зоткин. — 3-е изд. — Москва : МИСИ – МГСУ, 2018. — 140 с. — ISBN 978-5-7264-1810-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/108512

2. Конова, Е. А. Алгоритмы и программы. Язык C++ : учебное пособие / Е. А. Конова, Г. А. Поллак. — 5-е изд., стер. — Санкт-Петербург : Лань, 2020. — 384 с. — ISBN 978-5-8114-5431-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/140730

3. Барков, И. А. Объектно-ориентированное программирование : учебник / И. А. Барков. — Санкт-Петербург : Лань, 2019. — 700 с. — ISBN 978-5-

8114-3586-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/119661

4. Беляков С. Л. Основы разработки программ на языке С++ для систем информационной безопасности : учебное пособие / С. Л. Беляков, А. В. Боженюк, М. В. Петряева; Южный федеральный университет. - Ростов-на-Дону : Издательство Южного федерального университета (ЮФУ), 2020. - 152 с. - ВО - Бакалавриат. – Режим доступа: https://znanium.com/catalog/document?id=374992

б) Дополнительная литература

1. Фридман, А. Л. Язык программирования Си++ : [16+] / А. Л. Фридман. – 2-е изд., исправ. – Москва : Национальный Открытый Университет «ИНТУИТ», 2016. – 219 с. – (Основы информационных технологий). – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=578114

2. Языки программирования: учебное пособие / О.Л. Голицына, Т.Л. Партыка, И.И. Попов. - 2-е изд., перераб. и доп. - М.: Форум, 2010. - 400 с.: ил.; 60x90 1/16. - (Профессиональное образование). (переплет) ISBN 978-5-91134-442-9. Режим доступа: http://znanium.com/go.php?id=226043

3. Кауфман В.Ш. Языки программирования. Концепции и принципы [Электронный ресурс] / В.Ш. Кауфман. — Электрон. текстовые данные. — Саратов: Профобразование, 2017. — 464 c. — 978-5-4488-0137-2. — Режим доступа: http://www.iprbookshop.ru/64055.html

4. Малиновская Е.А. Языки программирования. Часть 1 [Электронный ресурс] : лабораторный практикум / Е.А. Малиновская, Р.А. Рыскаленко. — Электрон. текстовые данные. — Ставрополь: Северо-Кавказский федеральный университет, 2016. — 103 c. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/69449.html

5. Страуструп Б. Язык программирования С++ для профессионалов [Электронный ресурс] / Б. Страуструп. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий

(ИНТУИТ), 2016. — 670 с. — 2227-8397. — Режим доступа: http://www.iprbookshop.ru/73737.html

6. Задачник-практикум по основам программирования [Электронный ресурс]: учебное пособие по курсу «Информатика»/ Н.И. Амелина [и др.].— Электрон. текстовые данные.— Ростов-на-Дону: Южный федеральный университет, 2009.— 192 с.— Режим доступа: http://www.iprbookshop.ru/46954.html

**2) Программное обеспечение**

| | |
|---|---|
| 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 ИС00000027 от 16.09.2011; |
| MATLAB R2012b | Акт предоставления прав № Us000311 от 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 |

**3) Современные профессиональные базы данных и информационные справочные системы**

1. ЭБС Лань https://e.lanbook.com/ Договор № 4-е/23 от 02.08.2023 г.

2. ЭБС Znanium.com https://znanium.com/ Договор № 1106 эбс от 02.08.2023 г.

3. ЭБС Университетская библиотека online https://biblioclub.ru Договор № 02-06/2023 от 02.08.2023 г.

4. ЭБС ЮРАЙТ https://urait.ru/ Договор № 5-е/23 от 02.08.2023 г.

5. ЭБС IPR SMART https://www.iprbookshop.ru/ Договор № 3-е/23К от 02.08.2023 г.

4) Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

http://www.cplusplus.com/reference - General information about the C++ programming language, including non-technical documents and descriptions

http://www.intuit.ru/studies/courses/648/504/info - Академия Microsoft: Структуры и алгоритмы компьютерной обработки данных: Авторы: Галина Ваныкина, Татьяна Сундукова

http://www.intuit.ru/studies/courses/16740/1301/info - Белоцерковская И. Е.Алгоритмизация. Введение в язык программирования C++ / И. Е. Белоцерковская, Н. В. Галина, Л. Ю. Катаева; И.Е. 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016.

http://www.intuit.ru/studies/courses/12181/1174/info - Седжвик Р. Алгоритмы на C++ / Р. Седжвик; Р. Седжвик. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016

## VI. Методические материалы для обучающихся по освоению дисциплины

***Методические рекомендации по организации самостоятельной работы студентов***

Процесс изучения дисциплины включает лекционные, лабораторные занятия и самостоятельную работу студента. Во время обучения применяется контактная технология преподавания (за исключением самостоятельно изучаемых студентами вопросов). При проведении занятий применяется

имитационный подход (анализ конкретных ситуаций), когда преподавателем разбирается на конкретном примере проблемная ситуация, все шаги решения задачи студентам демонстрируются при помощи мультимедийной техники. Затем студенты самостоятельно решают аналогичные задания. Так же при проведении занятий применяется частично-поисковый метод: студенты осуществляют поиск решения поставленной задачи. При этом постановочные задачи опираются на уже имеющиеся у студентов знания и умения, полученные в предшествующих темах. На занятиях практикуется выполнение заданий в малых группах, письменные работы, привлекаются ресурсы сети Интернет. Курс предусматривает выполнение тестов, контрольных и самостоятельных работ, домашних заданий.

Самостоятельная работа студентов в рамках данной дисциплины в основном состоит в подготовке к практическим занятиям и работе с разными источниками. Освоению учебного материала большую помощь окажет личный творческий подход, связанный с дополнительным просмотром материала по отдельным темам.

Самостоятельная работа является необходимой на всей стадиях и при всех формах изучения предмета. Важно помнить, что часы для самостоятельной работы, из всего объема времени, затраченного на дисциплину, будут превосходить иные виды работ.

Рекомендуется немедленно обсуждать любые возникшие в процессе обучения вопросы, проблемы и неясности с преподавателем, не откладывая это обсуждение до контрольной точки. Проконсультироваться с преподавателем можно во время и после занятий, во время консультаций, а также по электронной почте и в личном кабинете электронной образовательной среды (LMS).

**Список вопросов к зачету 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.

**Список вопросов к зачету 2 курс 4 семестр**

1. Понятие потока и файла. Двоичные и текстовые потоки. Работа с потоками средствами языка C. Функции для работы с файлами.
2. Понятие потока и файла. Двоичные и текстовые потоки. Работа с потоками средствами языка C++. Функции для работы с файлами.
3. Понятие и работа со списком. Однонаправленный и двунаправленный списки.
4. Понятие и работа с двоичным деревом.
5. Понятие и работа со стеком.
6. Понятие и работа с очередью.
7. STL - Контейнеры STL (вектор, двусвязный список, множество, карта (отображение), двусторонняя очередь) Итераторы. Алгоритмы STL.
8. Основные концепции ООП. Понятия класса и объекта. Абстракция. Инкапсуляция. Наследование. Полиморфизм. Методы, данные и свойства. Ограничение доступа к полям классам.
9. Классы. Методы и поля (данные) классов. Объявление класса в C++.

10. Разграничение доступа к полям и методам класса (спецификаторы доступа). Интерфейс и реализация класса.
11. Классы: функции-члены, конструкторы и деструкторы. Списки инициализации.
12. Конструкторы и деструкторы. Перегрузка конструкторов. (Конструктор по умолчанию. Конструктор копирования. Конструктор с аргументами)
13. Классы: Указатель this. Друзья класса. Дружественные классы и функции. Статические данные и методы. Особенности. Область применения. Пример.
14. Вложенные классы. Композиция.
15. Шаблоны класса: определение и инстанциирование.
16. Классы: перегрузка функций. перегрузка операторов.
17. Перегрузка бинарных операторов
18. Перегрузка операторов отношения и логических операторов
19. Перегрузка унарных операторов Перегрузка оператора присваивания и индекса массива
20. Классы: Принцип наследования в ООП. Варианты наследования. Публичное наследование. Защищенное наследование. Закрытое наследование. Иерархия классов
21. Классы: виртуальные функции и абстрактные классы.
22. Обобщенное программирование. Шаблоны функций. Шаблонные операторы. Шаблоны классов. Параметры шаблонов, не являющиеся типами.
23. Полиморфизм, его основные проявления, механизмы использования.
24. Понятие раннего и позднего связывания.
25. Использование виртуального механизма для реализации принципа полиморфизма.

