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

МДК.01.04 СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

Методические указания по выполнению лабораторных работ

по специальности:
09.02.07 – Информационные системы и программирование

Архангельск 2022

Маломан, Ю.С. МДК.01.04 Системное программирование. Методические указания по выполнению лабораторных работ. – Архангельск: АКТ (ф) СПбГУТ, 2022.

Методические указания по выполнению лабораторных работ содержат описания работ, предусмотренных рабочей программой по МДК.01.04 «Системное программирование». Каждая работа рассчитана на 2 часа, общий объём составляет 60 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 Архангельского колледжа телекоммуникаций (филиал) СПбГУТ.

Содержание

Лабораторная работа №1
Изучение процесса разработки линейных алгоритмов на языке ассемблера

1 Цель работы

1.1 Изучить процесс разработки линейных алгоритмов на языке ассемблера;

1.2 Научиться выполнять вычисление математических выражений на языке ассемблера;

1.3 Закрепить навык отладки приложений в MS Visual Studio.

2 Литература

2.1 Куляс, О. Л. Курс программирования на ASSEMBLER / О. Л. Куляс. – Москва : СОЛОН-ПРЕСС, 2017. – 220 с. – URL: https://ibooks.ru/reading.php?productid=361979. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.6-7.

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 Реализовать вычисление значения функции $y(x)=ax+b$ в ассемблерной вставке.

5.2 Реализовать вычисление значения функции $y(x)=ax^2+bx+c$ в ассемблерной вставке.

5.3 Реализовать вычисление значения частного и остатка от деления в ассемблерной вставке.

5.4 Реализовать обмен значений переменных a и b в ассемблерной вставке.

5.5 Реализовать заполнения каждого байта 32-битного регистра значением из 8-битного регистра.

6 Порядок выполнения работы

6.1 Запустить Microsoft Visual Studio, создать консольный проект согласно заданию, показать изменения значений регистров в режиме отладки.

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 применяется команда mov?

8.2 Какие арифметические команды применяются в языке ассемблера и какое у них назначение?

8.3 Какие побитовые команды применяются для ускорения умножения и деления и когда они применимы?

8.4 Каков размер в байтах регистров общего назначения EAX, AX, AH, AL?

8.5 Какие способы обнуления регистров могут применяться в языке ассемблера?

8.6 Можно ли записать значение переменной типа int в регистр AX и почему?

Лабораторная работа №2
Изучение процесса разработки разветвляющихся алгоритмов на языке ассемблера

1 Цель работы

1.1 Изучить процесс разработки разветвляющихся алгоритмов на языке ассемблера;

1.2 Закрепить навык отладки приложений в MS Visual Studio.

2 Литература

2.1 Куляс, О. Л. Курс программирования на ASSEMBLER / О. Л. Куляс. – Москва : СОЛОН-ПРЕСС, 2017. – 220 с. – URL: https://ibooks.ru/reading.php?productid=361979. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.7-8.

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, находящую минимум из двух чисел.

5.2 Написать программу, вычисляющую значение функции y:

– если $x < -10$, то $y(x) = a*x^2$;

– если $-10 \leq x < 10$, то $y(x) = a*|x|$;

– если $x \geq 10$, то $y(x) = a-x$.

5.3 Написать программу, находящую максимум из трех чисел.

5.4 Написать программу, определяющую по введенному пользователем номеру месяца, сколько дней в месяце (считать, что в феврале 28 дней).

5.5 Написать программу, запрашивающую сумму покупки и внесенную покупателем сумму. Если сдача не требуется, выводить на экран «Спасибо!»; если денег внесено больше, чем необходимо — «Возьмите сдачу» и указывать сумму сдачи; если денег недостаточно — сообщение об этом и указать размер недостающей суммы.

6 Порядок выполнения работы

6.1 Запустить Microsoft Visual Studio, создать консольный проект согласно заданию. Входные данные – целые числа. Проверку условия выполнять, используя ассемблерную вставку.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

Лабораторная работа №3
Изучение процесса разработки циклов со счетчиком на языке ассемблера

1 Цель работы

1.1 Изучить процесс разработки циклов со счетчиком на языке ассемблера;

1.2 Закрепить навык отладки приложений в MS Visual Studio.

2 Литература

2.1 Куляс, О. Л. Курс программирования на ASSEMBLER / О. Л. Куляс. – Москва : СОЛОН-ПРЕСС, 2017. – 220 с. – URL: https://ibooks.ru/reading.php?productid=361979. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.7-8.

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, вычисляющую и выводящую на экран значение факториала числа N. В случае некорректных данных (N<0) требуется вывести на экран сообщение, что факториал не существует.

5.2 Написать программу, выводящую на экран количество делителей положительного числа N, включая 1 и само число N.

5.3 Написать программу, выводящую на экран последовательность чисел, кратных 5, в диапазоне от N до 0 (от наибольшего к наименьшему, например: 20, 15, 10, 5, 0).

5.4 Написать программу, выводящую на экран таблицу умножения (в первой строке и первом столбце – множители, в ячейках – произведение).

6 Порядок выполнения работы

6.1 Запустить Microsoft Visual Studio, создать консольный проект согласно заданию. Входные данные – целые числа. Цикл реализовать, используя ассемблерную вставку.

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 команды asm требуются для организации цикла со счетчиком?

8.2 Как на языке ассемблера реализовать досрочный выход из цикла?

8.3 Какие команды asm требуются, чтобы увеличить значение счетчик на 1 и на большее значение (для увеличения значения счетчика)?

8.4 Какие команды asm требуются, чтобы уменьшить значение счетчик на 1 и на большее значение (для уменьшения значения счетчика)?

Лабораторная работа №4
Изучение процесса разработки циклов с условием на языке ассемблера

1 Цель работы

1.1 Изучить процесс разработки циклов с предусловием и постусловием на языке ассемблера;

1.2 Закрепить навык отладки приложений в MS Visual Studio.

2 Литература

2.1 Куляс, О. Л. Курс программирования на ASSEMBLER / О. Л. Куляс. – Москва : СОЛОН-ПРЕСС, 2017. – 220 с. – URL: https://ibooks.ru/reading.php?productid=361979. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.7-8.

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 с использованием цикла с предусловием, 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.

5.2 Написать программу с использованием цикла с постусловием, в которой пользователь пытается угадать число.

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.

5.3 Написать программу с использованием цикла с предусловием, 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.

6 Порядок выполнения работы

6.1 Запустить Microsoft Visual Studio, создать консольный проект согласно заданию. Входные данные – целые числа. Цикл реализовать, используя ассемблерную вставку.

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 минимальное количество раз выполняется цикл с предусловием?
8.2 Какое минимальное количество раз выполняется цикл с постусловием?
8.3 Какова общая форма цикла с предусловием на языке ассемблера?
8.4 Какова общая форма цикла с постусловием на языке ассемблера?

Лабораторная работа №5
Изучение принципов работы математического сопроцессора

1 Цель работы

1.1 Изучить принципы работы сопроцессора и методы его программирования средствами ассемблера.

1.2 Закрепить навык отладки приложений в MS Visual Studio.

2 Литература

2.1 Куляс, О. Л. Курс программирования на ASSEMBLER / О. Л. Куляс. – Москва : СОЛОН-ПРЕСС, 2017. – 220 с. – URL: https://ibooks.ru/reading.php?productid=361979. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.17.

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 находящую на основе введенных целочисленных данных:

- 1 вариант: площадь кольца ($S=\pi*(R^2-r^2)$).
- 2 вариант: радиус сферы ($R=0.5*\sqrt{(S/\pi)}$).
- 3 вариант: площадь сферы ($S=4*\pi*R^2$).

Результат — вещественный.

5(2 Написать программу, вычисляющую значение функции $f(x)$ на языке ассемблера при вещественном значении x (вводится пользователем). Функции показаны на рисунке 1 (вариант = № ПК).

1. $f(x) = x^2 + 2x + 3$	**8.** $f(x) = (x^2 - 2x)^3$
2. $f(x) = (x^3 + 1) + 12x$	**9.** $f(x) = x^3 - 2x$
3. $f(x) = x^3 - 3x^2$	**10.** $f(x) = x^2 + 3x + 3$
4. $f(x) = x^3 + (x + 1)^2$	**11.** $f(x) = x^3 - 2x + 3$
5. $f(x) = (x^2 + 1)^2 - 1$	**12.** $f(x) = (x^3 + 5) + 2x$
6. $f(x) = x^2 + 3x - 4$	**13.** $f(x) = (x + 1)^2 - 1$
7. $f(x) = x^2 + 4x - 10$	**14.** $f(x) = (x^2 + x)^3$

Рисунок 1 — Функции для заданий 2-3

5(3 Написать программу, вычисляющую значения функции $f(x)$ на языке ассемблера при вещественных значениях x из промежутка [x1, x2] с шагом 1. Значения x1 и x2 вводятся пользователем. Функции показаны на рисунке 1 (вариант = № ПК).

