

ФЕДЕРАЛЬНОЕ АГЕНСТВО СВЯЗИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)**

**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ
ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ
(АКТ (ф) СПбГУТ)**

**Составил
Ю.С. Маломан**

ОП.04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

Методические указания по выполнению лабораторных работ

по специальностям:

09.02.07 – Информационные системы и программирование

Архангельск 2020

Маломан, Ю.С. ОП.04 Основы алгоритмизации и программирования. Методические указания по выполнению лабораторных работ. – Архангельск: АКТ (ф) СПбГУТ, 2020.

Методические указания по выполнению лабораторных работ содержат описания работ, предусмотренных рабочей программой по ОП.04 «Основы алгоритмизации и программирования». Каждая работа рассчитана на 2 часа, общий объём составляет 76 часов. Методические указания по выполнению лабораторных работ предназначены для студентов очной формы обучения по специальности 09.02.07 – Информационные системы и программирование.

Рассмотрено и одобрено на заседании цикловой комиссии Информационных технологий и математических дисциплин АКТ (ф) СПбГУТ.

© АКТ (ф) СПбГУТ

Усл. печ.л. 2,77

Содержание

| | |
|--|----|
| Лабораторная работа №1 Знакомство со средой программирования | 5 |
| Лабораторная работа №2 Составление программ линейной структуры | 8 |
| Лабораторная работа №3 Составление программ разветвляющейся структуры | 10 |
| Лабораторная работа №4 Составление программ циклической структуры | 12 |
| Лабораторная работа №5 Обработка одномерных массивов | 14 |
| Лабораторная работа №6 Обработка двумерных массивов | 16 |
| Лабораторная работа №7 Работа со строками | 18 |
| Лабораторная работа №8 Организация функций | 20 |
| Лабораторная работа №9 Организация процедур | 22 |
| Лабораторная работа №10 Работа с текстовыми файлами | 24 |
| Лабораторная работа №11 Работа с текстовыми файлами с использованием потоков | 26 |
| Лабораторная работа №12 Разработка структур | 28 |
| Лабораторная работа №13 Работа с двоичными файлами | 31 |
| Лабораторная работа №14 Изучение правил разработки схем алгоритмов | 33 |
| Лабораторная работа №15 Тестирование, отладка приложения | 35 |
| Лабораторная работа №16 Программирование модуля | 37 |
| Лабораторная работа №17 Создание библиотеки подпрограмм | 39 |
| Лабораторная работа №18 Работа с алгоритмами сортировки массивов | 41 |
| Лабораторная работа №19 Использование указателей для организации связанных списков | 42 |
| Лабораторная работа №20 Изучение процесса разработки классов на C++ | 44 |
| Лабораторная работа №21 Изучение процесса разработки дочерних классов на C++ | 46 |
| Лабораторная работа №22 Изучение процесса разработки шаблонов функций и классов на C++ | 48 |
| Лабораторная работа №23 Изучение среды разработки | 50 |
| Лабораторная работа №24 Создание проекта с использованием компонентов для работы с текстом | 53 |
| Лабораторная работа №25 Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени | 55 |
| Лабораторная работа №26 Создание проекта с использованием компонентов выбора | 57 |
| Лабораторная работа №27 Создание проекта с использованием компонентов отображения списков | 59 |
| Лабораторная работа №28 Изучение способов представления и обработки табличных данных в приложениях на C# | 61 |
| Лабораторная работа №29 Создание проекта с использованием компонентов стандартных диалогов | 65 |
| Лабораторная работа №30 Создание проекта с использованием кнопочных компонентов | 67 |
| Лабораторная работа №31 Создание проекта с использованием меню и панели инструментов | 69 |

| | |
|---|----|
| Лабораторная работа №32 Обработка исключений | 71 |
| Лабораторная работа №33 Обработка файлов в оконном приложении | 72 |
| Лабораторная работа №34 Разработка интерфейса приложения | 74 |
| Лабораторная работа №35 Разработка многооконного приложения | 76 |
| Лабораторная работа №36 Разработка функциональной схемы работы приложения | 78 |
| Лабораторная работа №37 Разработка игрового приложения | 80 |
| Лабораторная работа №38 Разработка приложений | 84 |

Лабораторная работа №1

Знакомство со средой программирования

1 Цель работы

- 1.1 Ознакомиться с интерфейсом Visual Studio;
- 1.2 Научиться создавать приложение на C в Visual Studio;
- 1.3 Научиться работать в среде программирования Visual Studio.

2 Литература

2.1 Тракимус, Ю. В. Разработка консольных приложений с помощью Microsoft Visual Studio 2017: учебное пособие / Ю. В. Тракимус. – Новосибирск : Новосибирский государственный технический университет, 2018. – URL: <https://ibooks.ru/reading.php?productid=367816>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – с.4-37.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание лабораторной работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Создание и запуск первого приложения в Visual Studio

5.1.1 Запустить Microsoft Visual Studio и создать новый проект консольного приложения, выбрав команду в меню **Файл>Создать>Проект**.

5.1.2 В появившемся окне выбрать язык **C++** и шаблон проекта **Мастер классических приложений**. После этого в новом окне заполнить текстовые поля и нажать **Создать**.

5.1.3 Далее в открывшемся окне установить флажок **Предкомпилированный заголовок** и нажать на кнопку **ОК**.

- 5.1.4 В открывшемся окне будет заготовка программы со следующим кодом:

```
#include <iostream>

int main()
{
    std::cout<>>"Hello world\n";
}
```

5.1.5 Для проверки работы запустить приложение, нажав **Ctrl+F5** (запуск без отладки) или кнопку с пиктограммой зеленый треугольник. После этого откроется окно, в котором можно будет вводить исходные данные и видеть результаты работы программы.

5.1.6 Изменить содержимое функции **main** на следующее строку для вывода на экран **Hello World**:

```
printf("Hello world");  
return 0;
```

Для проверки работы запустить приложение, сравнить результат с предыдущим запуском.

5.1.7 Добавить в строку вывода спецификатор для перевода на новую строку:

```
printf("Hello world\n");
```

Для проверки работы запустить приложение, сравнить результат с предыдущим запуском.

5.2 Добавление в решение нового проекта

5.2.1 Во вкладке **Обозреватель решений** нажать ПКМ (правой кнопкой мыши) на название решения и выбрать **Добавить > Создать проект**

В открывшемся окне выбрать **Консольное приложение** и указать настройки проекта.

5.2.2 В обозревателе решений запускаемый проект выделяется полужирным. По умолчанию запускаемым считается последний созданный. Чтобы сделать запускаемым другой проект, нужно нажать на него ПКМ в обозревателе решений и выбрать **Назначить в качестве запускаемого**

5.2.3 Запустить приложение

5.3 Ввод и вывод данных

5.3.1 Добавить в решение новый проект ThirdApp и сделать его запускаемым.

5.3.2 Написать программу, в которой пользователь вводит значения скорости в км/ч (int v;) и времени в часах (float t;).

Реализовать вывод пройденного с указанной скоростью за указанное пользователем время расстояния ($\text{float } s = v * t$;). Округлить результат до 3 знаков после запятой.

5.4 Отладка и просмотр значений переменных

5.4.1 Выполнить пошаговое выполнение программы в последнем приложении.

5.4.2 В процессе пошагового выполнения изучить значения во вкладке **Локальные**.

5.4.3 Добавить в приложение три точки останова. Выполнить по ним проход.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать консольный проект C++ и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

- 8.1 Как создать новый проект на C в Visual Studio?
- 8.2 Как запустить созданное приложение в Visual Studio?
- 8.3 Как запустить приложение в режиме отладки?
- 8.4 Как добавить в приложение брейкпоинты (точки останова)?
- 8.5 Как запустить приложение с переходом по брейкпоинтам?

Лабораторная работа №2

Составление программ линейной структуры

1 Цель работы

- 1.1 Изучить процесс разработки линейных алгоритмов на языке Си;
- 1.2 Научиться считывать введенные пользователем данные;
- 1.3 Научиться выводить данные в форматированном виде.

2 Литература

2.1 Тракимус, Ю. В. Разработка консольных приложений с помощью Microsoft Visual Studio 2017: учебное пособие / Ю. В. Тракимус. – Новосибирск : Новосибирский государственный технический университет, 2018. – URL: <https://ibooks.ru/reading.php?productid=367816>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – с.4-37.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание лабораторной работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Создать консольное приложение, которое выводит на экран три строки, используя команду printf и спецификатор перевода на новую строку. В первой строке — группа, во второй — фамилии разработчиков, в третьей — текущий год. Текст должен выводиться на русском языке.

5.2 Создать консольное приложение, выводящее результат арифметических операций (сложение, вычитание, умножение, деление, остаток от деления) для двух целых чисел а и b. Значения чисел задать программно.

Результат вывести согласно образцу:

$$11 + 3 = 14$$

$$11 - 3 = 8$$

$$11 * 3 = 33$$

$$11 / 3 = 3,67$$

$$11 \% 3 = 2$$

5.3 Создать консольное приложение, находящее площадь кольца на основе значений внешнего и внутреннего радиусов. Радиусы (вещественные числа) вводятся пользователем. Результат вывести с тремя знаками после запятой. Площадь круга вычисляется по формуле: $S=\pi \cdot R^2$. В приложение π объявить как константу 3.1415.

6 Порядок выполнения работы

- 6.1 Используя Microsoft Visual Studio, создать проект C++ и выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 С чего начинается программа на языке программирования Си?
- 8.2 Как объявить переменные на языке программирования Си?
- 8.3 С чего начинается основная программа на языке программирования Си?
- 8.4 Как вывести информацию на экран на языке программирования Си?
- 8.6 Как осуществить ввод информации с клавиатуры на языке программирования Си?
- 8.7 Как присвоить значение переменной?
- 8.8 Какие арифметические операции можно применять к переменным на языке программирования Си?

Лабораторная работа №3

Составление программ разветвляющейся структуры

1 Цель работы

1.1 Изучить процесс разработки разветвляющихся алгоритмов на языке Си.

2 Литература

2.1 Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C++ / Т. И. Немцова, С. Ю. Голова, А. И. Терентьев. – Москва : Форум, 2019. – URL: <https://ibooks.ru/reading.php?productid=361544>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.4.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Написать программу, определяющую по введенному пользователем номеру месяца сезон (зима, весна, лето, осень). Если введен некорректный номер месяца, вывести сообщение, что такого месяца нет.

5.2 Написать программу, определяющую по введенному пользователем году и номеру месяца (считать, что год и номер месяца введены корректно):

- является ли введенный пользователем год високосным (т.е. кратен 4 и не кратен 100 или кратен 400);

- количество дней в месяце.

5.3 Написать приложение «Конвертер валют». Курс валют (доллары, евро) задать программно, исходную сумму в рублях и требуемую валюту указывает пользователь. На выходе — сумма в валюте, округленная до двух знаков после запятой. Если введена некорректная валюта, выводить исходную сумму.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C++ и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Для чего используется условный оператор?

- 8.2 Какие существуют формы оператора if и каков их вид?
- 8.3 Для чего используется тернарный оператор?
- 8.4 Для чего используется оператор множественного выбора?
- 8.5 Чем операция «логического И» отличается от операции «логического ИЛИ»?

Лабораторная работа №4

Составление программ циклической структуры

1 Цель работы

1.1 Изучить процесс разработки циклических алгоритмов на языке Си.

2 Литература

2.1 Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C++ / Т. И. Немцова, С. Ю. Голова, А. И. Терентьев. – Москва : Форум, 2019. – URL: <https://ibooks.ru/reading.php?productid=361544>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.5.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Написать программу с использованием цикла с параметром:

- 1 вариант: выводящую на экран значение факториала числа n ($n! = 1 * 2 * 3 * \dots * n$, $0! = 1$). В случае некорректных данных ($n < 0$) требуется вывести на экран сообщение, что факториал не существует.

- 2 вариант: выводящую на экран шкалу температуры по шкалам Цельсия и Фаренгейта в диапазоне от 100 до -50 градусов по шкале Цельсия с шагом 10. В первом столбце – температура по шкале Цельсия, во втором – по шкале Фаренгейта ($F = C * 1.8 + 32$).

- 3 вариант: выводящую на экран делители положительного числа n , включая 1 и само число n , и общее количество делителей числа n .

5.2 Написать программу с использованием цикла с предусловием:

- 1 вариант: вычисляющую результат деления двух чисел (a – делимое, b – делитель). В случае некорректного ввода значения делителя обеспечить его повторный ввод до тех пор, пока не будет введено корректное значение (отличное от нуля). Цикл — для повторного ввода делителя.