**ВОПРОСЫ К ЭКЗАМЕНУ 2 курс 3 семестр**

1. Особенности языков С и С++. История создания языков.
2. Типы данных в языке С++. Размеры данных. Примеры. Переменные. Константы. Примеры. Переменные, особенности описания идентификаторов.
3. Область видимости переменных и время жизни объектов. Локальные и глобальные переменные. Классы памяти.
4. Унарные, бинарные, тернарная операции в языке С++. Различные виды оператора присваивания.
5. Операторы. (арифметические, инкремент и декремент, sizeof). Приоритет операций и очередность вычислений.
6. Операторы (логические, условные выражения, побитовые операции).
7. Структура программы на языке С++.
8. Вывод на экран и ввод с клавиатуры в языке С. Функции printf() и scanf(). Модификаторы точности, выравнивание.
9. Вывод на экран и ввод с клавиатуры в языке С++.
10. Условные операторы в С++. Управляющие структуры: инструкции и блоки, конструкция if-else, if-else-if, switch.
11. Базовые конструкции структурного программирования С++. Операторы цикла в С++: while и do-while.
12. Базовые конструкции структурного программирования С++. Оператор цикла for в С++. Вложенные циклы.
13. Управляющие структуры: инструкции break, continue, goto.
14. Понятие массива как структурированного типа данных. Одномерные и двумерные статические массивы: объявление, инициализация, ввод / вывод элементов.
15. Сортировка массива методом пузырька и методом выбора.
16. Указатели: понятие, применение, описание, инициализация. Адресная арифметика.
17. Динамическое распределение памяти в языке С. Выделение и освобождение динамической памяти. Функции malloc () и free ().

18. Динамическое распределение памяти. Выделение и освобождение динамической памяти. Операции new и delete. Утечка памяти.
19. Динамические массивы. Связь массивов с указателями. Различия между массивами и указателями.
20. Многомерные массивы. Динамические многомерные массивы.
21. С-строки: объявление, инициализация. Размещение строки в динамической памяти. Функции ввода-вывода С-строк: **getchar(), putchar(), getch(), getche(), scanf(), printf(), gets(), puts()**. Методы **getline** и **get** класса **istream**.
22. Операции со строками: **strcpy( ), strcpy( ), strcat( ), strlen( ), strcmp( ), strchr( ), strstr( )**.
23. Массивы строк: объявление, инициализация, ввод-вывод. Стандартные функции библиотеки <**ctype.h**> для работы с символами**: isalnum( ), isalpha( ), iscntrl( ), isdigit( ), isxdigit( ), isgraph( ), isprint( ), ispunct( ), isspace( ), islower( ), issupper( )**.
24. Понятие функции. Назначение функций, виды функций. Особенности функций в языке С.
25. Объявление и определение функций. Вызов функций. Возврат значений. Прототипы функций.
26. Формальные и фактические параметры. Локальные и глобальные переменные. Область видимости функции.
27. Особенности передачи параметров в функцию. Передача параметров по значению и по ссылке.
28. Функции в С++: массивы (одномерный и многомерный) как параметры функций.
29. Параметры со значениями по умолчанию. Рекурсивные функции.
30. Подставляемые функции. Перезагрузка функций.
31. Шаблоны функций: назначение, описание и вызов.
32. Структуры: описание, определение, объявление переменных структурного типа. Допустимые встроенные операции над структурами.

33. Инициализация переменных структурного типа и массива структур. Операторы доступа. Передача структур функциям.

