**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

**Отчеты по лабораторным и практическим работам**

**по МДК.01.01**

Студент: Матигоров Н.И.

Группа: ИСПП-31

Преподаватель: Маломан Ю.С.

Архангельск 2025

**Лабораторная работа №1**

**Оценка сложности алгоритмов сортировки**

**1 Цель работы**

1.1 Научиться реализовывать и оценивать сложность алгоритмов сортировки

массивов на C#

**2 Контрольные вопросы**

2.1 Что такое «массив»?

Массив — это упорядоченная коллекция однотипных элементов, каждый из которых доступен по своему уникальному индексу.

2.2 Как описывается одномерный массив?

тип\_массива[] название\_массива;

2.3 Как обратиться к некоторому элементу одномерного массива?

название\_массива[индекс элемента]

2.4 Как можно задать одномерный массив?

тип\_массива[] название\_массива = new тип\_массива[размер\_массива];

2.5 Что такое «сортировка»?

Сортировка в программировании — это упорядочивание элементов массива или списка в определённом порядке, обычно по возрастанию или по убыванию.

2.6 Что такое «алгоритм сортировки»?

Алгоритм сортировки — это последовательность операций, которая упорядочивает элементы в массиве или списке по определённому признаку.

2.7 Какие виды сортировки массивов существуют?

Простые, продвинутые и гибридные.

**3 Вывод**

3.1 Я научился реализовывать и оценивать сложность алгоритмов сортировки

массивов на C#.

**Лабораторная работа №2**

**Оценка сложности алгоритмов поиска**

**1 Цель работы**

1.1 Научиться реализовывать и оценивать сложность алгоритмов поиска элементов массивов на C#.

**2 Контрольные вопросы**

2.1 Что такое «алгоритм сортировки»?

Алгоритм сортировки — это последовательность операций, которая упорядочивает элементы в массиве или списке по определённому признаку.

2.2 Какие виды поиска элементов массивов существуют?

Линейный, двоичный, поиск прыжками, алгоритм Фибоначчиевого поиска и др.

2.3 В чем особенность алгоритма линейного поиска и какова его временная

сложность?

Особенность алгоритма линейного поиска (последовательного поиска) — последовательный просмотр каждого элемента списка до тех пор, пока не будет найден искомый элемент или не будет достигнут конец списка. Временная сложность O(n).

2.4 В чем особенность алгоритма двоичного поиска и какова его временная

сложность?

Особенность алгоритма двоичного поиска в том, что он делит заранее отсортированный массив данных пополам, чтобы обнаружить нужный элемент. Временная сложность O(log(N)).

2.5 В чем особенность алгоритма поиска прыжками и какова его временная

сложность?

Особенность алгоритма поиска прыжками — похож на двоичный поиск, но движение только вперёд. Такой поиск требует отсортированной коллекции Временная сложность O(sqrt(N)).

**3 Вывод**

3.1 Я научился реализовывать и оценивать сложность алгоритмов поиска элементов массивов на C#.

**Лабораторная работа №3**

**Оценка сложности рекурсивных алгоритмов**

**1 Цель работы**

1.1 Научиться разрабатывать и оценивать сложность рекурсивных функций в программах на C#.

**2 Контрольные вопросы**

2.1 Что такое «рекурсия»?

Рекурсия — это метод, при котором функция вызывает саму себя для решения задачи.

2.2 Какие проблемы могут возникать при реализации рекурсивных алгоритмов на электронных вычислительных машинах?

При реализации рекурсивных алгоритмов на электронных вычислительных машинах могут возникать проблемы, связанные с переполнением стека, неэффективностью и сложностью отладки.

2.3 Какое определение функции может быть названо рекурсивным? Привести

примеры.

Рекурсивной может быть названа функция, которая в процессе выполнения вызывает саму себя. К примеру, функция вычисления факториала.

2.4 Что такое «глубина рекурсии»?

Глубина рекурсии — это максимальное количество уровней рекурсивных вызовов, которые функция может выполнить до достижения базового случая или предела, установленного интерпретатором языка программирования.