- 2 вариант: выводящую на экран значение факториала числа n ($n! = 1 * 2 * 3 * \dots * n$, $0! = 1$). В случае некорректных данных ($n < 0$) требуется вывести на экран сообщение, что факториал не существует.

- 3 вариант: запрашивающую у пользователя сумму, на которую он хочет открыть вклад, и процент годовых. Вывести на экран, через сколько лет он станет миллионером и сумму на вкладе за каждый год. Ежегодно размер вклада увеличивается на указанный процент, на эти деньги в следующем году также будут начислены проценты. Цикл — для расчета вклада за каждый год.

5.3 Написать программу с использованием цикла с постусловием:

- 1 вариант: в которой пользователь пытается угадать число, объявленное программно. Если введено число меньше или больше загаданного, вывести на экран соответствующую надпись («требуется ввести большее число» или «требуется ввести меньшее число») и дать возможность заново угадать (повторять до тех пор, пока не будет названо корректное число). После того, как пользователь угадал, сообщить, что он молодец. Цикл — для ввода числа пользователем.

- 2 вариант: запрашивающую у пользователя рост учеников и выводящую средний рост после того, как пользователь ввел число 0. Цикл — для ввода роста каждого из учеников.

- 3 вариант: вычисляющую результат деления двух чисел (a – делимое, b – делитель). Пользователь должен вводить делитель до тех пор, пока не будет введено

корректное значение (отличное от нуля). Цикл — для ввода делителя.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C++ и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какие операции отношения используются в Си?

8.2 Какие логические операции используются в Си?

8.3 Когда следует применять оператор for?

8.4 В чем отличие оператора while от оператора do-while?

8.5 Какое минимальное количество раз выполняется каждый из операторов цикла?

8.6 Какие операторы досрочного выхода из цикла применяются в Си?

Лабораторная работа №5

Обработка одномерных массивов

1 Цель работы

1.1 Изучить процесс обработки одномерных массивов на языке Си.

2 Литература

2.1 Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C++ / Т. И. Немцова, С. Ю. Голова, А. И. Терентьев. – Москва : Форум, 2019. – URL: <https://ibooks.ru/reading.php?productid=361544>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.8.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Написать программу, в которой пользователь заполняет с клавиатуры одномерный массив из 5 целых чисел.

5.2 Написать программу, выводящую на экран одномерный массив из 10 целых чисел.

5.3 Написать программу, выводящую на экран позиции элементов одномерного массива из 5 целых чисел, значение которого совпадает с указанным пользователем.

5.4 Написать программу, выводящую на экран сумму элементов одномерного массива из 10 вещественных чисел.

5.5 Написать программу, выводящую на экран значение минимального элемента одномерного массива из 10 вещественных чисел.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C++ и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое массив?

8.2 Как описывается одномерный массив?

8.3 Как обратиться к некоторому элементу одномерного массива?

8.4 Как можно задать одномерный массив?

8.5 Как заполнить одномерный массив?

8.6 Как вывести на экран заполненный одномерный массив?

Лабораторная работа №6

Обработка двумерных массивов

1 Цель работы

1.1 Изучить процесс обработки двумерных массивов на языке Си.

2 Литература

2.1 Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C++ / Т. И. Немцова, С. Ю. Голова, А. И. Терентьев. – Москва : Форум, 2019. – URL: <https://ibooks.ru/reading.php?productid=361544>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.10.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Составить программу, в которой требуется заполнить двумерный массив случайными числами и вывести его на экран в виде таблицы:

- 1 вариант: массив вещественных чисел;

- 2 вариант: массив целых чисел.

Размерность массива (количество строк и столбцов) указываются программно.

5.2 Составить программу, в которой требуется заполнить динамический двумерный массив случайными числами и вывести его на экран в виде таблицы:

- 1 вариант: массив целых чисел;

- 2 вариант: массив вещественных чисел;

Размерность массива (количество строк и столбцов) указываются пользователем с клавиатуры.

5.3 Составить программу, в которой требуется обработать данные двумерного массива:

- 1 вариант: найти сумму значений элементов массива, находящихся в столбце, указанном пользователем;

- 2 вариант: заменить все отрицательные значения элементов массива на 0 и вывести полученный массив на экран.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C++ и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Как можно задать двумерный массив?
- 8.2 Как объявить динамический двумерный массив?
- 8.3 Как обратиться к некоторому элементу двумерного массива, используя индексы?
- 8.4 Как обратиться к некоторому элементу двумерного массива, используя указатели?
- 8.5 Как заполнить двумерный массив?
- 8.6 Как вывести на экран заполненный двумерный массив?

Лабораторная работа №7

Работа со строками

1 Цель работы

1.1 Изучить процесс обработки символьных данных средствами языка Си.

2 Литература

2.1 Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C++ / Т. И. Немцова, С. Ю. Голова, А. И. Терентьев. – Москва : Форум, 2019. – URL: <https://ibooks.ru/reading.php?productid=361544>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.9.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Написать программу, выводящую на экран количество символов во введенной пользователем строке и количество символов без учета пробелов.

5.2 Написать программу, выводящую на экран все позиции, на которых во введенной пользователем строке находится указанный пользователем символ. Если совпадение не найдено, сообщить об этом.

5.3 Написать программу, изменяющую регистр букв во введенной строке на указанный пользователем: верхний (ТЕКСТ), нижний (текст).

5.4 Написать программу, разделяющую введенный текст на предложения, каждое должно выводиться на отдельной строке без пробелов по краям и без знаков препинания в конце предложения. Конец предложения определяется символом точка, вопросительный знак, восклицательный знак или их комбинацией.

5.5 Написать программу, определяющую, является ли введенное слово палиндромом.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C++ и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

- 8.1 Что представляет собой строка в языке Си?
- 8.2 Каким символом должна заканчиваться строка?
- 8.3 С помощью каких функций можно считать последовательность символов с клавиатуры?
- 8.4 С помощью каких функций можно вывести последовательность символов на экран?
- 8.5 Как получить доступ к произвольному символу строки?
- 8.6 Какой заголовочный файл необходимо подключить для использования стандартных функций работы со строками?

Лабораторная работа №8

Организация функций

1 Цель работы

- 1.1 Изучить процесс написания функций средствами языка Си.
- 1.2 Изучить процесс написания рекурсивных функций средствами языка Си.

2 Литература

2.1 Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C++ / Т. И. Немцова, С. Ю. Голова, А. И. Терентьев. – Москва : Форум, 2019. – URL: <https://ibooks.ru/reading.php?productid=361544>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.11-12.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание лабораторной работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

Написать и протестировать описанные функции. Все данные в функции передавать через параметры.

5.1 Написать и протестировать функцию с двумя вещественными параметрами. Функция должна возвращать 1, если первое число больше второго; 0, если числа равны; -1, если первое число меньше второго.

5.2 Написать и протестировать в одной программе две функции с одинаковыми именами, но различным списком параметров (т.е. перегруженные функции). Первая функция должна вычислять площадь квадрата по переданной через параметры длине стороны, вторая — площадь прямоугольника по переданным через параметры длине и ширине.

5.3 Написать и протестировать функцию, вычисляющую a^x . Число a — любое, число x — любое целое неотрицательное. Для вычисления использовать цикл. Стандартные функции не использовать.

5.4 Написать и протестировать рекурсивную функцию, вычисляющую a^x . Число a — любое, число x — любое целое. Стандартные функции не использовать.

6 Порядок выполнения работы

- 6.1 Используя Microsoft Visual Studio, создать проект C++ и выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое функция?

8.2 Что такое возвращаемое значение?

8.3 Чем стандартные функции отличаются от функций, определенных пользователем?

8.4 Каков синтаксис объявления функции в общем виде?

8.5 Что такое «рекурсивная функция»?

Лабораторная работа №9

Организация процедур

1 Цель работы

1.1 Научиться выводить данные на html-странице, используя JavaScript.

2 Литература

2.1 Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C++ / Т. И. Немцова, С. Ю. Голова, А. И. Терентьев. – Москва : Форум, 2019. – URL: <https://ibooks.ru/reading.php?productid=361544>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.11-12.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

Написать и протестировать описанные процедуры. Все данные в процедуры передавать через параметры.

5.1 Написать и протестировать процедуру с двумя вещественными параметрами. Процедура должна менять местами значения двух переданных параметров.

5.2 Написать и протестировать процедуру, выводящую в одну строку через пробел элементы переданного в параметрах одномерного массива символов. Количество элементов массива также передавать через параметр.

5.3 Написать и протестировать процедуру, заполняющую случайными числами переданный в параметрах одномерный массив целых чисел. Количество элементов массива также передавать через параметр.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C++ и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое процедура?

8.2 Какой тип возвращаемого значения указывается у процедур на языке Си?

8.3 Какое ключевое слово позволяет вернуться из процедуры?

8.4 Каков синтаксис объявления процедур в языке Си?

Лабораторная работа №10

Работа с текстовыми файлами

1 Цель работы

- 1.1 Научиться проводить отладку JavaScript-сценариев;
- 1.2 Научиться применять средства разработчика в браузере;
- 1.3 Закрепить навык разработки JavaScript-сценариев.

2 Литература

2.1 Ашарина, И. В. Объектно-ориентированное программирование в C++: лекции и упражнения. Учебное пособие для вузов. – 2-е изд., перераб. и доп. Москва : Горячая Линия–Телеком, 2017. – URL: <https://ibooks.ru/bookshelf/359752/reading>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – п.9.2-9.3.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание лабораторной работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Написать программу, запрашивающую у пользователя имя файла. Если файла с указанным именем не существует, сообщить об этом пользователю.

5.2 Изменить приложение из п.5.1, чтобы пользователь после ввода существующего имени файла увидел на экране его содержимое, считанное построчно из файла и выведенное на экран (считать, что в файле содержатся строковые данные).

5.3 Написать программу, дописывающую данные из открытого для чтения файла в файл 1.txt.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C# и выполнить задания из п.5.

- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Как объявить указатель файла в классическом языке Си?
- 8.2 Когда применяют функцию `fopen` в языке Си?
- 8.3 Как открыть файл для добавления информации в классическом языке Си?
- 8.4 Каков порядок закрытия файла в классическом языке Си?

8.5 Как открыть файл для чтения информации в классическом языке Си?

Лабораторная работа №11

Работа с текстовыми файлами с использованием потоков

1 Цель работы

1.1 Изучить процесс обработки текстовых файлов средствами языка C++ с использованием потоков.

2 Литература

2.1 Ашарина, И. В. Объектно-ориентированное программирование в C++: лекции и упражнения. Учебное пособие для вузов. – 2-е изд., перераб. и доп. Москва : Горячая Линия–Телеком, 2017. – URL: <https://ibooks.ru/bookshelf/359752/reading>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.23.

2.2 Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C++ / Т. И. Немцова, С. Ю. Голова, А. И. Терентьев. – Москва : Форум, 2019. – URL: <https://ibooks.ru/reading.php?productid=361544>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.19.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Написать программу, запрашивающую у пользователя имя файла. Если файла с указанным именем не существует, сообщить об этом пользователю.

5.2 Изменить приложение из п.5.1, чтобы пользователь после ввода существующего имени файла увидел на экране его содержимое, считанное построчно из файла и выведенное на экран.

5.3 Написать программу, считывающую из файла 1.txt данные посимвольно. Подсчитать число вхождений в файл каждой из букв a, b, c. Результат вывести в файл 2.txt в виде списка с комментариями.

5.4 Написать программу, считывающую из файла 3.txt набор целых чисел и записывающую в файл 4.txt количество считанных чисел, их сумму и среднее арифметическое (каждое значение в файле 2.txt должно быть с подписью, что означает каждое из записанных чисел).

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C++ и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какие классы используются для работы с файловыми потоками?

8.2 Какие режимы доступа могут использоваться при работе с файлами?

8.3 Какая функция используется для открытия файла?

8.4 Какие операции доступны для работы с файлами?

8.5 Какие функции позволяют определить конец файла при чтении из него информации?

Лабораторная работа №12

Разработка структур

1 Цель работы

1.1 Изучить процесс разработки и применения структур на языке Си.

2 Литература

2.1 Ашарина, И. В. Объектно-ориентированное программирование в C++: лекции и упражнения. Учебное пособие для вузов. – 2-е изд., перераб. и доп. Москва : Горячая Линия–Телеком, 2017. – URL: <https://ibooks.ru/bookshelf/359752/reading>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – п.7.1.