6 Порядок выполнения работы

6.1 Запустить Microsoft Visual Studio, создать консольный проект согласно заданию. Входные данные – целые числа.

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 Какая команда используется для инициализации сопроцессора?

8.2 Какие команды сопроцессора используются для загрузки констант?

8.3 Что такое ST(0)?

8.4 Какие арифметические команды имеются в сопроцессоре?

8.5 Какие команды сопроцессора используются для передачи данных?

Лабораторная работа №6
Изучение принципов работы цепочечных команд

1 Цель работы

1.1 Изучить принципы обработки цепочек данных, используя asm.

1.2 Закрепить навык отладки приложений в MS Visual Studio.

2 Литература

2.1 Куляс, О. Л. Курс программирования на ASSEMBLER / О. Л. Куляс. – Москва : СОЛОН-ПРЕСС, 2017. – 220 с. – URL: https://ibooks.ru/reading.php?productid=361979. – 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, копирующую содержимое одной строки в другую строку (длина строки-приемника не м.б. меньше строки-источника).

5.2 Написать программу, копирующую содержимое одного массива в другой массив данных типа int (int – 32 бита, т.е. двойное слово).

5.3 Написать программу, копирующую подстроку длиной n с i-ой позиции из исходной строки s1 в строку s2 (i начинается с нуля).

5.4 Написать программу, возвращающую:

- 1 вариант: позицию, на которой в строке находится введенный символ;
- 2 вариант: количество совпадений значений элементов целочисленного массива со введенным пользователем значением.

6 Порядок выполнения работы

6.1 Запустить Microsoft Visual Studio, создать консольный проект согласно заданию 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 в операциях с цепочками используется флаг DF?

8.2 Для чего используются префиксы в операциях с цепочками данных?

8.3 Какие команды используются для сравнения цепочек данных?

8.4 Операнды каких размеров могут использоваться в операциях с цепочками данных?

8.5 Какие регистры предназначены для задания источника и приемника в операциях с цепочками данных?

Лабораторная работа №7
Изучение процесса разработки модулей на языке ассемблера

1 Цель работы

1.1 Изучить процесс разработки модулей с использованием ассемблера MASM.

2 Литература

2.1 Куляс, О. Л. Курс программирования на ASSEMBLER / О. Л. Куляс. – Москва : СОЛОН-ПРЕСС, 2017. – 220 с. – URL: https://ibooks.ru/reading.php?productid=361979. – 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 Разработать модуль на языке ассемблера, содержащий процедуру, вычисляющую сумму двух чисел a и b. Для передачи данных использовать стек.

5.2 Разработать модуль на языке ассемблера, содержащий процедуру, вычисляющую значение 2^x (x – неотрицательное).

Результат нужно возвращать в вызывающий модуль.

5.3 Разработать модуль на языке ассемблера, содержащий процедуру, сравнивающую две строки. Для передачи строк использовать общую область памяти.

Результат сравнения (строки равны или строки не равны) нужно отобразить, используя MessageBox.

6 Порядок выполнения работы

6.1 Запустить Microsoft Visual Studio, создать консольный проект согласно заданию 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 процесс разработки программ на языке ассемблера?

8.2 Из каких сегментов состоит программа на языке ассемблера и как объявляется каждый из сегментов?

8.3 Как в программе на языке ассемблера объявить целое число, строку, массив?

8.4 Для чего предназначен сегмент кода программы на языке ассемблера?

8.5 Что такое MASM?

8.6 Каков синтаксис вызова процедур в программах на языке ассемблера?

8.7 Какие способы передачи параметров между процедурами используются в программах на языке ассемблера?

Лабораторная работа №8
Дизассемблирование приложений

1 Цель работы

1.1 Научиться применять дизассемблеры для изучения и модификации ПО.

2 Литература

2.1 Владстон, Ф. Ф. Теоретический минимум по Computer Science. Все, что нужно программисту и разработчику / Ф. Ф. Владстон. – Санкт-Петербург : Питер, 2018. – 224 с. – URL: https://ibooks.ru/bookshelf/358138/reading. – Режим доступа: только для зарегистрированных пользователей. – Текст: электронный. – п.7.2.

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 Изучение дизассемблированного кода в Visual Studio

Создать в Visual Studio два консольных приложения на языке C++, вычисляющих в цикле со счетчиком сумму чисел от 1 до n (вводится пользователем). В первом приложении переменная-счетчик должна быть объявлена в разделе for, во втором — перед циклом.

В режиме отладки включить окно «Дизассемблированный код» (меню Отладка — Окна — Дизассемблированный код), в контекстном меню которого отметить галочками все пункты «Показать ...» кроме «Показать байты кода».

Сравнить дизассемблированный код разработанных приложений.

5.2 Применение дизассемблера onlinedisassembler.com

Приложение запрашивает у пользователя пароль для того, чтобы можно было работать с программой в определенном режиме. В случае, если пароль корректен, программа приветствует пользователя, иначе — сообщает о некорректных данных.

Требуется подобрать пароль для приложения, проанализировав его код с помощью онлайн-дизассемблера onlinedisassembler.com.

При загрузке файла оставить настройки по умолчанию. Для решения задачи нужно изучить содержимое разделов .data и .rdata. Строки оканчиваются байтом 00. Для ускорения поиска можно использовать фильтрацию из боковой панели (нажать кнопку A, в поле Filter ввести текст известных строковых данных и нажать Enter).

Во вкладке HEX отображаются данные в строковом виде с указанием байтов.

Во вкладке Disassembly каждый байт отображается построчно.

5.3 Анализ дизассемблированного кода

Проанализировать код дизассемблированного приложения.

Восстановить алгоритм работы программы и указать, что нужно изменить в ассемблерном коде, чтобы при вводе любого пароля пользователю выводилось приветствие.

6 Порядок выполнения работы

6.1 Используя онлайн-дизассемблер ODA (onlinedisassembler.com) и встроенный дизассемблер Visual Studio, выполнить анализ программного обеспечения согласно описанию в 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 «дизассемблирование»?
8.2 Для чего выполняется дизассемблирование программного кода?
8.3 Что такое «дизассемблер»?
8.4 Какие существуют программы-дизассемблеры?
8.5 Как открыть окно дизассемблированного программного кода в Visual Studio?
8.6 Для чего выполняется обфускация кода?

Лабораторная работа №9
Создание проекта в эмуляторе Arduino

1 Цель работы

1.1 Научиться создавать имитационные модели проектов, работающих под управлением микроконтроллеров для Arduino;

1.2 Получить навыки работы с платой Arduino Uno и макетной платой.

2 Литература

2.1 Благодаров, А. В. Программирование микроконтроллеров семейства 1986ВЕ9х компании Миландр / А. В. Благодаров. – Москва: Горячая Линия–Телеком, 2020. – 232 с. – URL: https://ibooks.ru/bookshelf/372218/reading. – Режим доступа: только для зарегистрированных пользователей. – Текст: электронный. – гл.1-2.

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 простейшую Arduino-программу, в которой одноцветный светодиод должен гореть 2 секунды с паузой 0,5 секунды.

5.2 Создать дубликат модели из п.5.1. Переименовать модель и добавить в нее светодиод, подключив его параллельно существующему.

5.3 Создать дубликат модели из п.5.1. Переименовать модель и добавить в нее светодиод, подключив его последовательно с существующим.

6 Порядок выполнения работы

6.1 Перейти на сайт https://www.tinkercad.com/, зарегистрироваться и авторизоваться.

6.2 Выполнить задания из п.5.

6.3 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 и для чего используется «Arduino Uno»?

8.2 Что такое «макетная плата» и для чего она используется?

8.3 Для чего предназначен каждый из типов рельс на макетной плате?

8.4 Как соединить элементы на макетной плате последовательно?
8.5 Как соединить элементы на макетной плате параллельно?
8.6 В каком порядке должно выполняться подключение элементов на схеме?

Лабораторная работа №10
Разработка скетчей для Arduino

1 Цель работы

1.1 Научиться создавать скетчи (программное обеспечение на языке Си) для микроконтроллеров Arduino;

1.2 Получить навыки работы с платой Arduino Uno и макетной платой.