## ВОПРОСЫ К ЭКЗАМЕНУ 3 курс 5 семестр

1. Файловая система в С. Бинарный поток. Открытие и закрытие файла. Режимы открытия и их назначение. Чтение и запись в бинарном режиме. Функции **rewind( ), fseek( ), ftell( )**.
2. Файловая система в С++. Потоки **ifstream, ofstream, fstream**. Открытие и закрытие файла. Режимы открытия и их назначение.
3. Чтение и запись текстовых файлов. Основные функции работы с файлами: **get( ),put( ), getline( ), gcount( ), ignore( ), peek( ), putback( ), flush( ), write( ), read( ), seekg( ), seekp( ), seek( ), tellg( ), tellp( ).**
4. Динамические структуры данных: описание, основные понятия, классификация. Элемент динамической структуры.
5. Однонаправленные списки: стек и очередь. Формирование списка, добавление и удаление элементов.
6. Однонаправленный линейный список: формирование списка, поиск по ключу, добавление и вставка элементов, удаление элементов по ключу.
7. Двунаправленный линейный список: формирование списка, поиск по ключу, добавление и вставка элементов, удаление элементов по ключу.
8. Библиотека STL. Понятие контейнера, алгоритма, итератора.
9. Библиотека STL. Контейнерные классы – краткое описание.
10. Библиотека STL. Векторы: объявление, операторы, функции-члены: **size( ), begin( ), end( ) , push_back( ), insert( ) и erase( ),** основные алгоритмы: **count( ) или count_if( ), remove( ), remove_copy( ), replace_copy( ), copy( ), reverse( ),fill( ), fill_n( ), adjacent_find( ), merge( ).**
11. . Контейнерный класс **string**: конструкторы, операторы, конкатенация, функции-члены класса **string**: **assign( ), append ( ), insert( ) и replace( ), erase( ), find( ) и rfind( ).**
12. Деревья. Основные понятия. Терминология.

13. Двоичные деревья: основные определения, описание вершины, деревья минимальной высоты, обход дерева.
14. Дерево поиска. Построение дерева поиска. Схема поиска, добавления и удаления элементов.
15. Основные концепции ООП. Понятия класса и объекта. Абстракция. Инкапсуляция. Наследование. Полиморфизм. Методы, данные и свойства. Ограничение доступа к полям классам.
16. Классы. Методы и поля (данные) классов. Объявление класса в C++.
17. Разграничение доступа к полям и методам класса (спецификаторы доступа). Интерфейс и реализация класса.
18. Классы: функции-члены. Конструкторы и деструкторы. Перегрузка конструкторов. (Конструктор по умолчанию. Конструктор копирования. Конструктор с аргументами)
19. Классы: Указатель this.
20. Друзья класса. Дружественные классы и функции.
21. Статические данные и методы. Особенности. Область применения. Пример.
22. Взаимодействия между объектами классов. Композиция. Агрегация. Ассоциация.
23. Вложенные классы. Композиция.
24. Шаблоны класса: определение и инстанциирование.
25. Классы: перегрузка функций. перегрузка операторов.
26. Классы: Принцип наследования в ООП. Варианты наследования. Публичное наследование. Защищенное наследование. Закрытое наследование. Иерархия классов
27. Классы: виртуальные функции и абстрактные классы.
28. Обобщенное программирование. Шаблоны функций. Шаблонные операторы. Шаблоны классов. Параметры шаблонов, не являющиеся типами.
29. Полиморфизм, его основные проявления, механизмы использования.

30. Использование виртуального механизма для реализации принципа полиморфизма.

## Вопросы для контрольных тестов и самоконтроля

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

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

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

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

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

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

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

8. Объясните принцип инкапсуляции.

9. Что такое композиция?

10. Для чего используются ключевые слова public и private?

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

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

13. Обязательно ли делать поля класса приватными?

14. Что такое метод? Как вызывается метод?

15. Может ли метод быть приватный?

16. Как определить метод непосредственно внутри класса? А вне класса? Чем эти определения отличаются?

17. Можно в методах присваивать параметрам значения по умолчанию?

18. Что обозначается ключевым словом this?

19. Зачем нужны константные методы? Чем отличается определение константного метода от обычного?

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

21. Объясните принцип полиморфизма.