2.2 Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C++ / Т. И. Немцова, С. Ю. Голова, А. И. Терентьев. – Москва : Форум, 2019. – URL: <https://ibooks.ru/reading.php?productid=361544>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – п.13.3.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Разработать структуру по описанию согласно варианту в таблице 1 (вариант = номер ПК). Включить в структуру указанные поля. Типы данных для полей подобрать наиболее подходящие.

5.2 Протестировать разработанную структуру в функции main, создав переменную типа структура и присвоив программно каждому полю структуры значение. После присваивания вывести значения полей структуры на экран.

5.3 Создать в функции main массив из n переменных типа структура. Данные должны быть введены пользователем.

5.4 Выполнить поиск по массиву согласно одному из указанных критериев поиска и вывести соответствующие критерию элементы массива.

Таблица 1 — Варианты описаний структур

| № | Название и поля структуры | Критерии поиска объектов в массиве |
|---|---------------------------------------|--|
| 1 | Студент (ФИО, специальность, курс) | а) студентов заданной специальности; б) студентов указанной специальности и курса |
| № | Название и поля структуры | Критерии поиска объектов в массиве |
| 2 | Абитуриент (ФИО, класс, средний балл) | а) абитуриентов, закончивших 11 класс; |

| | | |
|----|---|---|
| | | б) абитуриентов со средним баллом выше заданного |
| 3 | Авиарейс (пункт назначения, номер рейса, вместимость) | а) рейсы до заданного пункта назначения; б) рейсы с вместимостью больше заданной |
| 4 | Книга (название, автор, цена) | а) книги указанного автора; б) книги, цена которых находится в заданном диапазоне |
| 5 | Работник (ФИО, должность, зарплата) | а) работников, зарплата которых больше заданной; б) работников, занимающих заданную должность |
| 6 | Поезд (номер поезда, пункт назначения, количество свободных мест) | а) поезда, следующие до заданного пункта назначения; б) поезда, имеющие не меньше указанного количества свободных мест |
| 7 | Товар (тип, наименование, цена) | а) товары заданного типа; б) товары, цена которых находится в заданном диапазоне |
| 8 | Пациент (ФИО, номер полиса, год рождения) | а) пациентов указанного года рождения; б) пациентов с указанной ФИО |
| 9 | Питомец (кличка, порода, возраст) | а) питомцев указанной породы; б) питомцев с указанной кличкой |
| 10 | Заказчик (ФИО, адрес, потраченная сумма) | а) заказчиков по указанному адресу; б) заказчиков, потративших больше указанной суммы |
| 11 | Файл (имя файла, полный путь к файлу, размер файла) | а) файлы с указанным названием; б) файлы, размер которых превышает заданный |
| 12 | Статья (название, автор, год публикации) | а) статьи указанного автора; б) статьи, изданные после указанного года |
| 13 | Квартира (адрес, количество комнат, площадь) | а) квартиры, расположенные по указанному адресу; б) квартиры с площадью больше заданной |
| 14 | Абонент (ФИО, номер телефона, время разговоров) | а) абонентов со временем разговоров больше заданного; б) абонентов с указанным номером телефона |

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C++ и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

- 8.1 Что такое «структура»?
- 8.2 Каким образом можно описать пользовательский тип данных структуры?
- 8.3 Как описать переменную структурного типа?
- 8.4 Как обратиться к полю структуры?
- 8.5 Верно ли, что все поля структуры должны быть различных типов?

Лабораторная работа №13

Работа с двоичными файлами

1 Цель работы

1.1 Изучить процесс обработки двоичных файлов средствами языка C++ с использованием потоков.

2 Литература

2.1 Ашарина, И. В. Объектно-ориентированное программирование в C++: лекции и упражнения. Учебное пособие для вузов. – 2-е изд., перераб. и доп. Москва : Горячая Линия–Телеком, 2017. – URL: <https://ibooks.ru/bookshelf/359752/reading>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.24

2.2 Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C++ / Т. И. Немцова, С. Ю. Голова, А. И. Терентьев. – Москва : Форум, 2019. – URL: <https://ibooks.ru/reading.php?productid=361544>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – п.19.8-19.11.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Написать программу, запрашивающую у пользователя имя файла и записывающую в него n случайных целых чисел от 1 до 100 (n вводится пользователем).

5.2 Написать программу, запрашивающую у пользователя имя файла, считывающую из него целые числа до конца файла и выводящую их на экран.

5.3 Написать программу, запрашивающую у пользователя имя файла и записывающую в него 3 переменных типа структура (тип описан в ЛР №12).

5.4 Написать программу, запрашивающую у пользователя имя файла, считывающую из него 3 переменные типа структура и выводящую значения их полей на экран.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C++ и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 В чем преимущества использования двоичных файлов?

8.2 С помощью каких функций можно записывать информацию в двоичные файлы?

8.3 С помощью каких функций можно считывать информацию из двоичных файлов?

8.4 Как считать переменные стандартных типов данных из двоичного файла?

8.5 Как считать переменные структурированных типов данных из двоичного файла?

Лабораторная работа №14

Изучение правил разработки схем алгоритмов

1 Цель работы

- 1.1 Изучить процесс построения схем алгоритмов согласно ГОСТ 19.701-90;
- 1.2 Научиться применять графические редакторы на этапе проектирования программного обеспечения;
- 1.3 Закрепить навык разработки программ методами структурного программирования.

2 Литература

2.1 Ашарина, И. В. Объектно-ориентированное программирование в С++: лекции и упражнения. Учебное пособие для вузов. – 2-е изд., перераб. и доп. Москва : Горячая Линия–Телеком, 2017. – URL: <https://ibooks.ru/bookshelf/359752/reading>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – с.323-328.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание лабораторной работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Разработать схему алгоритма поиска минимального значения из чисел а и b. Значения чисел задаются программно.

5.2 Разработать схему алгоритма поиска и вывода на экран частного двух чисел, вводимых пользователем с клавиатуры. Предусмотреть ввод некорректных данных (требовать повторный ввод делителя, если он равен нулю).

5.3 Разработать схему алгоритма заполнения одномерного массива вводимыми пользователем числами, поиска максимального и минимального из них, среднего арифметического значения массива. Количество элементов массива также вводится пользователем.

6 Порядок выполнения работы

- 6.1 Выполнить задания из п.5, используя графический редактор для построения схем алгоритмов.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

8.1 Какой нормативный документ содержит правила оформления схем алгоритмов?

8.2 Какие вкладки в выбранном графическом редакторе содержат графические примитивы для построения схем алгоритмов?

8.3 Как выполнить выравнивание элементов в выбранном графическом редакторе?

8.4 Как внести изменения в размещенную на схеме в выбранном графическом редакторе соединительную линию, чтобы добавить или удалить отображение стрелки на концах линии?

8.5 Как изменить размеры фигур в выбранном графическом редакторе?

Лабораторная работа №15

Тестирование, отладка приложения

1 Цель работы

1.1 Изучить возможности отладки приложений в MS Visual Studio.

2 Литература

2.1 Тракимус, Ю. В. Разработка консольных приложений с помощью Microsoft Visual Studio 2017: учебное пособие / Ю. В. Тракимус. – Новосибирск : Новосибирский государственный технический университет, 2018. – URL: <https://ibooks.ru/reading.php?productid=367816>. – Режим доступа: для зарегистрированных пользователей. – Текст : электронный. – с.27-37.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Пошаговая отладка и просмотр значений переменных

Создать консольное приложение, содержащее цикл со счетчиком, в котором вычисляется значение факториала f от программно заданного числа n . Результат должен выводиться на экран. Указать $n=6$ и выполнить пошаговую отладку с отслеживанием значений переменных во вкладке Локальные. Реализовать отображение данных во время пошаговой отладки.

5.2 Создание и отключение брейкпоинтов (точек останова)

Создать консольное приложение, дублирующее приложение из п.5.1. Добавить брейкпоинт в конце цикла (после изменения значения переменной f) и запустить отладку с остановками на брейкпоинтах с отслеживанием значений переменных во вкладке Локальные. Сделать имеющийся брейкпоинт неактивным (Ctrl+F9).

5.3 Настройка остановки по условию

Создать консольное приложение, дублирующее приложение из п.5.1. Добавить брейкпоинт в конце цикла (после изменения значения переменной f). Изменить брейкпоинт так, чтобы остановки выполнялись только если i нечетное или $f > 30$. Запустить отладку с остановками на брейкпоинтах с отслеживанием значений переменных во вкладке Локальные.

5.4 Настройка действия у брейкпоинта

Создать консольное приложение, дублирующее приложение из п.5.1. Добавить брейкпоинт в конце цикла (после изменения значения переменной f) и запустить отладку с остановками на брейкпоинтах с отслеживанием значений переменных во вкладке Локальные.

Изменить брейкпоинт так, чтобы при его достижении в отладочную консоль (окно Вывод) выводились значения i и f в следующем виде:

$i! = f$

Образец вывода:

$0! = 1$

$1! = 1$

$2! = 2$

$3! = 6$

5.5 Отладка с заходом в функции

Создать консольное приложение, дублирующее приложение из п.5.1.

Провести рефакторинг, выделив функцию для вычисления факториала из функции `main`. Значение n в основной программе должно запрашиваться у пользователя, функция вычисления факториала должна возвращать число (в случае некорректных данных должен вернуть -1).

Установить брейкпоинт на вызове метода вычисления факториала из функции `main` и выполнить пошаговую отладку с заходом в функцию.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C++ и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «отладка»?

8.2 Какие этапы включает в себя отладка?

8.3 Что такое «точка останова»?

8.4 Как добавить точки останова в Visual Studio?

8.5 Как в Visual Studio запустить приложение в режиме отладки?

8.6 Какие окна в Visual Studio отображают значения переменных и для чего предназначено каждое из этих окон?

8.7 Какие комбинации «горячих» клавиш используются в Visual Studio для пошагового прохождения?

Лабораторная работа №16

Программирование модуля

1 Цель работы

1.1 Изучить процесс создания и применения модулей в программах на языке Си.

2 Литература

2.1 Ашарина, И. В. Объектно-ориентированное программирование в C++: лекции и упражнения. Учебное пособие для вузов. – 2-е изд., перераб. и доп. Москва : Горячая Линия–Телеком, 2017. – URL: <https://ibooks.ru/bookshelf/359752/reading>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.11.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Создать свой модуль, содержащий функцию для нахождения суммы двух вещественных чисел. Проверить работу созданной функции, подключив модуль к основной программе и вызвав созданную функцию в функции main.

5.2 Добавить в созданный модуль две функции (первая — для нахождения максимума, вторая — минимума из двух вещественных чисел). Проверить работу функций, вызвав их в функции main для определения максимума и минимума из трех введенных пользователем чисел.

5.3 Добавить в созданный модуль функцию для нахождения a^x (a – вещественное число, x – целое). Стандартную функцию для возведения в степень не использовать. Проверить работу созданной функции, вызвав ее в функции main.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C++ и выполнить задания из п.5. Прототипы функций должны быть размещены в заголовочном файле, реализация — в отдельном файле `cpp`, к основному модулю должен быть подключен файл с прототипом.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Каково назначение модуля программиста?

8.2 Из каких файлов состоит модуль программиста и для чего предназначен каждый из типов файлов?

8.3 Как подключить модуль программиста к основной программе?

Лабораторная работа №17

Создание библиотеки подпрограмм

1 Цель работы

1.1 Изучить процесс создания и применения библиотек в программах на языке Си.

2 Литература

2.1 Ашарина, И. В. Объектно-ориентированное программирование в C++: лекции и упражнения. Учебное пособие для вузов. – 2-е изд., перераб. и доп. Москва : Горячая Линия–Телеком, 2017. – URL: <https://ibooks.ru/bookshelf/359752/reading>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.11.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание лабораторной работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Создать свою библиотеку, содержащую две функции (первая — для нахождения суммы, вторая — для нахождения разности двух вещественных чисел, переданных через параметры).

5.2 Создать новый проект. Подключить созданную библиотеку к проекту и проверить работу созданных функций, вызвав их в функции main.

5.3 Добавить в созданную библиотеку функцию, вычисляющую факториал числа n с использованием рекурсии. Если $n < 0$, возвращать -1. Проверить работу созданной функции, вызвав ее в функции main.

5.4 Добавить в созданную библиотеку функцию, возвращающую номер индекса, на котором в строке находится переданный в параметрах символ.

Заголовок функции должен быть следующим:

```
int poisk(char stroka[], int dlina, char simvol)
```

Проверить работу созданной функции, вызвав ее в функции main. Перед вызовом функции создать в main массив из 80 символов, считать его с клавиатуры и передать в параметрах функции (второй параметр — длина = 80, третий параметр — символ — любой, заданный программно в основной программе).