2 Литература

2.1 Благодаров, А. В. Программирование микроконтроллеров семейства 1986ВЕ9х компании Миландр / А. В. Благодаров. – Москва: Горячая Линия–Телеком, 2020. – 232 с. – URL: https://ibooks.ru/bookshelf/372218/reading. – Режим доступа: только для зарегистрированных пользователей. – Текст: электронный. – гл.1-2.

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 Программирование светофора

Разработать скетч для Arduino, в котором используется три одноцветных светодиода (красный, желтый, зеленый), которые должны загораться согласно следующим правилам:

- зеленый: горит 3 секунды,
- зеленый мигающий: «гаснет – горит» 3 раза по 0,25 секунды,
- желтый: горит 0,5 секунды,
- красный: горит 3 секунды,
- красный и желтый: горят 1 секунду и гаснут.

Для того, чтобы светодиод загорался, нужно подавать питание на пин, с которым он соединен, чтобы переставал гореть — переставать подавать питание.

Для создания светофора нужно разместить на рабочей поверхности плату Arduino Uno R3, макетную плату и набор светодиодов и резисторов.

5.2 Программирование плавно загорающегося и гаснущего светодиода

Разместить на рабочей поверхности плату Arduino Uno R3, макетную плату, светодиод и резистор на 100 Ом. Для питания нужно использовать цифровой пин, помеченный тильдой (например: ~11, ~10, ~9 и т. д.), т. к. эти пинам можно подавать аналоговую величину (ШИМ волну).

Для подачи сигнала на пин используется функция analogWrite() (для ее вызова analogWrite() не надо устанавливать тип вход/выхода функцией pinMode()).

5.3 Программирование проигрывателя мелодий

Для получения звука используется пьезоэлемент. Для взаимодействия с пьезоэлементом используется функция tone:

tone(пин, частота звука в герцах); // проигрывает звук заданной частоты

delay(длительность ноты в милисекундах);

Для проигрывания мелодии нужно разместить на рабочей поверхности плату Arduino Uno R3, макетную плату, резистор и пьезоэлемент.

5.4 Вывод времени работы в монитор последовательного интерфейса Разместить на рабочей поверхности плату Arduino Uno R3.

Реализовать отображение времени, прошедшего с запуска, в формате мм:сс.

6 Порядок выполнения работы

6.1 Перейти на сайт https://www.tinkercad.com/, авторизоваться.

6.2 Выполнить задания из п.5.

6.3 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 предназначены и когда вызываются функции setup() и loop()?

8.2 Для чего используется функция pinMode(...), какие параметры она принимает?

8.3 Для чего используется функция digitalWrite(...), какие параметры она принимает?

8.4 Какие функции используются для реализации программной задержки, какие параметры принимают эти функции?

8.5 Что такое «скетч»?

8.6 Как подключить библиотеки к скетчу?

Лабораторная работа №11
Разработка приложений для обработки файлов

1 Цель работы

1.1 Научиться использовать файловые потоки в приложении на C#;

1.2 Научиться применять классы для работы с файлами в приложениях на C#.

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2019. – 512 с. – 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 Разработать консольное приложение, запрашивающее у пользователя имя файла.

Если файл с указанным именем не существует, сообщить об этом пользователю.

Если файл с указанным именем существует, вывести его содержимое на экран.

5.2 Разработать консольное приложение, запрашивающее у пользователя имя файла и дописывающее в конец файла строки, пока пользователь не ввел строку end.

Если файл не существует, сообщить, что файл с указанным названием будет создан.

Если файл существует, сообщить, что файл открыт на дозапись.

5.3 Разработать консольное приложение, которое должно запускаться с командной строки с двумя параметрами: имя файла и искомый текст.

Если файл не существует, сообщить об этом пользователю.

Если файл существует, вывести те строки файла, в которых содержится указанный пользователем текст, с указанием их номеров в файле.

5.4 Разработать консольное приложение, которое должно запрашивать у пользователя логин и пароль.

Если введенный логин имеется в файле logins.txt, то запрашивать у пользователя повторный ввод логина до тех пор, пока не будет введен уникальный, затем запрашивать ввод пароля.

После указания логина, не существующего в файле logins.txt, записать введенные логин, пароль и дату регистрации в конец файла и сообщить

пользователю, что он зарегистрирован. Данные в файле должны храниться в виде логин;пароль;дата регистрации

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать консольное приложение C#.

6.2 Выполнить все задания из п.5 в одном решении.

6.3 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 файлах?

Лабораторная работа №12
Разработка приложений для поиска файлов

1 Цель работы

1.1 Научиться применять классы для работы с файлами и каталогами в приложениях на C#.

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2019. – 512 с. – 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 Разработать оконное приложение, запрашивающее у пользователя имя каталога (папки) и часть имени файла.

На форме должны быть расположены:

- поле ввода для указания части имени файла,
- кнопка «Выбрать каталог», при нажатии на которую пользователь может с использованием FolderBrowserDialog выбрать требуемую папку,
- метка с выводом имени папки (по умолчанию — текущая папка),
- кнопка «Найти»,
- список, в который при нажатии кнопки «Найти» должны выводиться полные имена файлов, соответствующие указанным параметрам поиска.

При каждом нажатии на «Найти» список вначале должен очищаться, чтобы отображать актуальные данные.

5.2 Внести изменения в приложение из п.5.1, добавив два переключателя, с помощью которых пользователь может указать, требуется искать файлы только в указанной папке или в указанной и вложенных в нее папках.

5.3 Внести изменения в приложение из п.5.1, добавив флажок и поля ввода для указания минимального и максимального размера файла в КБ, с помощью которых пользователь может ограничить поиск по размеру искомого файла, если он отметил флажок «Учитывать размер файла». Если флажок не отмечен, поля ввода должны быть недоступны.

5.4 Внести изменения в приложение из п.5.1, добавив флажок и поле ввода даты для указания самой ранней даты создания файла, с помощью которых пользователь может ограничить поиск по дате создания искомого файла, если он отметил флажок «Учитывать дату создания». Если флажок не отмечен, поле ввода даты должно быть недоступно.

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать оконное приложение C#.
6.2 Выполнить все задания из п.5 в одном решении.
6.3 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 отличие между классами Directory и DirectoryInfo?
8.2 В чем отличие между классами File и FileInfo?
8.3 Как получить список файлов и папок определенного каталога?
8.4 Как задать шаблон поиска и опции поиска файлов и папок?
8.5 Какие свойства класса FileInfo позволяют получить информацию о файле?

Лабораторная работа №13
Разработка приложений для сортировки файлов

1 Цель работы

1.1 Научиться применять классы для работы с файлами и каталогами в приложениях на C#.

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2019. – 512 с. – 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 Разработать консольное приложение, запрашивающее у пользователя имя каталога (папки) и сортирующее находящиеся в каталоге файлы по расширениям файлов.

Если каталог с указанным именем существует, перебрать список всех его файлов (без учета тех, что находятся во вложенных папках) и перенести каждый файл в папку с названием, совпадающим с расширением файла (например, для txt-файлов папка должна называться TXT, для pdf-файлов — PDF и т.д.).

Если в указанном каталоге нет папки с требуемым названием, она должна быть создана, если есть — перенести в нее файлы с соответствующим расширением.

5.2 Разработать консольное приложение, запрашивающее у пользователя имя первого (исходного) каталога и копирующее находящиеся в исходном каталоге файлы во второй каталог с группировкой по типам файлов (для каждого типа должен быть свой подкаталог).

Если исходный каталог с указанным именем существует, перебрать список всех его файлов (без учета тех, что находятся во вложенных папках) и перенести каждый файл во второй каталог в подкаталог с названием, совпадающим с типов файла. Во втором каталоге должны создаваться следующие подкаталоги:

- архивы (для zip, rar, 7z)
- изображения (для jpeg, jpg, bmp, png, gif)
- текстовые документы (для txt, rtf, odt, doc, docx)
- другое (для всех остальных типов файлов)

5.3 Разработать консольное приложение, запрашивающее у пользователя имя каталога (папки) и сортирующее находящиеся в каталоге файлы по дате изменения файлов.

Если каталог с указанным именем существует, перебрать список всех его файлов (без учета тех, что находятся во вложенных папках) и перенести каждый файл в папку с названием, соответствующим дате изменения файла:

каталог\год\номер месяца\день (например: c:\temp\download\2020\9\25)

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать оконное приложение C#.