2.5 Что такое «рекурсивный спуск»?

Рекурсивный спуск в программировании — это метод синтаксического анализа, при котором программа представляет себя в виде набора подпрограмм, каждая из которых читает и обрабатывает свою конструкцию, а также вызывает (в том числе рекурсивно) соответствующие подпрограммы для анализа вложенных конструкций.

2.6 Что такое «рекурсивный подъём»?

Это завершение работы рекурсивных подпрограмм, вплоть до самой первой, инициировавшей рекурсивные вызовы.

**3 Вывод**

3.1 Я смог научиться разрабатывать и оценивать сложность рекурсивных функций в программах на C#.

**Лабораторная работа №4**

**Оценка сложности эвристических алгоритмов**

**1 Цель работы**

1.1 Научиться реализовывать, оценивать и применять эвристические алгоритмы.

**2 Контрольные вопросы**

2.1 Что такое «эвристика»?

Эвристика — совокупность приёмов и методов, которые облегчают и упрощают решение познавательных, конструктивных и практических задач..

2.2 Зачем используются эвристические алгоритмы?

Эвристические алгоритмы используются, чтобы ускорить решение задачи в тех случаях, когда точное решение не может быть найдено.

2.3 Какие алгоритмы относятся к эвристическим?

Алгоритм поиска локального максимума, жадный алгоритм, метод ветвей и границ

**3 Вывод**

Я начал реализовывать, оценивать и применять эвристические алгоритмы.

**Лабораторная работа №5**

**Работа с классами**

**1 Цель работы**

1.1 Изучить процесс разработки и применения классов на языке C#;

1.2 Изучить реализацию механизма инкапсуляции на языке C#.

**2 Контрольные вопросы**

2.1 Какова общая форма объявления класса в C#?

Class имя\_класса

{

// тело класса

}.

2.2 На какие виды делятся данные класса в C#?

Поля, константы, события.

2.3 На какие виды делятся функции класса в C#?

Статические и экземплярные(зависящие от свойств определённого экземпляра).

2.4 Для чего применяются конструкторы классов?

Конструкторы классов применяются для инициализации объектов.

2.5 Что такое «цепочка конструкторов»?

Цепочка конструкторов — это механизм, с помощью которого конструктор класса вызывает другой конструктор того же класса.

2.6 Для чего применяются свойства классов?

Свойства классов (акссесоры) используются для доступа к внутреннему состояния объекта.

2.7 Когда используются автоматически реализуемые свойства классов?

В классах или структурах с модификатором record.

2.8 В чем отличие вызова статические членов классов от членов экземпляра класса?

Член экземпляра класса относится к конкретному экземпляру класса. Статический член класса, напротив, относится к классу в целом, а не к конкретному экземпляру класса.

**3 Вывод**

3.1 Был изучен процесс разработки и применения классов на C#

3.2 Был изучен процесс реализации механизма инкапсуляции на языке C#.

**Лабораторная работа №6**

**Перегрузка методов**

**1 Цель работы**

1.1 Изучить процесс выполнения перегрузки методов на языке C#;

1.2 Изучить различные способы передачи параметров в метод.

**2 Контрольные вопросы**

2.1 Что такое «перегрузка методов»?

Это один из мощнейших механизмов языка, который позволяет определить несколько методов с одним и тем же именем, но с различными параметрами.

2.2 К какому виду полиморфизма относится перегрузка методов?

Статический.

2.3 Какие типы функций класса можно перегружать?

Любые, кроме некоторых операторов.

2.4 Какие требования предъявляются к сигнатуре перегружаемых функций класса?

Различный набор параметров.

2.5 Какие типы функций класса нельзя перегружать?

Некоторые оперторы: ^x, x = y, x.y, x?.y, c ? t : f, x ?? y, x ??= y, x..y, x->y, =>, f(x), as, await, checked, unchecked, default, delegate, is, nameof, new, sizeof, stackalloc, switch, typeof, with.

**3 Вывод**

3.1 Я изучил процесс выполнения перегрузки методов на языке C#;

3.2 Были изучены различные способы передачи параметров в метод.