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C++ и выполнить задания из п.5. Прототипы функций должны быть размещены в заголовочном файле библиотеки, реализация — в отдельном файле `.cpp` библиотеки. К проекту TestProject подключить библиотеку и файл с заголовком библиотеки и протестировать в консольном приложении все созданные функции.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Каково назначение библиотеки?
- 8.2 Какое расширение может быть у библиотек?
- 8.3 Для чего предназначена спецификатор `__declspec`?
- 8.4 Что должно быть написано в верхней части заголовочного файла библиотеки?
- 8.5 Что надо написать перед объявлением типа функции в заголовочном файле библиотеки?

Лабораторная работа №18

Работа с алгоритмами сортировки массивов

1 Цель работы

1.1 Изучить принципы реализации алгоритмов сортировки массивов средствами языка программирования Си.

2 Литература

2.1 Ашарина, И. В. Объектно-ориентированное программирование в С++: лекции и упражнения. Учебное пособие для вузов. / И. В. Ашарина. – 2-е изд., перераб. и доп. – Москва : Горячая Линия–Телеком, 2017. – 336 с. – URL: <https://ibooks.ru/reading.php?productid=359752>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – п.6.3.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Реализовать и протестировать в консольном приложении алгоритм сортировки массивов методом пузырька.

5.2 Реализовать и протестировать в консольном приложении алгоритм сортировки массивов методом выбора минимального элемента.

5.3 Реализовать и протестировать в консольном приложении алгоритм сортировки массивов простым включением.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект С# и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что означает «сортировка массива»?

8.2 Какие виды сортировки массива существуют?

8.3 Как обменивать значениями элементы массива?

8.4 Как обратиться к элементу одномерного массива?

8.5 Как обратиться к элементу одномерного динамического массива?

Лабораторная работа №19

Использование указателей для организации связанных списков

1 Цель работы

1.1 Изучить процесс применения указателей для организации списков в программах на языке Си.

2 Литература

2.1 Ашарина, И. В. Объектно-ориентированное программирование в C++: лекции и упражнения. Учебное пособие для вузов. / И. В. Ашарина. – 2-е изд., перераб. и доп. – Москва : Горячая Линия–Телеком, 2017. – 336 с. – URL: <https://ibooks.ru/reading.php?productid=359752>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.15.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание лабораторной работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Создать программу, в которой описать структуру типа «стек» и реализовать заполнение стека N элементами (все данные вводятся пользователем).
- 5.2 Реализовать вывод стека из N элементов на экран.
- 5.3 Реализовать извлечение из списка X элементов (X указывается пользователем).
- 5.4 Реализовать вычисление среднего арифметического всех элементов списка.
- 5.5 Реализовать вывод всех элементов стека на экран (количество элементов в стеке неизвестно).

6 Порядок выполнения работы

- 6.1 Используя Microsoft Visual Studio, создать проект C++ и выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Что такое динамические структуры?
- 8.2 Какие виды динамических структур существуют?
- 8.3 На что указывает указатель на структуру в самоссылочных структурах?

8.4 Почему динамические структуры называются самоссылочными?

8.5 По какому принципу организованы стек и очередь?

Лабораторная работа №20

Изучение процесса разработки классов на C++

1 Цель работы

- 1.1 Изучить процесс разработки классов на C++;
- 1.2 Изучить реализацию механизмов полиморфизма и инкапсуляции на языке C++;
- 1.3 Закрепить навык разработки программ методами структурного, процедурного и модульного программирования.

2 Литература

2.1 Ашарина, И. В. Объектно-ориентированное программирование в C++: лекции и упражнения. Учебное пособие для вузов. / И. В. Ашарина. – 2-е изд., перераб. и доп. – Москва : Горячая Линия–Телеком, 2017. – 336 с. – URL: <https://ibooks.ru/reading.php?productid=359752>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.18.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание лабораторной работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Создать класс Time для хранения информации о времени.

Добавить в класс три закрытых поля hours, minutes, seconds для хранения составляющих времени (часы, минуты, секунды).

Добавить в класс следующий набор открытых конструкторов:

- конструктор по умолчанию для задания стандартного времени 12:00:00;
- конструктор с тремя параметрами для задания значений всем полям класса.

Если значения параметров переданы некорректно, вместо некорректных данных подставлять 0 (часы не могут быть отрицательными, минуты и секунды должны быть в диапазоне от 0 до 59);

- конструктор с одним параметром seconds для задания значений всем полям класса (например, при значении 12108 время будет 3:21:48).

Протестировать полученный класс, создав объекты класса с использованием конструктора по умолчанию и конструкторов с параметрами. Данные для конструкторов задать программно.

- 5.2 Добавить в класс следующие элементы:

- метод Print для вывода информации о времени на консоль в формате h:mm:ss
- метод GetFullSeconds для возврата общего количества секунд в текущем значении времени (например, для значений 1:15:03 вернет 4503);

- перегруженный оператор индексации, возвращающий для индекса 1 значение hours, для индекса 2 значение minutes, для индекса 3 значение seconds, для всех остальных индексов значение -1.

Протестировать работу методов и оператора, вызвав их у созданных объектов класса.

5.3 Создать массив объектов класса, при создании объектов вызывать конструктор с тремя параметрами, значения должны вводиться пользователем.

Вывести информацию обо всех элементах массива на консоль, используя метод Print.

Выполнить поиск по массиву согласно одному из указанных критериев поиска (по вариантам):

1) вывести только те элементы массива, которые лежат в диапазоне между двумя значениями часов, указанных пользователем.

2) вывести только те элементы массива, которые позже часа и минуты, указанных пользователем.

3) вывести только те элементы массива, которые находятся за пределами диапазона между двумя значениями часов, указанных пользователем.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C++ и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «конструктор», «конструктор по умолчанию», «метод»?

8.2 Что такое «полиморфизм»?

8.3 Что такое «перегрузка метода»?

8.4 Что такое «инкапсуляция»?

8.5 Какие модификаторы доступа применяются в C++ и каково их назначение?

Лабораторная работа №21

Изучение процесса разработки дочерних классов на C++

1 Цель работы

- 1.1 Изучить процесс разработки дочерних классов на C++;
- 1.2 Изучить реализацию механизмов наследования, инкапсуляции и полиморфизма на языке C++;
- 1.3 Закрепить навык разработки программ методами структурного, процедурного и модульного программирования.

2 Литература

2.1 Ашарина, И. В. Объектно-ориентированное программирование в C++: лекции и упражнения. Учебное пособие для вузов. / И. В. Ашарина. – 2-е изд., перераб. и доп. – Москва : Горячая Линия–Телеком, 2017. – 336 с. – URL: <https://ibooks.ru/reading.php?productid=359752>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.21.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание лабораторной работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Создание родительского и дочернего классов

Создать родительский класс Person (человек) с закрытым строковым полем name (имя) и защищенным числовым полем age (возраст) для хранения информации о человеке.

Добавить в класс Person следующие открытые конструкторы:

- конструктор по умолчанию для задания стандартных значений полям класса;
- конструктор с двумя параметрами для задания значений всем полям класса.

Создать класс Worker (работник) с закрытыми полями seniority (стаж) и salary (зарплата). Сделать его наследником класса Person.

Добавить в класс Worker следующие открытые конструкторы:

- конструктор по умолчанию для задания стандартных значений полям класса;
- конструктор с четырьмя параметрами для задания значений всем полям класса.

После списка параметров вызвать родительский конструктор с параметрами.

Протестировать полученный дочерний класс, создав объекты класса с использованием конструктора по умолчанию и конструктора с параметрами. Данные для конструктора задать программно.

- 5.2 Создание методов в дочернем классе

Добавить в родительский класс Person открытый виртуальный метод Print, выводящий на экран информацию об объекте.

Добавить в дочерний класс `Worker` следующие открытые методы:

- метод `Print`, вызывающий реализацию метода `Print` из родительского класса и расширяющий его функциональность выводом информации о полях, объявленных в классе `Worker`;

- метод `GetStartWorkAge`, возвращающий возраст, в котором человек устроился на работу (находится вычитанием стажа из возраста человека).

Протестировать работу методов, вызвав их у созданных объектов класса.

5.3 Создать абстрактный класс `Фигура`, в котором объявить следующие чисто виртуальные функции:

- функцию для нахождения площади фигуры;
- функцию для нахождения периметра фигуры;
- функцию для вывода названия типа фигуры.

Создать два класса-наследника класса `Фигура`, в которых реализовать указанные функции и добавить требуемые поля (по вариантам):

1) классы `Квадрат` (поле `сторона`) и `Прямоугольник` (поля `длина` и `ширина`).

2) классы `Круг` (константа `Pi` и поле `радиус`) и `Прямоугольник` (поля `длина` и `ширина`).

3) классы `Квадрат` (поле `сторона`) и `Круг` (константа `Pi` и поле `радиус`).

Протестировать работу методов, создав объекты классов-наследников и вызвав переписанные методы.

6 Порядок выполнения работы

6.1 Используя `Microsoft Visual Studio`, создать проект `C++` и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «наследование»?

8.2 Что такое «множественное наследование»?

8.3 Какие классы называются родительскими?

8.4 Что такое «виртуальный метод»?

8.5 Какие классы называются абстрактными?

Лабораторная работа №22

Изучение процесса разработки шаблонов функций и классов на C++

1 Цель работы

- 1.1 Изучить процесс разработки шаблонов функций и классов на C++;
- 1.2 Изучить реализацию параметрического полиморфизма на языке C++;
- 1.3 Закрепить навык разработки программ методами структурного, процедурного и модульного программирования.

2 Литература

2.1 Ашарина, И. В. Объектно-ориентированное программирование в C++ : лекции и упражнения. Учебное пособие для вузов. / И. В. Ашарина. – 2-е изд., перераб. и доп. – Москва : Горячая Линия–Телеком, 2017. – 336 с. – URL: <https://ibooks.ru/reading.php?productid=359752>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.25.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание лабораторной работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Разработать шаблон функции Maximum для поиска и возврата максимального значения из двух переданных в параметрах.

Протестировать функцию, вызвав ее с значениями различных типов данных. Значения задать программно в основной программе.

5.2 Разработать шаблон функции IndexOfMax для поиска и возврата индекса элемента массива с максимальным значением. В функцию передается массив и его размерность:

```
template <typename T>
int IndexOfMax(T *array, const int size)
{
    // тело функции
}
```

Протестировать функцию, вызвав ее с значениями различных типов данных. Значения задать программно в основной программе, например:

```
int a[5] = {10, 34, 167, 12, -5};
```

5.3 Создать шаблон класса List (список, в котором по умолчанию 10 элементов, но можно передать другую размерность), в котором объявить закрытые поля array (массив) и count (количество элементов в списке):

```
template <typename T, int size=10>
class List
```



```
{
    T array[size];
    int count;
};
```

Добавить в шаблон класса следующие открытые элементы:

- конструктор по умолчанию, в котором присвоить count значение 0;
- метод int Count(), возвращающий количество элементов в списке.

Протестировать шаблон класса, создав объекты класса и вызвав разработанные функции.

5.4 Добавить в шаблон класса List метод void Add(T value), дописывающий в конец списка переданное в параметрах значение и увеличивающий на 1 значение поля count. При реализации учесть, что нельзя добавлять элементы, если size и count совпадают.

Протестировать разработанный метод, добавив в один из объектов типа список не менее 5 элементов.

5.5 Добавить в шаблон класса List следующий метод (по вариантам) и протестировать его:

1) IndexOf(T value), возвращающий индекс первого вхождения указанного элемента в список. Если совпадение не найдено, возвращать -1.

2) CountOf(T value), возвращающий количество вхождений указанного элемента в список.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C++ и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Для чего применяются шаблоны в C++?

8.2 Какова общая форма шаблона функции?

8.3 Какова общая форма шаблона класса?

8.4 Как передать в шаблон несколько обобщенных типов?

8.5 Как задаются значения параметров шаблона при создании объекта?

Лабораторная работа №23

Изучение среды разработки

1 Цель работы

1.1 Изучить процесс разработки оконных приложений на языке C# в среде Visual Studio;

1.2 Изучить способы настройки свойств формы и процесс обработки событий.

2 Литература

2.1 Фленов, М.Е. Библия C#. – 3 изд.– Санкт-Петербург: БХВ-Петербург, 2016. – Режим доступа: <https://ibooks.ru/reading.php?productid=353561>, только для зарегистрированных пользователей. – Загл. с экрана. – п.5.1-5.5, гл.11.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Настройка главной формы приложения

5.1.1 Перейти к дизайнеру формы, дважды нажав на MainForm.cs в обозревателе решений. Во вкладке Свойства (Properties) задать следующие свойства формы:

- Text (заголовок формы): Главная форма;
- StartPosition (положение при запуске): CenterScreen (по центру экрана);
- Size (размер): ширина: 250, высота: 350;
- BackColor (цвет фона): любой из вкладки Другой или Интернет;

5.1.2 В дизайнера формы во вкладке События (Events) создать обработчики для следующих событий:

- Loady);
- FormClosing.