6.2 Выполнить все задания из п.5 в одном решении.

6.3 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 проверить существование файла по его имени?

8.2 Как проверить существование каталога по его имени?

8.3 Какие методы позволяют создавать, удалять, копировать и переносить каталоги?

8.4 Какие методы позволяют создавать, удалять, копировать и переносить файлы?

Лабораторная работа №14
Разработка утилиты для поиска дубликатов файлов

1 Цель работы

1.1 Научиться получать и анализировать информацию о файлах в приложениях на C#.

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2019. – 512 с. – 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 Разработать оконное приложение, выполняющее поискдубликатов файлов по имени и отображение полного имени найденных файлов.

5.2 Разработать оконное приложение, выполняющее поиск дубликатов файлов по имени с отображением имени, пути, размера и даты изменения найденных дубликатов.

5.3 Разработать оконное приложение, выполняющее поиск дубликатов файлов с возможностью удаления выбранных.

5.4 Разработать оконное приложение, выполняющее поиск дубликатов файлов по набору критериев (имени, размеру, дате изменения).

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать оконное приложение C#.

6.2 Выполнить все задания из п.5 в одном решении.

6.3 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 свойство FileInfo возвращает имя файла?

8.2 Какое свойство FileInfo возвращает расширение файла?

8.3 Какое свойство FileInfo возвращает полное имя файла?
8.4 Какое свойство FileInfo возвращает дату изменения файла?

Лабораторная работа №15
Разработка утилиты для просмотра изображений

1 Цель работы

1.1 Научиться отображать файлы-изображения в приложениях на C#.

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2019. – 512 с. – URL: https://ibooks.ru/bookshelf/366634/reading. – Режим доступа: только для зарегистрированных пользователей. – Текст : электронный. – 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 Создать оконное приложение для отображения растрового файла Добавить в приложение:

- основное меню с пунктом «Файл» и подпунктами «Открыть» и «Выход»,
- StatusStrip (строку состояния, в которую может быть введена информация о файле),
- PictureBox для отображения выбранного файла.

При нажатии на пункт меню «Выход» приложение должно закрываться.

При нажатии на пункт меню «Открыть» требуется открывать окно выбора файла-изображения (поставить фильтр на расширения .bmp, .jpg, .jpeg, .png).

PictureBox должен заполнять всю форму.

После открытия файла:

- заголовок формы должен изменяться на имя файла (без имени папки),
- в StatusStrip должны отображаться размер выбранного файла и ширина и высота выбранного файла в пикселях.

5.2 Создать оконное приложение, отображающее изображение в исходном размере Добавить в приложение:

- основное меню с пунктом «Файл» и подпунктами «Открыть» и «Выход»,
- Panel (элемент управления, у которого доступны полосы прокрутки),
- PictureBox (разместить его в Panel, т.к. у самого PictureBox нет полос прокрутки, которые потребуются при просмотре файла, не вмещающего в размеры PictureBox).

Панель должна заполнять всю форму. После открытия файла заголовок формы должен изменяться на имя файла (без имени папки).

При запуске форма должна открываться на полный экран.

5.3 Создать оконное приложение с возможностью масштабирования изображения Добавить в приложение из п.5.2 пункт меню «Вид» и подпункты:

- 100% (размер изображения должен соответствовать исходному)
- 200% (размер изображения должен увеличиваться вдвое по ширине и высоте)
- 50% (размер изображения должен уменьшаться вдвое по ширине и высоте)
- Во весь экран (форма должна открываться в полный экран без границ, размер изображения должен подгоняться под размер формы)

Выход из полноэкранного режима должен выполняться при нажатии кнопки мыши.

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать оконное приложение C#.

6.2 Выполнить все задания из п.5 в одном решении.

6.3 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 предназначен класс Bitmap?

8.2 Для чего предназначен элемент управления PictureBox?

8.3 Для чего предназначен элемент управления Panel?

8.4 Как получить высоту и ширину изображения Bitmap?

8.5 Какие способы отображения изображений допустимы в PictureBox?

Лабораторная работа №16
Разработка утилиты Диспетчер задач

1 Цель работы

1.1 Научиться получать и отображать информацию о запущенных процессах в приложениях на C#.

2 Литература

2.1 Гуриков, С. Р. Введение в программирование на языке Visual C# : учебное пособие / С. Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2020. – 447 с. – URL: https://znanium.com/catalog/product/1092167. – Текст: электронный. – п.11.1.

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 информации о процессах

5.1.1 Создать оконное приложение со следующими элементами управления:

- MenuStrip с пунктами «Запустить новую задачу», «Выход»,
- ToolStrip с кнопками «Обновить», «Снять задачу», «Завершить дерево процессов» (подписи должны отображаться, для этого сменить у кнопок DisplayStyle на Text),
- ContextMenuStrip с пунктами «Снять задачу», «Завершить дерево процессов».
- StatusStrip с меткой для отображения количества запущенных процессов,
- TabControl со вкладками «Процессы» и «Приложения» (TabControl должен занимать всю форму).

5.1.2 Добавить во вкладку «Процессы» ListView и настроить его:

- изменить стиль заполнения (Dock) на Fill,
- добавить в список столбцы с названиями: Имя, ИД процесса, Память,
- изменить вид (View) на Details,
- связать ListView с созданным контекстным меню.

5.1.3 Реализовать при загрузке формы и при нажатии «Обновить» отображение информации о запущенных процессах в столбцах ListView и количества запущенных процессов в метке.

5.2 Отображение информации о запущенных приложениях

5.2.1 Добавить во вкладку «Приложения» ListView и настроить его:

- изменить стиль заполнения (Dock) на Fill,
- добавить в список столбцы с названиями: Приложение, Время запуска,
- изменить вид (View) на Details.

5.2.2 Создать метод, заполняющий список запущенных приложений, добавив в ListView те процессы, у которых заполнено свойство MainWindowTitle (заголовок окна).

В столбце Приложение должно отображаться значение свойства MainWindowTitle, в столбце Время запуска – значение свойства StartTime.

5.2.3 Реализовать при загрузке формы и при нажатии «Обновить» отображение информации о запущенных приложениях в столбцах ListView, вызвав созданный метод.

5.3 Завершение процесса

5.3.1 Создать метод, при вызове которого для завершения процесса нужно у процесса с указанным ИД, вызвать метод Kill() и метод WaitForExit().

5.3.2 При нажатии на «Снять задачу» требуется завершать выделенный процесс и в случае завершения обновлять список процессов.

5.4 Запуск процесса

5.4.1 Реализовать открытие новой формы при нажатии на «Запустить новую задачу».

В форме должно быть поле ввода для указания исполняемого файла, кнопки ОК, Обзор и Отмена.

5.4.2 При нажатии на ОК должно запускаться приложение, указанное в поле ввода (если имя некорректно, сообщать об этом пользователю). После нажатия ОК и Отмена форма должна запускаться, а список процессов в главной форме — обновляться.

5.4.3 При нажатии на Обзор должно открываться окно выбора файла с расширением .exe. После выбора файла его полное имя должно отображаться в поле ввода.

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать оконное приложение C#.

6.2 Выполнить все задания из п.5 в одном решении.

6.3 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 требуется подключить для работы с процессами?

8.2 Какой метод запускает процесс?

8.3 Какой метод завершает процесс?

8.4 Как получить список запущенных процессов?

8.5 Как получить процесс по его идентификатору?

Лабораторная работа №17
Разработка утилиты Файловый менеджер

1 Цель работы

1.1 Научиться применять элементы управления для отображения файлов и папок в приложениях на C#.

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2019. – 512 с. – URL: https://ibooks.ru/bookshelf/366634/reading. – Режим доступа: только для зарегистрированных пользователей. – Текст : электронный. – 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 Настройка внешнего вида файлового менеджера

5.1.1 Создать оконное приложение со следующими элементами управления:

- ToolStrip (предназначен для отображения в верхней части формы строки с именем папки и кнопок),
- SplitContainer (разделяет форму на две области: в левой части окна будет дерево папок, в правой — содержимое текущей папки, имя которой показано в поле ввода),
- TreeView (предназначен для отображения дерева папок),
- ListView (предназначен для отображения содержимого текущей папки),
- ImageList (предназначен для хранения пиктограмм, которые будут отображаться в проводнике).

5.1.2 Настроить элемент управления ImageList:

- добавить в него набор пиктограмм, которые должны отображаться в файловом менеджере,
- изменить размер изображений (ImageSize) на 24; 24,
- изменить разрядность изображений (ColorDepth) на Depth32Bit.