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;
}
```

## *Тематика курсовых работ и методические рекомендации по их выполнению* (4 семестр)

Целями курсового проектирования являются:

- закрепление знаний, полученных в ходе теоретического и практического изучения дисциплины «Программирование (Объектно-ориентированное программирование)»;
- приобретение навыков практического программирования с использованием объектно-ориентированной парадигмы;
- изучение современных систем программирования и сред для разработки объектно-ориентированных программ и систем;
- изучение отдельных разделов предметной области, не вошедших в программу теоретического обучения, формирование навыка поиска информации по конкретной теме, ее анализа и использования для решения задачи;

Курсовая работа позволяет сформировать способности будущего специалиста к самостоятельному решению практических задач и инжнерных проблем с использованием теоретических положений, а также знаний и умений, полученных в ходе обучения объектно-ориентированному программированию.

Задачей курсового проектирования является разработка иерархии типов в заданной предметной области, включающая в себя:

- разработку модели классов на языке UML;
- разработку абстрактных классов и интерфейсов;
- разработку тестирующего приложения;
- разработку документации;
- подготовку презентации.

1. Разработка класса для представления множества целых чисел на основе связанного списка. Операции – включение элемента, исключение элемента, поиск элемента, объединение множеств, пересечение множеств.
2. Разработка класса для представления упорядоченного множества строк на основе бинарного дерева. Операции – включение элемента, исключение элемента, поиск элемента, объединение множеств, пересечение множеств.

3. Создать шаблон циклической очереди. С помощью него обработать ввод с клавиатуры, заполнение информацией вашей памяти.
4. Создать шаблон приоритетной очереди. При добавлении элемента в такую очередь порядковый номер нового элемента определяется его приоритетом..
5. С помощью шаблона класса стек написать программу калькулятор с операциями сложения, вычитания, деления, умножения, возведения в степень.
6. 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.
7. Программа «Англо-русский и русско-английский словарь». «База данных» словаря должна содержать синонимичные перевода слов. Программа должна обеспечивать выбор с помощью меню и выполнение одной из следующих функций:

- Загрузка «базы данных» словаря (из файла).
- Выбор режима работы: англо-русский или русско-английский.
- Вывод перевода заданного английского слова.
- Вывод перевода заданного русского слова.
- Базу данных словаря реализовать в виде двух контейнеров типа тар.

8. Программа, реализующую игру «Крестики-нолики» между двумя игроками: пользователем и компьютером (роботом). В программе использовать контейнерные классы STL.
9. Создание игры «Змейка». Реализовать игру «Змейка» в виде апплета и разместить на сайте кафедры АСОИУ в разделе выполненных курсовых проектов. Описание: Игрок управляет длинным, тонким существом, напоминающим змею, которое ползает по плоскости (как правило,

ограниченной стенками), собирая еду (или другие предметы), избегая столкновения с собственным хвостом и краями игрового поля. В некоторых вариантах на поле присутствуют дополнительные препятствия. Каждый раз, когда змея съедает кусок пищи, она становится длиннее, что постепенно усложняет игру. Игрок управляет направлением движения головы змеи (обычно 4 направления: вверх, вниз, влево, вправо), а хвост змеи движется следом. Игрок не может остановить движение змейки.

10. Игра-аркада “Snake” Правила игры: Чтобы выиграть в KSnake, вам нужно съесть все яблоки в комнате и выйти через дверь, которая откроется вверху. С каждым съеденным яблоком вы становитесь длиннее. Если вы врежетесь в стену, вы умрете. Если вы врежетесь в себя, вы умрете. Если вам в голову попадет мяч, вы умрете. Если вы слишком долго не будете есть яблоки, появятся новые.