**Лабораторная работа №7**

**Определение операций в классе**

**1 Цель работы**

1.1 Изучить процесс определения операций в классе на языке C#.

**2 Контрольные вопросы**

2.1 Какое ключевое слово определяет операторный метод?

Operator.

2.2 Какие унарные операторы можно перегружать в C#?

+, -, !, ~, ++, --, true, false.

2.3 Какие бинарные операторы можно перегружать в C#?

x + y, x - y, x \* y, x / y, x % y, x & y, x | y, x ^ y, x << y, x >> y.

2.4 Какие операторы сравнения можно перегружать в C#?

x == y, x != y, x < y, x > y, x <= y, x >= y.

2.5 Какие операторы требуется перегружать в C# попарно?

x + y, x - y, x \* y, x / y, x % y, x & y, x | y, x ^ y, x << y, x >> y, x == y, x != y, x < y, x > y, x <= y, x >= y.

**3 Вывод**

3.1 Я изучил процесс определения операций в классе на языке C#.

**Лабораторная работа №8**

**Создание наследованных классов**

**1 Цель работы**

1.1 Изучить процесс разработки дочерних классов на языке C#;

1.2 Изучить реализацию механизма наследования на языке C#.

**2 Контрольные вопросы**

2.1 Что такое «наследование»?

Наследование — это принцип объектно-ориентированного программирования, который позволяет создавать иерархии классов, где один класс (подкласс) наследует свойства и методы другого класса.

2.2 Сколько родительских классов может быть у дочернего класса в C#?

Один.

2.3 Вопрос2?

Ответ2.

2.4 Вопрос2?

Ответ2.

2.5 Вопрос2?

Ответ2.

2.6 Вопрос2?

Ответ2.

**3 Вывод**

**Лабораторная работа №1**

**Название**

**1 Цель работы**

1.1 Цель1,

1.2 Цель2.

**2 Контрольные вопросы**

2.1 Вопрос1?

Ответ1.

2.2 Вопрос2?

Ответ2.

2.3 …

**3 Вывод**

**Лабораторная работа №1**

**Название**

**1 Цель работы**

1.1 Цель1,

1.2 Цель2.

**2 Контрольные вопросы**

2.1 Вопрос1?

Ответ1.

2.2 Вопрос2?

Ответ2.

2.3 …

**3 Вывод**

**Лабораторная работа №1**

**Название**

**1 Цель работы**

1.1 Цель1,

1.2 Цель2.

**2 Контрольные вопросы**

2.1 Вопрос1?

Ответ1.

2.2 Вопрос2?

Ответ2.

2.3 …

**3 Вывод**

**Лабораторная работа №1**

**Название**

**1 Цель работы**

1.1 Цель1,

1.2 Цель2.

**2 Контрольные вопросы**

2.1 Вопрос1?

Ответ1.

2.2 Вопрос2?

Ответ2.

2.3 …

**3 Вывод**

**Лабораторная работа №1**

**Название**

**1 Цель работы**

1.1 Цель1,

1.2 Цель2.

**2 Контрольные вопросы**

2.1 Вопрос1?

Ответ1.

2.2 Вопрос2?

Ответ2.

2.3 …

**3 Вывод**

**Лабораторная работа №1**

**Название**

**1 Цель работы**

1.1 Цель1,

1.2 Цель2.

**2 Контрольные вопросы**

2.1 Вопрос1?

Ответ1.

2.2 Вопрос2?

Ответ2.

2.3 …

**3 Вывод**

**Лабораторная работа №1**

**Название**

**1 Цель работы**

1.1 Цель1,

1.2 Цель2.

**2 Контрольные вопросы**

2.1 Вопрос1?

Ответ1.

2.2 Вопрос2?

Ответ2.

2.3 …

**3 Вывод**

**Лабораторная работа №1**

**Название**

**1 Цель работы**

1.1 Цель1,

1.2 Цель2.

**2 Контрольные вопросы**

2.1 Вопрос1?

Ответ1.

2.2 Вопрос2?

Ответ2.

2.3 …

**3 Вывод**