5.1.3 Настроить элемент управления TreeView:

- изменить стиль заполнения на Fill,
- изменить ImageList на созданный набор изображений.

5.1.4 Настроить элемент управления ListView:

- изменить стиль заполнения на Fill,
- изменить SmallImageList и LargeImageList на созданный набор изображений,
- добавить в список столбцов (Columns) 3 столбца с названиями Имя, Тип, Дата изменения,

- изменить вид (View) на Details.

5.1.5 Добавить в ToolStrip поле ввода (увеличить его ширину до 200 пикселей и указать как значение по умолчанию C:\).

5.2 Реализовать отображение содержимого указанной папки

5.2.1 Создать метод, который должен заполнять элемент ListView списком папок (имя текущей папки должно браться из поля ввода, номер пиктограммы должен соответствовать пиктограмме папки).

5.2.2 Реализовать вызов метода при нажатии клавиши Enter в поле ввода. Перед вызовом метода проверять, что указанная папка существует, иначе сообщать пользователю об ошибке и изменять путь на исходный корректный вариант.

5.2.3 Внести изменения в созданный метод заполнения ListView, чтобы после списка папок аналогичным образом заполнялся список файлов (номер пиктограммы должен соответствовать пиктограмме файла).

5.3 Реализовать отображение дерева папок

5.3.1 Реализовать отображение списка дисков в дереве папок при загрузке формы.

5.3.2 Реализовать добавление вложенных узлов в дереве папок.

5.4 Реализовать изменение внешнего вида содержимого папки.

Добавить в приложение контекстное меню с пунктом Вид и подпунктами:

- Таблица (у ListView за это отвечает стиль Details),
- Крупные значки (у ListView за это отвечает стиль LargeIcon),
- Список (у ListView за это отвечает стиль List),
- Мелкие значки (у ListView за это отвечает стиль SmallIcon)
- Плитка (у ListView за это отвечает стиль Tile).

При нажатии подпункт меню должен помечаться флажком как выбранный, стиль ListView – изменяться на выбранный.

Связать контекстное меню с ListView.

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать оконное приложение C#.

6.2 Выполнить все задания из п.5 в одном решении.

6.3 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 получить список дисков?

8.2 Для чего предназначен элемент управления TreeView?

8.3 Для чего предназначен элемент управления ListView?
8.4 Для чего предназначен элемент управления ImageList?
8.5 Какие способы отображения элементов допустимы в ListView?

Лабораторная работа №18
Разработка утилиты для анализа дисков

1 Цель работы

1.1 Научиться получать и отображать статистическую информацию о дисковом пространсте в приложениях на C#.

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2019. – 512 с. – URL: https://ibooks.ru/bookshelf/366634/reading. – Режим доступа: только для зарегистрированных пользователей. – Текст : электронный. – 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 Отображение информации о дисках

5.1.1 Создать оконное приложение со следующими элементами управления:

- SplitContainer (разделяет форму на две области: в верхней части окна будет информация о дисках, в нижней — о текущей папке, имя которой показано в поле ввода),
- ListView (предназначен для отображения информации о дисках системы).

5.1.2 Настроить элемент управления ListView:

- изменить стиль заполнения (Dock) на Fill,
- добавить в список столбцов (Columns) 5 столбцов с названиями Имя, Тип, Общий размер, Использовано %, Свободно,
- изменить вид (View) на Details.

5.1.3 Реализовать при загрузке формы отображение информации о дисковом пространстве в столбцах ListView, используя следующий код:

5.2 Отображение количества и размера файлов и папок

5.2.1 Добавить в нижнюю панель:

- ToolStrip (предназначен для отображения в строки с именем папки),
- TabControl (должен заполнять всю нижнюю панель, предназначен для отображения информации о папки в различных видах в отдельных вкладках).

У TabControl должны быть следующие вкладки:

- Общая информация,
- Топ 10 файлов,
- Объем файлов.

5.2.2 При вводе имени папки и нажатии Enter в поле ввода должно проверяться, что указанная папка существует. Если папка существует, отобразить в первой вкладке «Общая информация» следующую информацию о выбранной папке:

- количество файлов (всех, включая вложенные),
- количество папок (всех, включая вложенные),
- размер папки,
- % занятого папкой места на диске, где расположена папка.

5.3 Отображение списка самых больших файлов

5.3.1 Добавить во вкладку «Топ 10 файлов» ListView со столбцами:

- имя файла,
- размер файла,
- расположение файла (полный путь к нему),
- дата последнего изменения.

5.3.2 После указания имени папки в ToolStrip выводить в ListView 10 самых больших файлов.

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать оконное приложение C#.

6.2 Выполнить все задания из п.5 в одном решении.

6.3 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 класс предоставляет информацию о дисках?

8.2 Какой класс предоставляет информацию о файлах?

8.3 Для чего предназначен элемент управления Chart?

8.4 Какой метод позволяет просуммировать значения в списке?

8.5 Какие методы позволяют отсортировать список по возрастанию и убыванию?

Лабораторная работа №19
Разработка утилиты Архиватор

1 Цель работы

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

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2019. – 512 с. – 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 Создание приложения для архивации

Создать оконное приложение с кнопкой «Создать архив». При нажатии на кнопку «Создать архив» должно открываться окно сохранения файла (указать фильтр на файлы с расширением .zip) и выполняться архивирование папки и сохранение архива под указанным именем.

5.2 Создание приложения для распаковки архива

Создать оконное приложение с полем ввода имени архива и кнопкой «Разархивировать». При нажатии на кнопку «Разархивировать» должно открываться окно выбора папки и должна выполняться распаковка архива в указанную папку.

5.3 Создать оконное приложение с полем ввода имени файла-архива и кнопкой «Открыть архив». При нажатии на кнопку «Открыть архив» отобразить в ListBox содержимое архива (список файлов и папок).

5.4 Создать оконное приложение с кнопкой «Добавить в архив», полем ввода имени файла-архива и полем ввода имени добавляемого в архив файла. При нажатии на кнопку «Добавить в архив» добавлять указанный файл в архив.

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать оконное приложение C#.

6.2 Выполнить все задания из п.5 в одном решении

6.3 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

Лабораторная работа №20
Разработка утилиты скринсейвер

1 Цель работы

1.1 Научиться разрабатывать оконные приложения-заставки, используя таймер.

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2019. – 512 с. – URL: https://ibooks.ru/bookshelf/366634/reading. – Режим доступа: только для зарегистрированных пользователей. – Текст : электронный. – 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 заставки «Пустой экран»

Создать оконное приложение. Задать в качестве фонового цвета окна черный цвет. Убрать у формы границы. При открытии форма должна открываться на весь экран. Закрытие формы реализовать при движении мыши.

5.2 Создание заставки «Часы»

Создать приложение согласно описанию в п.5.1.

Добавить на форму метку, в которой реализовать отображение текущего времени (получить текущее время в формате чч:мм:сс от значения DateTime.Now). Настроить шрифт (цвет, размер от 40, начертание). Метка должна находиться в центре экрана (положение требуется определять программно при загрузке формы, используя значения свойств Width и Height). Значение на метке должно меняться ежесекундно (чтобы соответствовало текущему времени).

5.3 Изменение заставки «Часы»

Внести изменения в приложение из п.5.2: метка со временем должна перемещаться по экрану в пределах окна (траектория — как у бильярдного шара).

5.4 Создание заставки «Фотографии»

Создать приложение согласно описанию в п.5.1.

Добавить на форму элемент для отображения изображений PictuteBox (должен занимать всю форму, режим изменения изображения: Zoom).

Изображения должны браться из определенной папки images (задать путь к ней программно) и меняться каждые 10 секунд.

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать оконное приложение C#.
6.2 Выполнить все задания из п.5 в одном решении.
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 «утилита»?
8.2 Что такое «скринсейвер»?
8.3 Какое расширение должно быть у файлов-заставок в Windows?

Лабораторная работа №21
Разработка утилиты для вычисления хэш-суммы файлов

1 Цель работы

1.1 Научиться вычислять хэш-суммы файлов в приложениях на C#, используя встроенные алгоритмы шифрования.

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2019. – 512 с. – 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 Вычисление хэш-суммы файла по алгоритму MD5.

5.1.1 Разработать оконное приложение. На форме должны быть расположены поле ввода имени файла, кнопка «...», при нажатии на которую пользователь может с использованием OpenFileDialog выбрать требуемый файл, кнопка «Вычислить».