11. Реализовать игру «Сапёр». Описание: обычно двухмерное прямоугольное игровое поле поделено на клетки или другие части, некоторые из которых содержат скрытые мины. Игрок открывает клетки, стараясь не попасть на мину. Если игрок откроет клетку (или другую область) с миной, игра заканчивается. Если же мины нет, то в клетке появляется число, обозначающее количество мин в соседних клетках (в различных вариантах игры соседство может определяться по-разному). Рассчитав при помощи таких чисел расположение мин, игрок может пометить соответствующие клетки специальной меткой, чтобы случайно не открыть их.
12. Реализация игры «Жизнь» Конвея. Место действия этой игры, «вселенная» – это размеченная на клетки поверхность, безграничная, ограниченная, или замкнутая. В компьютерных реализациях игры чаще всего используют поверхность тора. Каждая клетка на этой поверхности может находиться в двух состояниях: быть живой или быть мёртвой. Клетка имеет восемь соседей. Распределение живых клеток в начале игры называется

первым поколением. Каждое следующее поколение рассчитывается на основе предыдущего по таким правилам:

- пустая (мёртвая) клетка ровно с тремя живыми клетками-соседями оживает;
- если у живой клетки есть две или три живые соседки, то эта клетка продолжает жить; в противном случае (если соседок меньше двух или больше трёх) клетка умирает (от «одиночества» или от «перенаселённости»).
- Игрок не принимает прямого участия в игре, а лишь расставляет «живые» клетки, которые взаимодействуют согласно правилам уже без его участия.
- Начальные конфигурации формируются случайно (с помощью генератора случайных чисел для заданной вероятности). Эволюция, порождаемая заданной начальной конфигурацией состояний клеток, может быть изучена в общем случае только путём её пошагового воспроизведения.

13. Реализация игры «Инь-Ян». Игра «Инь-Ян» представляет собой клеточный автомат, состояния ячеек и правила переключения которого приближённо отражают закон единства и борьбы противоположностей. Ячейки автомата имеют три состояния: пустая ячейка (мёртвая), живая ячейка Инь и живая ячейка Ян. Соседние по миру ячейки (у каждой ячейки их 8), если они живые, называются соседями. Правила переключения определены таким образом, чтобы популяции ячеек Инь и Ян противоборствовали, но не могли развиваться друг без друга. Вот эти правила:

- Рождение. У пустой ячейки ровно три соседа (живых) и они не все одинаковые, то в ней рождается Ян, когда среди соседей только один Ян, или Инь, когда среди соседей только один Инь.

- Гибель от перенаселения (одиночества). Живая ячейка, имеющая больше четырех (меньше двух) соседей, умирает от перенаселения (от одиночества);
- Гибель в неравном противостоянии. У живой ячейки ровно четыре соседа, из которых большинство – противоположного типа – ячейка умирает.
- Начальные конфигурации формируются случайно (с помощью генератора случайных чисел для заданных вероятностей Инь и Ян). Эволюция, порождаемая заданной начальной конфигурацией состояний клеток, может быть изучена в общем случае только путём её пошагового воспроизведения. Однако для каждого значения вероятности генерации можно отследить статистику результата через n шагов и говорить в каком случае будет больше вероятность вырождения автомата за заданное количество шагов.

Требования к рейтинг-контролю для студентов очной формы обучения.

Текущая работа студентов очной формы обучения во 2 семестре оценивается в 100 баллов, которые распределяются между двумя модулями (периодами обучения) следующим образом:

| Модуль (период обучения) | Максимальная сумма баллов в модуле | Максимальная сумма баллов за работу на практических и лабораторных занятиях | Максимальный балл за рейтинговую контрольную работу |
|---|---|---|---|
| 1 | 50 | 30 | 20 |
| 2 | 50 | 30 | 20 |

Текущая работа студентов в 3 семестре оценивается в 60 баллов, которые распределяются между двумя модулями (периодами обучения) следующим образом:

| Модуль (период обучения) | Максимальная сумма баллов в модуле | Максимальная сумма баллов за работу на практических и лабораторных занятиях | Максимальный балл за рейтинговую контрольную работу |
|---|---|---|---|
| 1 | 30 | 20 | 10 |
| 2 | 30 | 20 | 10 |

Текущая работа студентов в 4 семестре оценивается в 100 баллов, которые распределяются между двумя модулями (периодами обучения) следующим образом:

| Модуль (период обучения) | Максимальная сумма баллов в модуле | Максимальная сумма баллов за работу на практических и лабораторных занятиях | Максимальный балл за рейтинговую контрольную работу |
|---|---|---|---|
| 1 | 50 | 30 | 20 |
| 2 | 50 | 30 | 20 |

Текущая работа студентов в 5 семестре оценивается в 60 баллов, которые распределяются между двумя модулями (периодами обучения) следующим образом:

| Модуль (период обучения) | Максимальная сумма баллов в модуле | Максимальная сумма баллов за работу на практических и лабораторных занятиях | Максимальный балл за рейтинговую контрольную работу |
|---|---|---|---|
| 1 | 30 | 20 | 10 |
| 2 | 30 | 20 | 10 |

Правила формирования рейтинговой оценки и шкалу пересчета рейтинговых баллов в оценку на экзамене см. в «Положении о рейтинговой системе обучения в ТвГУ»:

https://tversu.ru/sveden/files/204-R_Pologhenie_o_reytingovoy_sisteme_obucheniya_v_TvGU.pdf

## VII. Материально-техническое обеспечение

Учебный процесс по данной дисциплине проводится в аудиториях, оснащенных мультимедийными средствами обучения. Для организации самостоятельной работы студентов необходимо наличие персональных компьютеров с доступом в Интернет.

| **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 промежуточной аттестации, Компьютерный класс математического факультета № 16 (Корпус 3, 170002, Тверская обл., г.Тверь, пер. Садовый, дом 35) | Столы, стулья, переносной ноутбук, компьютеры | 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; Git version 2.5.2.2-бесплатно; Google Chrome-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 ИС00000027 от 16.09.2011; MATLAB R2012b-Акт предоставления прав № Us000311 от 25.09.2012; Многофункциональный редактор ONLYOFFICE - бесплатно; ОС Linux Ubuntu бесплатное ПО-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» ; PostgreSQL 9.6 -бесплатно; Python 3.4.3-бесплатно; Visual Studio 2010 Prerequisites - English-Акт на передачу прав №785 от 06.08.2021 г. ; WCF RIA Services V1.0 SP2-бесплатно; WinDjView 2.1-бесплатно; WinPcap 4.1.3- |

| | | |
|---|---|---|
| | | бесплатно; Wireshark 2.0.0 (64-bit)-бесплатно; R studio-бесплатно. |
| Учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации,<br>Учебная аудитория № 314 (Корпус 3, 170002, Тверская обл., г.Тверь, пер. Садовый, дом 35) | Набор учебной мебели, меловая доска,<br>Мультимедийный комплект учебного класса (вариант № 2): Проектор Casio XJ-140 настенный проекц. экран Lumien 180*180,<br>Ноутбук Dell N4050, сумка 15,6", мышь;<br>Усилитель Roxton AA-120;<br>Радиосистема Shure PG288/PG58;<br>Микшер Mackie 402 VLZ;<br>Стационарный микрофон SOUNDKING EG002 с настольным держателем;<br>Мультимедийный проектор Casio XJ-H2650 с потолочным креплением и мотириз. экраном;<br>Шкаф напольный 19". | 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;<br>ОС Linux Ubuntu бесплатное ПО-бесплатно |

## VIII. Сведения об обновлении рабочей программы дисциплины

| №п.п. | Обновленный раздел рабочей программы дисциплины | Описание внесенных изменений | Реквизиты документа, утвердившего изменения |
|---|---|---|---|
| 1. | V. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины | Обновление списка литературы. Обновление ссылок из ЭБС. | Протокол № 1 от 27.09.2015 |
| 2. | VII. Методические указания для обучающихся по освоению дисциплины. | Корректировка планов практических (семинарских) занятий и методических рекомендаций к ним. | Протокол № 1 от 01.09.2016 |

| | | | |
|---|---|---|---|
| 3. | I - X | Корректировка всех разделов в соответствии с новым стандартом | Протокол № 6 от 28.02.2017 |
| 4. | V. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины | Дополнение списков. Обновление ссылок из ЭБС. | Протокол № 1 от 01.09.2019 |
| 5. | I - VIII | Корректировка всех разделов в соответствии с новым стандартом | Протокол № 10 от 29.06.2021 |
| 6. | V. Учебно-методическое и информационное обеспечение дисциплины | Обновление списков ПО. Обновление ссылок из ЭБС. | Протокол № 1 от 1.09.2023 |
| 7 | II. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий, IV. Оценочные материалы для проведения текущей и промежуточной аттестации | Корректировка наименований разделов и тем. Корректировка оценочных материалов | Протокол № 7 от 7.03.2024 |