В обоих обработчиках требуется использовать окно сообщений MessageBox.

При загрузке формы в окне сообщений с заголовком «Приветствие» должно выводиться сообщение «Добро пожаловать» и текущие дата и время.

Перед закрытием формы в окне сообщений с заголовком «Закрытие» и иконкой с вопросительным знаком требуется запрашивать подтверждение закрытия. Если пользователь нажимает «Нет», форма должна оставаться открытой.

5.1.4 Добавить на форму 3 кнопки с подписями «Задание 2», «Задание 3», «Задание 4».

5.2 Настройка формы-образца (будет родительской для форм, созданных в заданиях 3-4)

5.2.1 Добавить в обозревателе решений в проект форму Windows. Реализовать ее открытие при нажатии на кнопку «Задание 2» в главной форме. Настроить у формы в дизайнера следующие свойства:

- заголовок формы: Лабораторная работа №23;
- положение при запуске: CenterScreen;
- шрифт: любой, поддерживающий русский язык и отличный от стандартного;
- цвет фона: белый;
- минимальный размер: ширина: 300, высота: 250;
- иконка формы: выбрать файл с расширением *.ico.

5.2.2 На форму добавить из вкладки Панель элементов элемент Panel, настроить у нее следующие свойства:

- цвет фона: любой из вкладки Другой или Интернет;
- высота – любое число от 50 до 100.
- размещение: по верхнему краю.

5.2.3 Добавить на панель из вкладки Панель элементов элемент PictureBox в левую часть панели, настроить у него следующие свойства:

- изображение: выбрать растровый рисунок;
- режим подбора размера изображения: масштабирование.

5.3 Создание частично прозрачной формы, отображающей дату

5.3.1 Добавить в приложение новую форму. Реализовать открытие созданной формы из главной формы в диалоговом режиме при нажатии на кнопку «Задание 3»

5.3.2 Изменить у формы родительскую форму на форму из п.5.2.

5.3.3 В дизайнере настроить у формы свойство TransparencyKey, указав в нем белый цвет.

5.3.4 Создать обработчик загрузки формы и выполнить в обработчике изменение текста у метки на Задание №3.

5.3.5 В правый нижний угол добавить элемент Label, который должен отображать текущую дату. Выполнить привязку элемента к правому нижнему углу формы. Сделать текст метки полужирным.

5.4 Создание формы авторизации

5.4.1 Добавить в приложение новую форму. Реализовать открытие созданной формы из главной формы в диалоговом режиме при нажатии на кнопку «Задание 4»

5.4.2 Изменить у формы родительскую форму на форму из п.5.2.

5.4.3 Настроить у формы максимальный размер: ширина: 300, высота: 250.

5.4.4 Создать обработчик загрузки формы и выполнить в обработчике изменение текста у метки на Задание №4.

5.4.5 Добавить на форму два поля ввода с подписями в метках Логин и Пароль и кнопку «Авторизоваться».

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C# и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Для чего используется класс Application?

8.2 Какой метод является точкой входа в приложение?

8.3 Какой метод запускает главную форму приложения?

8.4 Какой класс является родительским для всех форм?

8.5 Какие файлы содержат описание формы и что описывается в каждом из этих файлов?

8.6 Какой метод содержит в себе описание настроек формы, сделанных в режиме дизайнера и где он вызывается по умолчанию?

8.7 Вызов каких методов выполняет открытие формы в диалоговом и недиалоговом режимах?

8.8 Вызов какого метода выполняет закрытие формы?

8.9 В каком обработчике событий формы можно отменить ее закрытие?

Лабораторная работа №24

Создание проекта с использованием компонентов для работы с текстом

1 Цель работы

1.1 Изучить различные типы полей ввода, применяющихся в оконных приложениях на С#;

1.2 Изучить свойства полей ввода и процесс обработки событий полей ввода.

2 Литература

2.1 Фленов, М.Е. Библия С#. – 3 изд.– Санкт-Петербург: БХВ-Петербург, 2016. – Режим доступа: <https://ibooks.ru/reading.php?productid=353561>, только для зарегистрированных пользователей. – Загл. с экрана. – п.5.7.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Создание формы с полями ввода

5.1.1 Создать форму регистрации пользователя. Настроить ее заголовок.

5.1.2 Добавить на форму поля ввода для того, чтобы пользователь ввел:

- логин;
- пароль;
- подтверждение пароля;
- страну;
- информацию о себе.

Снабдить поля ввода метками, чтобы пользователь понимал, что он должен ввести.

5.1.3 Настроить в панели свойств следующие свойства полей ввода:

- UseSystemPasswordChar (скрывать вводимые символы): true (для полей ввода пароля и подтверждения пароля);

У поля ввода информации о себе настроить свойство Multiline: true и увеличить его высоту.

5.1.4 Добавить на форму две кнопки: «Зарегистрироваться» и «Отмена».

При нажатии на «Отмена» форма должна закрываться.

5.2 Настройка размерности и привязки полей ввода к краям формы

Настроить у полей ввода максимально допустимую длину строки и привязку к краям формы.

5.3 Проверка заполнения полей ввода

При нажатии на «Зарегистрироваться» следует проверять на корректность введенные пользователем данные:

- логин не должен быть пустым или состоящим из пробельных символов;
- пароль и подтверждение пароля должны совпадать.

Если данные корректны, то нужно вывести окно сообщений с текстом «Логин, вы успешно зарегистрированы» (вместо Логин должно отображаться введенное пользователем имя).

5.4 Изменение внешнего вида полей ввода

Если данные некорректны, то нужно вывести окно сообщений с текстом «Ошибка заполнения»:

- если логин не заполнен или заполнен пробельными символами, менять фон поля ввода логина на красный;
- если пароль не совпадает с подтверждением пароля, менять фон поля ввода подтверждения пароля на красный.

5.5 Обработка событий полей ввода

5.5.1 При изменении текста в поле ввода его фон должен меняться на белый.

5.5.2 В поле ввода страны запретить ввод любых символов кроме букв и BackSpace.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C# и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Для чего используется элемент управления TextBox?

8.2 Для чего используется элемент управления Label?

8.3 Когда срабатывает событие TextChanged и в каких элементах управления оно поддерживается?

8.4 Какие события обрабатывают нажатия клавиш?

Лабораторная работа №25

Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени

1 Цель работы

1.1 Изучить различные типы полей ввода, применяющихся в оконных приложениях на C#;

1.2 Изучить свойства полей ввода и процесс обработки событий полей ввода.

2 Литература

2.1 Фленов, М.Е. Библия C#. – 3 изд.– Санкт-Петербург: БХВ-Петербург, 2016. – Режим доступа: <https://ibooks.ru/reading.php?productid=353561>, только для зарегистрированных пользователей. – Загл. с экрана. – п.5.7.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Добавить на форму поле ввода DateTimePicker для указания даты рождения. Снабдить его меткой, чтобы пользователь понимал, что он должен ввести. Для поля ввода даты рождения на загрузку формы программно задать текущую дату как максимальное значение.

5.2 В приложении из п.5.1 при изменении значения в поле ввода даты должен выполняться подсчет возраста пользователя и выводиться в метке справа от поля ввода даты рождения.

5.3 Добавить на форму бегунок для ввода числовой информации для того, чтобы пользователь мог указать максимальную цену товара (от 0 до 100000). Снабдить бегунок меткой, чтобы пользователь понимал, что он должен ввести. При изменении значения в бегунке полученное значение должно отображаться в поле ввода справа от бегунка.

5.4 Добавить на форму поля ввода числовой информации для того, чтобы пользователь мог указать требуемые длину и ширину прямоугольника. Снабдить поля ввода метками, чтобы пользователь понимал, что он должен ввести. Настроить у полей ввода количество знаков после запятой, минимальное и максимальное допустимые значения, оптимально подходящие для решения задачи. При изменении значений в полях ввода в метку должны выводиться результаты вычисления площади и периметра, округленные до двух знаков после запятой.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C# и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Для чего используется элемент управления TrackBar?

8.2 Для чего используется элемент управления NumericUpDown?

8.3 Для чего используется элемент управления DateTimePicker?

8.4 Когда срабатывает событие ValueChanged и в каких элементах управления оно поддерживается?

Лабораторная работа №26

Создание проекта с использованием компонентов выбора

1 Цель работы

1.1 Изучить свойства и процесс обработки событий элементов выбора в приложении на C#.

2 Литература

2.1 Фленов, М.Е. Библия C#. – 3 изд.– Санкт-Петербург: БХВ-Петербург, 2016. – Режим доступа: <https://ibooks.ru/reading.php?productid=353561>, только для зарегистрированных пользователей. – Загл. с экрана. – п.5.7.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

Создать оконное приложение с главной формой и дополнительными формами (на каждой выполнено одно из заданий). Реализовать переход к этим формам с главной формы при нажатии на соответствующие кнопки. Задание выполняется по вариантам (1 вариант — нечетный № ПК, 2 вариант — четный № ПК).

5.1 Применение CheckBox

1 вариант:

Создать форму авторизации пользователя. Настроить ее заголовок. Добавить на форму поля ввода для того, чтобы пользователь ввел логин и пароль и кнопку «Авторизоваться» (авторизацию реализовывать не нужно). Снабдить поля ввода метками, чтобы пользователь понимал, что он должен ввести. Добавить на форму CheckBox с подписью «Показать пароль». Если флажок отмечен, пароль должен отображаться, иначе – скрываться за звездочками.

2 вариант:

Создать форму подписки на рассылку. Настроить ее заголовок. Добавить на форму поля ввода для того, чтобы пользователь ввел логин и email и кнопку «Подписаться» (подписку реализовывать не нужно). Снабдить поля ввода метками, чтобы пользователь понимал, что он должен ввести. Добавить на форму CheckBox с подписью «Я согласен на получение рассылки». Если флажок отмечен, кнопка «Подписаться» должна быть доступной для нажатия, иначе – недоступной, но видимой.

5.2 Применение RadioButton

1 вариант:

Создать форму для вычислений. Настроить ее заголовок. Добавить на форму два поля ввода для того, чтобы пользователь ввел два числа, и кнопку «Вычислить». Снабдить поля ввода метками, чтобы пользователь понимал, что он должен ввести.

Добавить на форму 4 элемента RadioButton с подписями «+» «-» «*» «/». При нажатии на кнопку требуется выводить результат вычисления выбранной операции в метку. По умолчанию должно быть выбрано сложение.

2 вариант:

Создать форму для перевода длины в метры. Настроить ее заголовок. Добавить на форму поле ввода для того, чтобы пользователь ввел исходное значение, и кнопку «Перевести». Снабдить поле ввода меткой, чтобы пользователь понимал, что он должен ввести. Добавить на форму 4 элемента RadioButton с подписями «дюймы», «километры», «сантиметры», «футы» для выбора исходной единицы измерения. При нажатии на кнопку требуется выводить результат перевода из указанной единицы измерения в метры в метку. По умолчанию должны быть выбрано «километры».

5.3 Применение DateTimePicker с флажком

Создать форму для заказа билетов на самолет с двумя полями ввода города вылета и города прилета и поля ввода дат (дата вылета «Туда» и дата вылета «Обратно»). Снабдить все поля ввода метками, чтобы пользователь понимал, что он должен ввести.

У второго поля ввода даты указать свойство ShowCheckBox: true, чтобы можно было указать, требуется ли обратный билет.

Добавить на форму кнопку «Заказать», при нажатии на которую нужно выводить информацию о заказе в следующем виде:

Билет: дд.мм.гггг Город1 - Город2

Обратный билет: дд.мм.гггг Город2 - Город1

Если флажок у даты вылета «Обратно» не отмечен, то не выводить информацию об обратном билете.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C# и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое CheckBox и для чего он используется?

8.2 Что такое RadioButton и для чего он используется?

8.3 Когда срабатывает событие CheckedChanged и в каких элементах управления оно поддерживается?

8.4 Какое свойство является основным у элементов выбора?

Лабораторная работа №27

Создание проекта с использованием компонентов отображения списков

1 Цель работы

1.1 Изучить свойства и процесс обработки событий элементов выбора и элементов отображения списков в приложении на C#.