5.1.2 Создать метод для вычисления хэш-суммы указанного в параметрах файла file.

5.1.3 Реализовать вывод вычисленного хэша в метке.

5.2 Вычисление хэш-суммы файла по алгоритму SHA-1.

Добавить в приложение из п.5.1 вычисление хэш- суммы с помощью алгоритма SHA-1 и ее вывод в метку на форме.

5.3 Разработать оконное приложение, отображающее хэш-суммы всех файлов, находящихся в указанной пользователем папке. Данные о файлах (имя, хэш-сумма, путь, размер, дата изменения) должны выводиться в ListView.

5.4 Разработать оконное приложение, находящее дубликаты файлов по хэш-сумме. Найденные файлы-дубликаты должны выводиться в ListView с указанием имени и пути к файлу.

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать оконное приложение C#.

6.2 Выполнить все задания из п.5 в одном решении.

6.3 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 хэш-сумма?
8.2 Для чего применяются хэш-суммы файлов?
8.3 Какие алгоритмы применяются для вычисления хэш-суммы файлов?

Лабораторная работа №22
Разработка утилиты Менеджер паролей

1 Цель работы

1.1 Научиться выполнять шифрование и дешифрование данных в приложениях на C#, используя встроенные алгоритмы шифрования.

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2019. – 512 с. – 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 Отображение регистрационных данных пользователя

Разработать оконное приложение, отображающее данные из файла passwords.txt в ListView (столбцы: сайт/приложение, логин, пароль). Данные должны считываться из текстового файла passwords.txt (каждый набор значений – на отдельной строке, значения в наборе отделяются друг от друга точкой с запятой).

5.2 Добавление регистрационных данных пользователя

Реализовать в приложении из п.5.1 возможность добавления новых регистрационных данных. У пользователя должны запрашиваться сайт/приложение, логин, пароль. Введенные данные должны записываться в конец текстового файла passwords.txt при нажатии на кнопку «Добавить».

5.3 Генерация пароля

Реализовать в приложении из п.5.2 возможность генерации пароля заданной длины из символов английского алфавита при нажатии на кнопку «Сгенерировать пароль». Длина пароля указывается в поле ввода NumericUpDown. Сгенерированный пароль должен отображаться в поле ввода пароля.

5.4 Шифрование данных

Реализовать в приложении из п.5.2 шифрование пароля.

5.5 Дешифрование данных

Реализовать в приложении из п.5.4 отображение расшифрованных паролей.

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать оконное приложение C#.

6.2 Выполнить все задания из п.5 в одном решении.

6.3 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 «шифрование»?

8.2 Что такое «дешифрование»?

8.3 Что такое «AES»?

8.4 Какими могут быть размеры ключа в алгоритме AES?

8.5 Какое пространство имен требуется подключить для применения стандартных алгоритмов шифрования?

Лабораторная работа №23
Изучение процесса разработки DLL

1 Цель работы

1.1 Изучить процесс создания, применения, тестирования и документирования библиотеки классов в Microsoft Visual Studio.

2 Литература

2.1 Шарп, Д. Microsoft Visual C#. Подробное руководство. 8-е издание / Д. Шарп. – Санкт-Петербург: Питер, 2017. – URL: https://ibooks.ru/reading.php?productid=354026, только для зарегистрированных пользователей. – Загл. с экрана. – 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 Создание библиотеки классов

Создать новый проект типа библиотека классов, назвать его LabWorkLibrary. Переименовать стандартный класс Class1 в Maths (класс, в котором будет храниться набор статических открытых методов и констант, использующихся в математических выражениях). Добавить в класс следующие элементы:

- метод для вычисления суммы двух чисел;
- метод для вычисления разности двух чисел;
- метод для вычисления произведения двух чисел;
- метод для вычисления частного двух чисел (должен генерировать исключение при попытке деления на 0);
- метод для вычисления площади прямоугольника (должен генерировать исключение при некорректных исходных данных);
- константу PI = 3.1415.

Пересобрать решение и проверить, что в папке bin\Debug появился файл библиотеки.

5.2 Использование библиотеки классов

5.2.1 Создать новое решение с консольным проектом (здесь будет выполняться тестирование созданной библиотеки).

5.2.2 Подключить к проекту ссылку на созданную библиотеку, указав путь к ней в разделе Обзор в менеджере ссылок (меню Проект — Добавить ссылку — Обзор или Обозреватель решений — проект — Ссылки — Обзор).

5.2.3 В коде проекта подключить библиотеку, используя using ИмяБиблиотеки.

Использовать методы и константу библиотеки в консольном приложении.

5.3 Документирование библиотеки классов

5.3.1 Добавить в код библиотеки XML-комментарии ко всем элементам класса Maths (для генерации шаблона комментария требуется нажать /// над всеми классами и его элементами).

5.3.2 Пересобрать проект с библиотекой и консольное решение для того, чтобы проверить, отображаются ли комментарии. Для просмотра всех комментариев в проекте, к которому подключена библиотека, можно открыть ее в обозревателе объектов, дважды нажав на библиотеку в обозревателе решений.

5.3.3 Для того, чтобы XML-комментарии отображались в приложении, требуется открыть свойства проекта библиотеки классов и в разделе «Сборка» поставить флажок «XML-файл документации».

Пересобрать проект с библиотекой и консольное решение для того, чтобы проверить, отображаются ли комментарии. Изучить содержимое папки bin\Debug.

5.4 Структурирование кода с использованием пространств имен

5.4.1 Добавить в библиотеку классов папку UserData и создать в ней в отдельных файлах:

- перечисление «Роль пользователя» со значениями Покупатель, Менеджер, Администратор,
- класс «Пользователь».

Добавить в класс «Пользователь»:

- закрытые поля id (число), логин (строка), пароль (строка), привилегированный (логическое значение),
- открытый метод смены пароля на указанный в параметрах. Метод смены пароля должен возвращать false, если новый пароль совпадает с исходным или состоит из пустой строки, в остальных случаях метод возвращает true и изменяет пароль на новый,
- открытое свойство для чтения и изменения логина. Изменение должно выполняться, если логин — не пустая строка/строка из пробелов,
- открытый метод смены уровня привилегированности в зависимости от переданной в параметрах роли (администратор должен иметь привилегии, другие пользователи — не имеют привилегий).

При разработке снабжать код XML-документацией.

5.4.2 Пересобрать проект с библиотекой и консольное решение.

Подключить в консольном приложении пространство имен, в котором в библиотеке описаны роль пользователя и пользователь, используя using ИмяБиблиотеки.ИмяПапки.

Создать в консольном приложении объекты типов данных перечисление и класс, описанных в библиотеке в UserData.

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать на C# проект библиотеки классов.

6.2 Выполнить все задания из п.5. При разработке считать, что пользователь ввел данные требуемого типа, остальные возможные ошибки обрабатывать. При выполнении заданий использовать минимально возможное количество команд и переменных и выполнять форматирование и рефакторинг кода.

6.3 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 «библиотека классов»?

8.2 Какое расширение у файлов библиотек?

8.3 Как на C# создать библиотеку классов в Visual Studio?

8.4 Как подключить библиотеку к проекту?

8.5 Что такое «XML-документация»?

8.6 Какие действия нужно выполнить, чтобы XML-документация была видна при подключении библиотеки в стороннем решении?

Лабораторная работа №24
Использование потоков

1 Цель работы

1.1 Научиться разрабатывать многопоточные приложения на C#;

1.2 Научиться создавать и применять фоновые и основные потоки и выполнять обмен данных между ними в программах на C#.

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2019. – 512 с. – URL: https://ibooks.ru/bookshelf/366634/reading. – Режим доступа: только для зарегистрированных 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 Создание и приостановка потоков

Создать консольное приложение, где в основном потоке Main создается новый поток, исполняющий метод Print, который непрерывно печатает символ 1 в бесконечном цикле. Одновременно главный поток непрерывно печатает символ 0 в бесконечном цикле. В конце каждого шага цикла должна выполняться задержка:

Thread.Sleep(1000);

Вместо 1000 в разных циклах указать разные значения — поток будет приостанавливаться на указанное количество миллисекунд.

5.2 Настройка приоритетов потоков

5.2.1 Создать консольное приложение, в который добавить метод WriteString, принимающий объект N и в цикле выводящий его 1000 раз:

5.2.2 В методе Main запустить на выполнение 4 потока, каждый из которых будет вызывать метод WriteString и с его помощью выводить свой номер в окно консоли. При создании потоков в Main установить для них различные приоритеты.

5.2.3 Для большей наглядности в начале метода WriteString реализовать вывод на отдельной строке сообщения, что «Поток N запущен», в конце метода – «Поток N завершен» (вместо N отображать значение параметра N).

5.3 Обмен данными между потоками

Создать консольное приложение, в котором происходит обмен данными между потоками Main() и MyThread() через статическое поле (глобальную переменную) commonVar.

Второй поток исполняется до того момента, пока в первом переменной str не будет присвоено значение «x».

5.4 Создание пула потоков

Создать консольное приложение, в которое добавить метод, выполняющий поиск и вывод на экран делителей переданного в метод числа в следующем формате:

y делится нацело на x.

Пример:

6 делится нацело на 1.

6 делится нацело на 2.

6 делится нацело на 3.

6 делится нацело на 6.

Добавить в приложение пул потоков, в который добавить 10 потоков (должны вызывать метод поиска делителей, для каждого потока передавать свое число, например числа от N до N+10).

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать оконное приложение C#.

6.2 Выполнить все задания из п.5 в одном решении.

6.3 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 отличие потока от процесса?

8.2 Какие преимущества дает многопоточная архитектура?

8.3 Какие существуют основные средства синхронизации потоков?

8.4 Каким образом на однопроцессорных компьютерах исполняются многопоточные приложения?

8.5 Для чего в C# используется класс Thread?

Лабораторная работа №25
Разработка и вызов асинхронных методов

1 Цель работы

1.1 Научиться реализовывать и запускать асинхронные операции на C#;

1.2 Научиться выполнять вычисления, используя асинхронные операции на C#.

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2019. – 512 с. – URL: https://ibooks.ru/bookshelf/366634/reading. – Режим доступа: только для зарегистрированных 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 метод для вычисления a в степени x (a – любое, x – целое). Вычисление реализовать без использования стандартной математической функции. Метод должен выводить результат вычислений на экран в следующем виде (вместо переменных должны быть значения переменных):

a^x=result

Вызвать метод для вычисления a^x, используя последовательный асинхронный вызов для трех различных наборов параметров методов.

5.2 Вызвать метод из п.5.1 для вычисления a^x, используя параллельный асинхронный вызов для трех различных наборов параметров методов.

5.3 Создать метод для вычисления a в степени x (a – любое, x – целое). Вычисление реализовать без использования стандартной математической функции. Метод должен возвращать результат вычислений.

Вызвать метод для вычисления a^x, используя параллельный асинхронный вызов для трех различных наборов параметров методов. Полученные результаты вывести на экран.

5.4 Создать асинхронный метод, вычисляющий:

(a1^x1+a2^x2)/(a3^x3- a4^x4)

Для вычислений использовать последовательный вызов метода из п.5.3.

5.5 Добавить обработку исключений в метод из п.5.4. При перехвате исключений выводить на экран сообщение о возникшем исключении.

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать консольное приложение C#.

6.2 Выполнить все задания из п.5 в одном решении.

6.3 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 ключевые слова используются в C# для работы с асинхронными вызовами?

8.2 Какие типы возврата могут быть у асинхронных методов и для чего предназначен каждый из типов?

8.3 Как вызвать метод в асинхронном режиме?

8.4 Как указать, что в методе могут быть асинхронные вызовы?

8.5 Как обработать исключения, возникшие в асинхронных вызовах?

Лабораторная работа №26
Обмен данными

1 Цель работы

1.1 Научиться реализовывать и запускать асинхронные операции на C#;

1.2 Научиться выполнять HTTP-запросы, используя асинхронные операции на C#.

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2019. – 512 с. – URL: https://ibooks.ru/bookshelf/366634/reading. – Режим доступа: только для зарегистрированных 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

В качестве источника строковых данных в работе использовать ресурс https://jsonplaceholder.typicode.com/ — сайт, предоставляющий тестовые данные в формате JSON.

В качестве источника файлов в работе использовать страницу http://arcotel.ru/bibl/pom_stud.php — содержит набор документов.

5.1 Выполнить загрузку строковых данных по сети

Изучить данные по следующей ссылке: https://jsonplaceholder.typicode.com/posts.

Выполнить загрузку данных в строковом формате.

Для загрузки использовать класс WebClient из пространства имен System.Net:

```
var client = new WebClient();
var response = client.DownloadString(адрес страницы данных);
```

5.2 Десериализовать полученные данные

Изучить данные по ссылке https://jsonplaceholder.typicode.com/posts, создать класс для описания данных.

После загрузки строки десериализовать полученные данные в список с элементами разработанного типа данных.

5.3 Отображение полученных данных

5.3.1 Отобразить данные из списка в виде таблицы на форме.

Добавить возможность фильтрации по идентификатору пользователя, чтобы выводились только посты определенного пользователя.

В нижней части должно отображаться общее количество записей, количество пользователей, написавших посты, и среднее количество постов от пользователя.

5.3.2 Предоставить пользователю возможность ввода указания идентификатора, после ввода должны загружаться посты пользователя с соответствующим идентификатором.

Источник данных: https://jsonplaceholder.typicode.com/users/идентификатор/posts

5.4 Выполнить загрузку указанного файла по сети

Для загрузки используется метод DownloadFile:

```
client.DownloadFile(адрес файла, имя файла на компьютере пользователя)
```

Адрес файла указывается пользователем в поле ввода, новое имя – через диалоговое окно сохранения файла.

Проверить загрузку текстовых документов, pdf, изображений

5.5 Получение списка файлов на странице

Загрузить страницу http://arcotel.ru/bibl/pom_stud.php как строку.

Найти, используя регулярное выражения, все ссылки на файлы с расширением docx.

Отобразить найденные адреса файлов на форме (в многострочном поле ввода или ListBox).

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать приложение C#.

6.2 Выполнить все задания из п.5 в одном решении.

6.3 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 WebClient?

8.2 Какие методы для загрузки данных поддерживаются в классе WebClient?

8.3 Какое пространство имен требуется подключить для работы с WebClient?

Лабораторная работа №27
Сетевое программирование сокетов

1 Цель работы

1.1 Научиться примененять сокеты в клиент-серверных приложениях на C#.

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2019. – 512 с. – URL: https://ibooks.ru/bookshelf/366634/reading. – Режим доступа: только для зарегистрированных 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 Создание клиент-серверного приложения на сокетах TCP

5.1.1 Создать консольное приложение-сервер (назвать его SocketTcpServer), ожидающее подключений от клиента и выводящее дату получения и полученное сообщение на консоль.

5.1.2 Создать консольное приложение-клиент (назвать его SocketTcpClient), отправляющее на сервер введенный пользователем текст.

5.2 Создание клиент-серверного приложения с использованием классов TcpClient и TcpListener

5.2.1 Создать консольное приложение-клиент (назвать его TcpClientApp), выполняющее чтение данных из потока с сервера.

5.2.2 Создать консольное приложение-сервер (назвать его TcpListenerApp), ожидающее подключений от клиента и отправляющее им сообщение.

5.3 Создание многопоточного клиент-серверного приложения

5.3.1 Создать консольное приложение-клиент (назвать его ConsoleClient), отправляющее на сервер имя пользователя и введенный пользователем текст.

5.3.2 Создать консольное приложение-сервер (назвать его ConsoleServer), ожидающее подключений от клиентов и выводящее их. На сервере после имени клиента в скобках должны отображаться дата и время (формат даты и времени: чч:мм:сс дд.мм.гггг).

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать консольное приложение C#.

6.2 Выполнить все задания из п.5 в одном решении.

6.3 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 «сокет»?
8.2 Каков алгоритм работы сервера, использующего сокеты?
8.3 Каков алгоритм работы клиента, использующего сокеты?
8.4 Какие пространства имен требуется подключить для работы с сокетами?
8.5 Какие параметры требуется указать при создании сокета?
8.6 Как выполнить получение данных с использованием сокетов?

Лабораторная работа №28
Разработка приложения для загрузки и отправки данных по сети

1 Цель работы

1.1 Научиться разрабатывать RESTfulAPI для доступа к БД из приложения C#.

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2019. – 512 с. – URL: https://ibooks.ru/bookshelf/366634/reading. – Режим доступа: только для зарегистрированных 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 Создание проекта Web-API

5.1.1 Создать веб-приложение ASP.NET. В окне создания веб-приложения выбрать тип проекта «Веб-API» и убрать флажок «Настроить для HTTPS».