2 Литература

2.1 Фленов, М.Е. Библия C#. – 3 изд.– Санкт-Петербург: БХВ-Петербург, 2016. – Режим доступа: <https://ibooks.ru/reading.php?productid=353561>, только для зарегистрированных пользователей. – Загл. с экрана. – п.5.7.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

Создать оконное приложение с главной формой и дополнительными формами (на каждой выполнено одно из заданий). Реализовать переход к этим формам с главной формы при нажатии на соответствующие кнопки.

5.1 Применение ComboBox

Создать форму для заказа билетов с выпадающим списком ComboBox со стилем DropDownStyle: DropDownList для выбора пункта назначения. Снабдить список меткой, чтобы пользователь понимал, что он должен ввести.

Добавить на форму кнопку «Заказать», при нажатии на которую нужно выводить информацию о заказе в следующем виде:

Билет: Архангельск — Пункт назначения

5.2 Применение CheckedListBox

Создать форму записи на курсы. На форме должны быть:

- CheckedListBox для выбора курсов (на выбор должно быть предоставлено 10 наименований, значения указать в дизайнера);

- кнопка «Записаться».

При нажатии на кнопку должно выводиться сообщение (вместо многоточий – название выбранных курсов):

«Вы выбрали следующие курсы:

- ...;

- ...;

- ...»

Если пользователь не выбрал ни одно наименование курса, сообщить, что эти данные обязательны для заполнения.

5.3 Применение ListBox

Создать форму, добавить на нее два элемента ListBox. Заполнить списки программно различными значениями (по 5 значений в каждом).

Добавить на форму поле ввода и кнопку «Добавить». При нажатии на кнопку значение из поля ввода, если оно не пустое, должно заноситься в первый список.

5.4 Применение ListBox

Добавить на форму из п.5.3 кнопки с подписями «>» (перенести выделенные элементы во второй список) и «>>» (перенести все элементы во второй список).

Реализовать функциональность кнопок с подписями «>» и «>>». В зависимости от кнопки выделенные или все элементы из первого списка должны быть удалены, во второй – добавлены.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C# и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое ListBox и для чего он используется?

8.2 Что такое ComboBox и для чего он используется?

8.3 Что такое CheckedListBox и для чего он используется?

8.4 Когда срабатывает событие SelectedIndexChanged и в каких элементах управления оно поддерживается?

8.5 Когда срабатывает событие CheckedChanged и в каких элементах управления оно поддерживается?

Лабораторная работа №28

Изучение способов представления и обработки табличных данных в приложениях на С#

1 Цель работы

1.1 Изучить свойства и процесс обработки событий элемента управления DataGridView в приложении на С#;

1.2 Изучить процесс привязки данных к элементу отображения табличных данных.

2 Литература

2.1 Горелов, С. В. Современные технологии программирования: разработка Windows-приложений на языке С# : учебник : в 2 т. Т. II . / С. В. Горелов. – Москва: Прометей, 2019. – 378 с. – URL: <https://ibooks.ru/reading.php?productid=365815>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. –гл.2-3.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

Создать оконное приложение с главной формой и дополнительными формами (на каждой выполняется определенное задание или задания по описанию). Реализовать переход к этим формам с главной формы при нажатии на соответствующие кнопки.

5.1 Программное заполнение DataGridView

Добавить в приложение новую форму. Добавить на нее DataGridView. Запретить добавление, правку и удаление записей в режиме дизайнера. Изменить фон DataGridView на белый и установить ширину столбцов такой, чтобы они заполняли всю сетку.

Добавить в приложение новую форму. Добавить на нее DataGridView. Запретить добавление, правку и удаление записей в режиме дизайнера. Изменить фон DataGridView на белый и установить ширину столбцов такой, чтобы они заполняли всю сетку.

При загрузке формы программно заполнить сетку данными согласно описанию варианта в таблице 1 (образец заполнения показан на рисунках 1-2). Для программно заполнения заголовочных ячеек использовать следующие свойства:

- сетка.Columns[номер столбца].HeaderText – заголовок столбца
- сетка.Rows[номер строки].HeaderCell.Value – заголовок строки

При нажатии на ячейку таблицы в окне сообщений должно выводиться выражение и результат его вычисления (пример показан в последнем столбце таблицы 1).

Таблица 1 — Варианты таблиц для задания 2

| № | Содержимое сетки | Подписи столбцов | Подписи строк | Пример текста в окне сообщений |
|---|-------------------|------------------|------------------|--------------------------------|
| 1 | Таблица квадратов | Единицы | Десятки | $25 \times 25 = 625$ |
| 2 | Таблица умножения | Первый множитель | Второй множитель | $4 \times 7 = 28$ |

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|------|------|------|------|------|------|------|------|------|
| 1 | 121 | 144 | 169 | 196 | 225 | 256 | 289 | 324 | 361 |
| 2 | 441 | 484 | 529 | 576 | 625 | 676 | 729 | 784 | 841 |
| 3 | 961 | 1024 | 1089 | 1156 | 1225 | 1296 | 1369 | 1444 | 1521 |
| 4 | 1681 | 1764 | 1849 | 1936 | 2025 | 2116 | 2209 | 2304 | 2401 |
| 5 | 2601 | 2704 | 2809 | 2916 | 3025 | 3136 | 3249 | 3364 | 3481 |
| 6 | 3721 | 3844 | 3969 | 4096 | 4225 | 4356 | 4489 | 4624 | 4761 |
| 7 | 5041 | 5184 | 5329 | 5476 | 5625 | 5776 | 5929 | 6084 | 6241 |
| 8 | 6561 | 6724 | 6889 | 7056 | 7225 | 7396 | 7569 | 7744 | 7921 |
| 9 | 8281 | 8464 | 8649 | 8836 | 9025 | 9216 | 9409 | 9604 | 9801 |

Рисунок 1 – Таблица квадратов

| | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|----|----|----|----|----|----|----|----|
| 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 |
| 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 |
| 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36 |
| 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 |
| 6 | 12 | 18 | 24 | 30 | 36 | 42 | 48 | 54 |
| 7 | 14 | 21 | 28 | 35 | 42 | 49 | 56 | 63 |
| 8 | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 72 |
| 9 | 18 | 27 | 36 | 45 | 54 | 63 | 72 | 81 |

Рисунок 2 – Таблица умножения

5.2 Привязка данных к DataGridView

Добавить в приложение класс согласно описанию варианта в таблице 2. Класс и свойства должны быть открытыми.

Добавить в приложение новую форму. Добавить на нее DataGridView и привязать его к объекту типа разработанного класса. Закрепить DataGridView в родительском контейнере. Изменить фон DataGridView на белый и установить ширину столбцов такой, чтобы отображалось содержимое всех ячеек и заголовков. Используя правку столбцов, изменить заголовки столбцов на русскоязычный вариант.

Создать в форме поле List<Класс> и на загрузку формы программно заполнить его пятью строчками. Данные должны быть соответствующими описанию класса.

При загрузке формы привязать DataGridView к созданному и заполненному списку.

Таблица 2 — Варианты описаний классов для задания 3

| № | Название и свойства класса |
|-------|---|
| 1, 15 | Студент (ФИО, специальность, курс) |
| 2, 16 | Абитуриент (ФИО, класс, средний балл) |
| 3, 17 | Авиарейс (пункт назначения, номер рейса, вместимость) |
| 4, 18 | Книга (название, автор, цена) |
| 5, 19 | Работник (ФИО, должность, зарплата) |
| 6, 20 | Поезд (номер поезда, пункт назначения, количество свободных мест) |

| | |
|--------|---|
| 7, 21 | Товар (тип, наименование, цена) |
| 8, 22 | Пациент (ФИО, номер полиса, год рождения) |
| 9, 23 | Питомец (кличка, порода, возраст) |
| 10, 24 | Заказчик (ФИО, адрес, потраченная сумма) |
| 11, 25 | Файл (имя файла, полный путь к файлу, размер файла) |
| 12, 26 | Статья (название, автор, год публикации) |
| 13, 27 | Квартира (адрес, количество комнат, площадь) |
| 14, 28 | Абонент (ФИО, номер телефона, время разговоров) |

5.3 Настройка столбцов DataGridView в режиме дизайнера

Добавить в приложение новую форму. Добавить на нее DataGridView. Настроить свойство Anchor (Top, Left, Right).

Используя добавление или правку столбцов, добавить в таблицу столбцы со следующими заголовками:

- Код (тип столбца: DataGridViewTextBoxColumn);
- Название (тип столбца: DataGridViewTextBoxColumn);
- Цена (тип столбца: DataGridViewTextBoxColumn);
- В корзину (тип столбца: DataGridViewCheckBoxColumn).

При загрузке формы реализовать программное добавление 10 строк в таблицу, используя метод Add:

```
сетка.Rows.Add(код, "название товара", цена, false);
```

Код – это число от 1 до 10 по порядку.

Сделать столбец с кодом невидимым для пользователя.

Настроить разный цвет фона для четных и нечетных строк таблицы DataGridView.

Настроить денежный (валютный) формат отображения столбца с ценой, используя настройку стиля ячейки столбца с ценой.

5.4 Анализ данных DataGridView (на примере оформления заказов)

Добавить на форму из п.5.3 метку «Итог» и кнопку «Заказать». После нажатия этой кнопки в метку должны выводиться названия и цена тех товаров, которые отмечены флажком «В корзину» (каждый товар должен быть выведе с новой строки), после списка товаров должна выводиться стоимость заказа.

Для проверки, поставлен ли флажок, использовать следующее условие:

```
сетка[номер столбца с флажком, номер строки].Value != null &&  
(bool)сетка[номер столбца с флажком, номер строки].Value
```

Первое условие проверяет, что ячейка заполнена, второе – что ее значение равно true.

Если ни один товар не выбран, вывести в окне сообщения предупреждение, что нужно отметить товары для заказа, в метку «Итог» должно быть записано исходное значение.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C# и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое DataGridView и для чего он используется?

8.2 Что такое BindingSource и для чего он используется?

8.3 Какие режимы работы с данными поддерживаются в элементе DataGridView?

8.4 Как связать DataGridView с данными типа список (List<T>)?

8.5 Какие типы столбцов поддерживаются в DataGridView?

8.6 Как получить доступ к значению определенной ячейки?

8.7 Какие свойства позволяют настроить стиль отображения строк и ячеек DataGridView?

Лабораторная работа №29

Создание проекта с использованием компонентов стандартных диалогов

1 Цель работы

1.1 Научиться применять компоненты стандартных диалоговых окон в приложении на C#.

2 Литература

2.1 Горелов, С. В. Современные технологии программирования: разработка Windows-приложений на языке C# : учебник : в 2 т. Т. I. / С. В. Горелов. – Москва: Прометей, 2019. – 378 с. – URL: <https://ibooks.ru/reading.php?productid=365806>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.28.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Открытие текстовых документов

5.1.1 Добавить на форму меню MenuStrip с пунктом «Файл» и подпунктом «Открыть» и поле ввода RichTextBox. Поле ввода должно заполнять всю форму.

5.1.2 При нажатии на «Открыть» предоставить пользователю возможность выбора текстового файла с использованием OpenFileDialog. Настроить у диалогового окна выбора файла фильтрацию на следующие типы файлов:

- Текстовые документы (*.txt);
- Текст в формате rtf (*.rtf);
- Все файлы (*.*)

5.1.3 Реализовать отображение в поле ввода RichTextBox содержимого выбранного в диалоговом окне файла.

5.2 Сохранение текстовых документов

5.2.1 Добавить в пункт меню «Файл» подпункт «Сохранить».

5.2.2 При нажатии на «Сохранить» предоставить пользователю возможность указания имени сохраняемого текстового файла с использованием SaveFileDialog. Настроить фильтрацию на следующие типы файлов:

- Текстовые документы (*.txt);
- Текст в формате rtf (*.rtf).

5.2.3 Реализовать сохранение содержимого поля ввода RichTextBox в файл, указанный в диалоговом окне сохранения файла.

5.3 Изменение цвета и шрифта выделенного текста

5.3.1 Добавить в меню пункт «Настройки» с подпунктами «Цвет» и «Шрифт».

5.3.2 При нажатии на «Цвет» предоставить пользователю возможность выбора настроек цвета с использованием ColorDialog. Реализовать изменение цвета выделенного текста на выбранный в диалоговом окне.

5.3.3 При нажатии на «Шрифт» предоставить пользователю возможность выбора настроек шрифта с использованием FontDialog. Реализовать изменение шрифта выделенного текста на выбранный в диалоговом окне

5.4 Изменение стандартного пути сохранения файлов

5.4.1 Добавить в пункт меню «Настройки» подпункт «Папка по умолчанию».

5.4.2 Используя окно выбора папки FolderBrowserDialog, реализовать смену папки, открываемой по умолчанию.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C# и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое ColorDialog и какое у него основное свойство?

8.2 Что такое FontDialog и какое у него основное свойство?

8.3 Что такое OpenFileDialog и какое у него основное свойство?

8.4 Что такое SaveFileDialog и какое у него основное свойство?

8.5 Как настроить фильтрацию на определенные типы файлов в OpenFileDialog и SaveFileDialog?

8.6 Что такое FolderBrowserDialog и какое у него основное свойство?

Лабораторная работа №30

Создание проекта с использованием кнопочных компонентов

1 Цель работы

1.1 Научиться применять и настраивать кнопки в приложении на C#.

2 Литература

2.1 Фленов, М. Е. Библия C# / М. Е. Фленов. – Санкт-Петербург : БХВ-Петербург, 2016. – Режим доступа: <https://ibooks.ru/bookshelf/353561/reading>. Загл. с экрана. – п.5.7.1, п.5.9.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Настройка меню и горячих клавиш для работы с документом

5.1.1 Добавить на форму главное меню MenuStrip с пунктом «Файл» и подпунктами «Создать», «Открыть», «Сохранить», «Заккрыть».

Реализовать закрытие формы при нажатии на «Заккрыть».

5.1.2 Добавить на форму поле ввода RichTextBox. Поле ввода должно заполнять всю форму (настроить с помощью свойства Anchor или Dock). При нажатии на «Создать» поле ввода RichTextBox должно очищаться от содержимого, а заголовок окна меняться на текст «Безымянный».

5.1.3 При нажатии на «Открыть» предоставить пользователю возможность выбора текстового файла. Настроить у диалогового окна выбора файла фильтрацию на следующие типы файлов: *.txt, *.rtf, *.*.

Реализовать отображение в поле ввода RichTextBox содержимого выбранного в диалоговом окне файла. Заголовок окна должен меняться на полное название файла.

5.1.4 При нажатии на «Сохранить» предоставить пользователю возможность указания имени сохраняемого текстового файла. Настроить фильтрацию на следующие типы файлов: *.txt, *.rtf.

Реализовать сохранение содержимого поля ввода RichTextBox в файл, указанный в диалоговом окне сохранения файла.

5.2 Использование флажков в пунктах меню

5.2.1 Добавить в главное меню формы из п.5.1 пункт «Вид» с подпунктом «Ночная тема». Настроить свойства подпункта, чтобы он работал как флажок.

5.2.2 Реализовать смену фона в поле ввода на темную (темно-синий или другой подобный), если отмечен флажок «Ночная тема», иначе фон поля ввода должен быть белым. Для темного фона цвет текста надо делать светлым и наоборот.

5.3 Связь кнопок с клавишами

5.3.1 Создать новую форму (переход на форму сделать с формы из п.5.1) и добавить на нее две кнопки: «Привет» и «Заккрыть». При нажатии на первую кнопку должно выводиться окно с сообщением «Здравствуйете», при нажатии на вторую форма должна закрываться.

5.3.2 В свойствах формы указать кнопку «Привет», которая будет срабатывать по нажатию на клавишу Enter.

5.3.3 В свойствах формы указать кнопку «Заккрыть», которая будет срабатывать по нажатию на клавишу Esc.

5.4 Настройка внешнего вида кнопок

Изменить стиль кнопок из п.5.3, изменив его на плоский с пиктограммой и надписью.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C# и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое Button и для чего она используется?

8.2 Какое основное событие происходит при нажатии на кнопку?

8.3 Для чего используется свойство AcceptButton?

8.4 Для чего используется свойство CancelButton?

8.5 Какое минимальное количество раз выполняется цикл с постусловием?

Лабораторная работа №31

Создание проекта с использованием меню и панели инструментов

1 Цель работы

1.1 Научиться применять и настраивать различные виды меню, панель инструментов, строку состояния в приложении на C#.

2 Литература

2.1 Фленов, М. Е. Библия C# / М. Е. Фленов. – Санкт-Петербург : БХВ-Петербург, 2016. – Режим доступа: <https://ibooks.ru/bookshelf/353561/reading>. Загл. с экрана. – п.5.9.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Настройка меню и горячих клавиш для работы с документом

5.1.1 Добавить на форму главное меню MenuStrip с пунктом «Файл» и подпунктами:

- «Создать» (горячие клавиши: Ctrl+N),
- «Открыть» (горячие клавиши: Ctrl+O),
- «Сохранить» (горячие клавиши: Ctrl+S),
- разделитель,
- «Закрыть».

Реализовать закрытие формы при нажатии на «Закрыть».

5.1.2 Добавить на форму поле ввода RichTextBox. Поле ввода должно заполнять всю форму. При нажатии на «Создать» поле ввода RichTextBox должно очищаться от содержимого, а заголовок окна меняться на текст «Безымянный».

5.1.3 При нажатии на «Открыть» предоставить пользователю возможность выбора текстового файла. Настроить у диалогового окна выбора файла фильтрацию на следующие типы файлов: *.txt, *.rtf, *.*.

Реализовать отображение в поле ввода RichTextBox содержимого выбранного в диалоговом окне файла. Заголовок окна должен меняться на полное название файла.

5.1.4 При нажатии на «Сохранить» предоставить пользователю возможность указания имени сохраняемого текстового файла. Настроить фильтрацию на следующие типы файлов: *.txt, *.rtf.

Реализовать сохранение содержимого поля ввода RichTextBox в файл, указанный в диалоговом окне сохранения файла.

5.2 Настройка контекстного меню для настройки цвета и шрифта текста

5.2.1 Создать контекстное меню и добавить в него пункты «Цвет» и «Шрифт».

5.2.2 При нажатии на «Цвет» предоставить пользователю возможность выбора настроек цвета. Реализовать изменение цвета выделенного текста на выбранный в диалоговом окне.

5.2.3 При нажатии на «Шрифт» предоставить пользователю возможность выбора настроек шрифта. Реализовать изменение шрифта выделенного текста на выбранный в диалоговом окне.

5.2.4 Привязать контекстное меню к RichTextBox.

5.2.5 Добавить в главное меню пункт «Форматирование» и в свойстве DropDown выбрать созданное контекстное меню.

5.3 Настройка панели инструментов

5.3.1 Добавить на форму панель инструментов ToolStrip для настройки текста.

5.3.2 Добавить в панель инструментов выпадающий список для отображения списка доступных шрифтов и изменения шрифта в поле ввода.

5.3.3 Добавить в панель инструментов три кнопки с подписями «По левому краю», «По центру», «По правому краю» (будут использоваться для настройки выравнивания текста). По умолчанию должен быть установлен флажок у кнопки «По левому краю».

5.4 Настройка отображения информации в строке состояния

5.4.1 Добавить на форму строку состояния.

5.4.2 Реализовать отображение в строке состояния текущей даты.

5.4.3 Реализовать отображение в строке состояния количества символов в поле ввода и количества символов в поле ввода без учета пробелов.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C# и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое MenuStrip и для чего он используется?

8.2 Что такое ContextMenuStrip и для чего он используется?

8.3 Что такое ToolStrip и для чего он используется?

8.4 Что такое StatusStrip и для чего он используется?

Лабораторная работа №32

Обработка исключений

1 Цель работы

1.1 Научиться обрабатывать исключительные ситуации в приложениях.

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд. / М. Е. Фленов. – Санкт-Петербург : БХВ-Петербург, 2019. – URL: <https://ibooks.ru/bookshelf/366634/reading>. – Режим доступа: только для зарегистрированных пользователей. – Текст : электронный. – гл.9.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Создать консольное приложение на языке C#, в котором пользователь должен ввести 2 целых числа (делимое и делитель), после чего результат деления должен быть выведен на экран. Обработать исключение деления на 0.

5.2 Добавить в приложение из п.5.1 обработку некорректного ввода данных.

5.3 Изменить в приложении из п.5.1 тип данных переменных с целых на вещественный. Для корректности вычислений при вводе делителя 0 генерировать исключение `DivideByZeroException`.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C# и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «исключение»?

8.2 Для чего предназначен блок `try`?

8.3 Для чего предназначен блок `catch`?

8.4 Какой оператор позволяет программно сгенерировать исключение?

Лабораторная работа №33

Обработка файлов в оконном приложении

1 Цель работы

1.1 Научиться применять классы для работы с файловой системой в приложениях на C#.

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд. / М. Е. Фленов. – Санкт-Петербург : БХВ-Петербург, 2019. – URL: <https://ibooks.ru/bookshelf/366634/reading>. – Режим доступа: только для зарегистрированных пользователей. – Текст : электронный. – п.14.2-14.4.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Запись заметок в существующий текстовый файл (дозапись данных в текстовый файл)

5.1.1 Создать новую форму и реализовать переход к ней с главной формы.

5.1.2 Добавить на форму заметок многострочное поле ввода (TextBox с параметром Multiline: true) и кнопку «Записать». При нажатии на кнопку требуется записывать в конец файла Notes.txt (считать, что хранится в одной папке с файлом приложения) следующие данные:

- на отдельной строке текущую дату и время,
- на отдельной строке содержимое поля ввода,
- пустую строку (чтобы заметки не сливались в сплошной текст).

5.1.3 Перед записью данных в файл реализовать проверку: если файл Notes.txt существует, то предупредить пользователя, что создается новый файл с заметками.

5.2 Проверка существования пользователя в списке пользователей (чтение данных из текстового файла)

5.2.1 Создать новую форму и реализовать переход к ней с главной формы.

5.2.2 Добавить на форму авторизации поле ввода логина и кнопку «Авторизоваться». При нажатии на кнопку следует проверять, что введенный логин хранится в файле logins.txt. Если имя найдено, приветствовать пользователя и закрывать форму, иначе сообщить о некорректном логине.

Для реализации следует считать в цикле набор данных о каждом пользователе из файла и сравнить введенный логин с логином каждого из пользователей, информация о котором хранится в файле logins.txt.

5.3 Сохранение данных о покупках в текстовый документ формата csv (запись данных в текстовый файл)

5.3.1 Создать новую форму и реализовать переход к ней с главной формы.

5.3.2 Добавить на форму покупок DataGridView со столбцами Товар, Цена, Количество и кнопку «Сохранить». При нажатии на кнопку должно открываться окно указания имени сохраняемого файла (включить фильтр на форматы txt и csv).

5.3.3 После указания имени файла в него должны записываться данные из DataGridView в формате csv (первой строкой должна идти строка заголовков столбцов, остальными строками – значения каждой строки таблицы).

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C# и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какое пространство имен содержит себе классы для работы с файловой системой и файловыми потоками?

8.2 Для чего предназначены классы StreamReader и StreamWriter?

8.3 Для чего предназначены классы BinaryReader и BinaryWriter?

8.4 Какие классы предоставляют информацию о дисках и каталогах?

8.5 Какие классы предоставляют информацию о файлах?

Лабораторная работа №34

Разработка интерфейса приложения

1 Цель работы

1.1 Научиться программно создавать и настраивать элементы управления в приложениях на C#;

1.2 Научиться использовать таймер в приложениях на C#.

2 Литература

2.1 Фленов М.Е. Библия C#. – Санкт-Петербург : БХВ-Петербург, 2016. – Режим доступа: <https://ibooks.ru/reading.php?productid=353561>, только для зарегистрированных пользователей. – Загл. с экрана. – гл.5, гл.13.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Отображение времени на форме

5.1.1 Создать новую форму и реализовать переход к ней с главной формы.

5.1.2 Добавить на форму из панели элементов Timer, две кнопки с подписями «Старт» и «Пауза» соответственно и метку для вывода текущего времени.

5.1.3 В обработчиках нажатий на кнопки «Старт» и «Пауза» вызвать методы запуска и остановки таймера соответственно.

5.1.4 В обработчике события Tick реализовать вывод времени в формате мм:сс (минуты:секунды).

5.2 Анимация движения персонажа

5.2.1 Создать новую форму и реализовать переход к ней с главной формы.

5.2.2 Добавить на форму из панели элементов Timer, ImageList и PictureBox.

Изменить фон формы.

Добавить у ImageList в коллекцию Images набор изображений:

- 1 рисунок: герой стоит

- 2 рисунок: герой сделал шаг одной ногой

- 3 рисунок: герой сделал шаг другой ногой

5.2.3 Реализовать движение героя при запуске формы, используя смену изображений в PictureBox по таймеру.

5.3 Программная генерация клеточного игрового поля на основе данных из файла

5.3.1 Создать новую форму и реализовать переход к ней с главной формы.

5.3.2 Предоставить пользователю возможность выбора текстового файла карты с помощью диалогового окна при нажатии на кнопку на форме или на пункт меню.

5.3.3 После выбора файла требуется найти ширину (количество описанных клеток в одной строке файла) и высоту (количество строк в файле) игрового поля и реализовать формирование игрового поля путем программного создания и отображения его клеток.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C# и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какие основные свойства и события используются у таймера в приложении C#?

8.2 Как запустить и остановить таймер в приложении C#?

8.3 Что такое ImageList и для чего он используется в приложении C#?

8.4 Какие режимы изменения размеров изображения (SizeMode) поддерживаются у PictureBox и как изменяет изображение каждый из режимов?

8.5 Как реализовать скрытие заголовков вкладок у TabControl в приложении C#?

8.6 Как программно создать и добавить элементы на форму в приложении C#?

Лабораторная работа №35

Разработка многооконного приложения

1 Цель работы

1.1 Научиться создавать различные типы многооконных приложений на C#.

2 Литература

2.1 Фленов М.Е. Библия C#. – Санкт-Петербург : БХВ-Петербург, 2016. – Режим доступа: <https://ibooks.ru/reading.php?productid=353561>, только для зарегистрированных пользователей. – Загл. с экрана. – гл.5, гл.13.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Создание оконного приложения из набора MDI-форм

Создать оконное приложение Windows Forms. Добавить в проект 3 формы.

5.2 Реализация открытия форм

Добавить на стартовую форму кнопки «Открыть форму» и «Открыть форму в диалоговом режиме». Реализовать открытие одной из созданных форм в диалоговом и недиалоговом режимах при нажатии на соответствующие кнопки.

5.3 Создание MDI-приложения

Добавить на стартовую форму кнопку «Открыть mdi-форму»

Реализовать открытие одной из созданных форм, сделав ее MDI-контейнером. В MDI-контейнере добавить пункты меню, позволяющие открыть 2 из созданных форм.

5.4 Реализация закрытия всех форм MDI-приложения

Реализовать закрытие всех открытых в MDI-контейнере форм.

5.5 Активация открытых форм MDI-приложения

Реализовать активацию в MDI-контейнере требуемой формы вместо ее повторного открытия

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C# и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какие виды оконных приложений существуют?

8.2 Что такое MDI-приложение?

8.3 В чем отличие при открытии формы с использованием Show() и ShowDialog()?

8.4 Что такое MessageBox?

Лабораторная работа №36

Разработка функциональной схемы работы приложения

1 Цель работы

1.1 Научиться разрабатывать функциональную схему работы приложения.

2 Литература

2.1 Фленов М.Е. Библия С#. – Санкт-Петербург : БХВ-Петербург, 2016. – Режим доступа: <https://ibooks.ru/reading.php?productid=353561>, только для зарегистрированных пользователей. – Загл. с экрана. – гл.5.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Отобразить в функциональной схеме следующее:

При загрузке формы на ней размещаются элементы игры: мяч, прямоугольник-ракетка, кирпичи. Кирпичи в каждом ряду должны быть разного цвета, всего 4 ряда по 10 кирпичей в каждом, ряд занимает всю строку от одной до другой стены, между каждым кирпичом должно быть расстояние в несколько пикселей;

5.2 Отобразить в функциональной схеме следующее:

Игрок управляет ракеткой, передвигая ее с помощью мыши горизонтально от одной стенки до другой и подставляя её под мяч для предотвращения его падения,

5.3 Отобразить в функциональной схеме следующее:

Мяч движется по таймеру и отталкивается от ракетки и левой, правой и верхней стен игрового поля, меняя при этом вектор своего движения. При попадании за нижнюю границу игрок теряет мяч и получает новый,

5.4 Отобразить в функциональной схеме следующее:

Удар мяча по кирпичу (столкновение элементов) приводит к разрушению кирпича. После того, как все кирпичи уничтожены, появляется сообщение о победе,

5.5 Отобразить в функциональной схеме следующее:

При нажатии на клавишу пробел игра запускается и ставится на паузу (таймер останавливается и запускается).

6 Порядок выполнения работы

6.1 Используя текстовый или табличный редактор, выполнить задания из п.5, описав функциональную схему работы Арканоида, заполнив таблицу со столбцами Элемент управления, Событие, Действие.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Как определяется функциональная схема приложения?
- 8.2 На какие виды делятся компоненты форм?
- 8.3 Что требуется создать, чтобы реагировать на события форм

Лабораторная работа №37

Разработка игрового приложения

1 Цель работы

1.1 Научиться разрабатывать интерактивные игровые приложения на C#.

2 Литература

2.1 Фленов М.Е. Библия C# / М.Е. Фленов. – Санкт-Петербург : БХВ-Петербург, 2016. – Режим доступа: <https://ibooks.ru/reading.php?productid=353561>, только для зарегистрированных пользователей. – Загл. с экрана. – гл.5, гл.13.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Настройка интерфейса приложения

5.1.1 Настроить форму:

- изменить цвет фона, текста, шрифт
- форма должна открываться по центру экрана и быть фиксированного размера
- заголовок формы: Арканойд

5.1.2 В режиме дизайнера добавить на форму:

- метку для отображения счета
- метку для отображения количества потерянных мячей
- панель-ракетку (Name: racket)
- панель-мяч (Name: ball)

Настроить внешний вид добавленных элементов.

5.1.3 На загрузку формы прописать код, программно размещающий на форме панели-кирпичи (4 ряда по 10 штук в каждом, у каждого ряда – свой цвет, ряд занимает всю строку от одной до другой стены, между каждым кирпичом должно быть расстояние в несколько пикселей).

5.1.4 На загрузку формы прописать следующий код для изменения контура панели-мяча на круглую (высота и ширина панели должны быть одинаковыми):

```
System.Drawing.Drawing2D.GraphicsPath path =  
    new System.Drawing.Drawing2D.GraphicsPath();  
path.AddEllipse(ball.ClientRectangle);  
ball.Region = new Region(path);
```

5.1.5 На загрузку формы прописать код, изменяющий начальное положение мяч (мяч должен появляться в случайном месте в нижней части игрового поля).

5.2 Реализация движения ракетки и мяча

5.2.1 Для того, чтобы ракетка двигалась за курсором мыши в пределах формы, создать у формы обработчик движения мыши и прописать в нем следующий код:


```

// определение центра ракетки, по нему будет находиться курсор мыши
int racketCenterX = racket.Width / 2;
// this.ClientSize - это прямоугольная клиентская область формы
// this.ClientSize.Width - это доступная (клиентская) ширина формы
if (e.X > racketCenterX && e.X < this.ClientSize.Width - racketCenterX)
{
    // изменение местоположения ракетки
    racket.Location = new Point(e.X - racketCenterX, racket.Top);
}

```

5.2.2 Для того, чтобы мяч двигался по таймеру, добавить на форму таймер и реализовать его запуск при загрузке формы. Создать обработчик срабатывания таймера, в котором изменять местоположение мяча. При старте игры мяч должен лететь вверх и влево или вправо. Для выполнения этого добавить в форму два целочисленных поля:

- ballSpeedX – скорость по координате X
- ballSpeedY – скорость по координате Y

Начальные значения полей генерировать при загрузке формы (не должны быть равны нулю).

5.3 Реализация запуска и остановки игры при нажатии пробела

5.3.1 Установить у формы свойство KeyPreview = true, чтобы форма перехватывала нажатия клавиш.

5.3.2 Добавить на форму обработчик нажатия клавиши KeyDown. В обработчике написать код, проверяющий, что нажата клавиша пробел (e.KeyCode == Keys.Space).

Если нажат пробел и игра запущена, то таймер должен останавливаться, на игровом поле должна появляться надпись «Пауза».

Если нажат пробел и игра остановлена, то таймер должен запускаться, на игровом поле не должно быть надписи «Пауза».

5.4 Реализация отскокиваний мяча и появления нового мяча

5.4.1 Для того, чтобы мяч отскакивал от стен, в обработчике срабатывания таймера учесть следующее:

- при столкновении левого края мяча с левым краем формы или правого края мяча с правым краем формы скорость движения мяча по координате X должна меняться на противоположную по знаку,
- при столкновении верхнего края мяча с верхним краем формы скорость движения мяча по координате Y должна меняться на противоположную по знаку,

5.4.2 Для того, чтобы мяч отскакивал от ракетки, в обработчике срабатывания таймера учесть следующее:

- при столкновении нижнего края мяча с верхним краем ракетки (если правый или левый край мяча касается ракетки) скорость движения мяча по координате Y должна меняться на противоположную по знаку.

5.4.3 Для появления мяча после потери в новой точке в обработчике срабатывания таймера при уходе мяча за нижний край формы следует:

- останавливать таймер,

- увеличивать счетчик потерянных мячей, видимый пользователю,
- выводить окно с сообщением, что мяч потерян,
- изменять местоположение мяча аналогично тому, как это сделано при загрузке формы,
- запускать таймер.

5.5 Реализация получения очков за выбитые кирпичи

При реализации потребуются следующие свойства и методы проверки пересечения прямоугольных областей, чтобы не проверять отдельно столкновения мяча с краями каждого кирпича:

- объект **Bounds** – объект типа **Rectangle** (прямоугольник), возвращающий местоположение и размер объекта.
- прямоугольник1.**IntersectsWith**(прямоугольник2) – метод возвращает истину, если прямоугольники пересеклись.
- прямоугольник1.**Intersect**(прямоугольник2); – метод записывает в первый прямоугольник прямоугольник, находящийся на пересечении двух указанных прямоугольников.

5.5.1 Для определения столкновения панели-мяча с панелями-кирпичами и удаления тех, с которыми было столкновение, в обработчике срабатывания таймера после изменения местоположения мяча перебрать все элементы управления из коллекции `this.Controls`, используя следующий код:

```
// прямоугольная область мяча
Rectangle ballRectangle = ball.Bounds;
// проход по всем элементам управления формы
for (int i = this.Controls.Count - 1; i >= 0; i--)
{
    // item - текущий элемент управления
    Control item = this.Controls[i];
    // проверка, что item является панелью-кирпичом
    // (объект типа Panel, не мяч и не ракетка)
    if (item is Panel && item != ball && item != racket)
    {
        //проверка пересечения прямоугольных областей
        if (ballRectangle.IntersectsWith(item.Bounds))
        {
            //код, который должен быть выполнен при столкновении
        }
    }
}
```

В случае пересечения следует удалять кирпич, с которым было пересечение, из коллекции `this.Controls`. При удалении последнего сообщить о победе.

5.5.2 При столкновении начислять определенное количество баллов (чем выше ряд, тем больше баллов) и выводить результат в метку Счет.

5.5.3 При столкновении для изменения вектора движения мяча следует до удаления кирпича из коллекции получить прямоугольник, находящийся на пересечении мяча и кирпича:

```
ballRectangle.Intersect(item.Bounds);
```

Если у прямоугольника `ballRectangle` высота меньше ширины, то было столкновение с нижним или верхним краем мяча, скорость движения мяча по координате `Y` должна меняться на противоположную по знаку. Иначе было столкновение с левым или правым краем мяча, скорость движения мяча по координате `X` должна меняться на противоположную по знаку.

6 Порядок выполнения работы

6.1 Используя Microsoft Visual Studio, создать проект C# и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какие обработчики событий применяются для обработки событий клавиатуры?

8.2 Какие обработчики событий применяются для обработки событий мыши?

8.3 Для чего применяются методы `IntersectsWith()` и `Intersect()`?

8.4 Какие методы используются для программного добавления и удаления элементов управления из коллекции элементов формы?

Лабораторная работа №38

Разработка приложений

1 Цель работы

- 1.1 Научиться разрабатывать приложения на C#.

2 Литература

2.1 Фленов М.Е. Библия C# / М.Е. Фленов. – Санкт-Петербург : БХВ-Пете.рбург, 2016. – Режим доступа: <https://ibooks.ru/reading.php?productid=353561>, только для зарегистрированных пользователей. – Загл. с экрана. – гл.5, гл.13.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание лабораторной работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Спроектировать и разработать интерфейс оконного приложения
- 5.2 Реализовать функциональность приложения
- 5.3 Выполнить тестирование разработанного приложения

6 Порядок выполнения работы

- 6.1 Используя Microsoft Visual Studio, создать проект C# и выполнить задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Из каких этапов состоит разработка приложений?
- 8.2 Какие операторы языка C# позволяют обработать исключения в программе?
- 8.3 Какие элементы управления использованы в разработанном приложении?
- 8.4 Для чего выполняется тестирование приложений?