5.1.2 Для просмотра доступных методов запустить приложение и на открывшейся странице нажать на гиперссылку с подписью «API»: после перехода на странице отобразится список методов с комментариями.

5.1.3 Добавить и протестировать контроллер, возвращающий сумму двух чисел, переданных в параметрах метода.

5.2 Создание модели и контроллера EntityFramework

5.2.1 Добавить в папку Models модель БД ADO.NET EDM, используя Entity Framework. После создания модели пересобрать и сохранить проект.

5.2.2 В контекстном меню папки Controllers выбрать Добавить — Контроллер.

Добавить в проект «Контроллер Web API 2 с действиями, использующий Entity Framework».

Выбрать для контроллера класс модели (таблицу Товары из БД) и класс контекста данных.

5.2.3 Проверить в браузере работу методов GET созданного контроллера.

5.3 Разработка собственных GET-методов

5.3.1 Добавить в контроллер метод с атрибутом маршрута Route для возврата списка значений, отсортированных по возрастанию по одному из столбцов.

5.3.2 Проверить в браузере работу созданного метода сортировки.

5.4 Передача параметров в метод GET

5.4.1 Добавить в контроллер метод с атрибутом маршрута Route для фильтрации данных по одному из двух столбцов или по обоим столбцам (зависит от переданных пользователем параметров).

5.4.2 Проверить в браузере работу созданного метода фильтрации (параметры указать в адресной строке браузера при вызове метода).

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать консольное приложение C#.

6.2 Выполнить все задания из п.5 в одном решении.

6.3 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 REST-запрос?

8.2 Что такое RESTful?

8.3 Для чего в контроллере используется метод GET?

8.4 Для чего в контроллере используется метод POST?

8.5 Для чего в контроллере используется метод PUT?

8.6 Для чего в контроллере используется метод DELETE?

Лабораторная работа №29
Вызов сервисов

1 Цель работы

1.1 Научиться проверять работоспособность RESTful API в клиентском приложении.

1.2 Научиться выполнять тестирование RESTful API методом черного ящика.

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2019. – 512 с. – URL: https://ibooks.ru/bookshelf/366634/reading. – Режим доступа: только для зарегистрированных 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

Для выполнения тестирования должен быть запущен проект Web-API (ЛР №28).

5.1 Создание клиентского приложения

5.1.1 Создать консольное приложение .Net Framework (для .NET нужно изменить методы работы с HTTP-запросами).

5.1.2 Создать класс с открытыми автореализуемыми свойствами на чтение и запись для работы с объектами, отображающими данные таблицы Games.

Пример создания класса для таблицы Products:

```
public class Product
{
      public string Id { get; set; }
      public string Name { get; set; }
      public double Price { get; set; }
}
```

5.1.3 Добавить в класс Program метод для отображения значений свойств переданного в параметрах объекта класса из п.3.1.2.

5.1.4 Добавить в класс Program статическое поле client:

static HttpClient client = new HttpClient();

5.1.5 Реализовать в методе Main или отдельном методе, который вызвать в Main, настройку параметров подключения у объекта client:

```
client.BaseAddress = new Uri("http://localhost:порт/");
client.DefaultRequestHeaders.Accept.Clear();
client.DefaultRequestHeaders.Accept.Add(new
```

```
MediaTypeWithQualityHeaderValue("application/json"));
```

5.1.6 Для того, чтобы приложение получало данные в формате JSON, нужно загрузить библиотеку Microsoft.AspNet.WebApi.Client через диспетчер пакетов NuGet.

5.2 Создать асинхронный метод, возвращающий объект созданного класса по идентификатору, используя API (метод GET с параметром).

```
static async Task<Класс> GetКлассAsync(int id)
{
   Класс объект = null;
   HttpResponseMessage response = await client.GetAsync($"api/Контроллер/{id}");
   if (response.IsSuccessStatusCode)
   {
      объект = await response.Content.ReadAsAsync<Класс>();
   }
   return объект;
}
```

Вызвать созданный асинхронный метод в клиентском приложении и вывести информацию о полученном объекте на экран.

5.3 Создать асинхронный метод, удаляющий объект из таблицы по идентификатору, используя API (метод DELETE).

```
static async Task<HttpStatusCode> DeleteКлассAsync(int id)
{
                              HttpResponseMessage response = await
            client.DeleteAsync($"api/Контроллер/{id}");
   return response.StatusCode;
}
```

Вызвать созданный асинхронный метод в клиентском приложении.

Вывести на экран код http-ответа и сообщение, удалось ли удалить запись.

Проверить, что в БД добавлена удалена строка с указанным в клиентском приложении идентификатором.

5.4 Создать асинхронный метод, создающий объект в таблице и возвращающий его адрес, используя API (метод POST).

```
static async Task<Uri> CreateКлассAsync(Класс объект)
{
    HttpResponseMessage response = await client.PostAsJsonAsync("api/Контроллер",
                объект);
   response.EnsureSuccessStatusCode();
   return response.Headers.Location;
}
```

Создать в клиентском приложении объект класса, вызвать созданный асинхронный метод, передав в параметрах созданный объект.

Вывести на экран адрес страницы созданного объекта (если запись добавлена) и сообщение, удалось ли добавить запись.

Проверить, что в БД добавлена новая строка с данными, полученными из клиентского приложения.

5.5 Создать асинхронный метод, изменяющий переданный в параметрах объект в таблице по его идентификатору, используя API (метод PUT).

```
static async Task<Класс> UpdateКлассAsync(Класс объект)
{
    HttpResponseMessage response = await client.PutAsJsonAsync($"api/Контроллер/
                {объект.Id}", объект);
   response.EnsureSuccessStatusCode();
   объект = await response.Content.ReadAsAsync<Класс>();
   return объект;
}
```

Для тестирования:

- получить объект по id,
- изменить в клиентском приложении значение одного из свойств полученного объекта класса,
- вызвать созданный асинхронный метод, передав в параметрах созданный объект.

Вывести на экран данные обновленного объекта и сообщение, удалось ли изменить запись.

Проверить, что в БД данные строки изменены на данные, полученные из клиентского приложения.

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать консольное приложение C#.

6.2 Выполнить все задания из п.5 в одном решении.

6.3 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 указать у объекта HttpClient ссылку на веб-страницу?

8.2 Как указать, что объект HttpClient принимает данные в формате JSON?

8.3 Для чего в объекте HttpClient используется метод GetAsync?

8.4 Для чего в объекте HttpClient используется метод DeleteAsync?

8.5 Для чего в объекте HttpClient используется метод PostAsync?

8.6 Для чего в объекте HttpClient используется метод PutAsync?

Лабораторная работа №30
Работа с буфером экрана

1 Цель работы

1.1 Научиться настраивать интерфейс консольных приложений на C#;

1.2 Научиться применять параметры командной строки в приложениях на C#

2 Литература

2.1 Фленов, М. Е. Библия C#. 4 изд / М. Е. Фленов. – Санкт-Петербург: БХВ-Петербург, 2019. – 512 с. – URL: https://ibooks.ru/bookshelf/366634/reading. – Режим доступа: только для зарегистрированных 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 Применение параметров командной строки (параметр args метода Main)

Создать консольное приложение.

В случае, если в параметрах командной строки после названия приложения указан один из следующих параметров: /? или /h или /help (т.е. args[0] равен одному из этих значений), вывести на экран фамилии разработчиков.

Если значение первого параметра -view (без учета регистра), то вывести на экран все параметры, переданные при запуске приложения из командной строки, кроме параметра args[0]. Каждый параметр – на отдельной строке.

Для тестирования запустить приложение с командной строки с параметрами и без них:

cd путь к приложению

название приложения параметр1 параметр2 параметр3 …

5.2 Создание меню в консольном приложении

Сменить заголовок приложения на «МДК.01.04».

Реализовать в цикле (или бесконечном цикле) вывод на экран названия следующих пунктов меню: «Запись файла», «Чтение файла», «Выход».

После вывода пунктов на экран в цикле ожидается указание пользователем требуемого пункта меню.

При выборе пункта меню «Выход» приложение должно закрываться.

При выборе пункта меню «Запись файла» должен вызываться метод InputText, при выборе пункта «Чтение файла» – метод ReadFile. Оба метода типа void, вместо реализации методов – Console.ReadKey();

После указания пользователем пунктов меню «Запись файла» или «Чтение файла» заголовок формы должен меняться на название выбранного пункта меню.

После завершения работы методов InputText и ReadFile должен выполняться возврат к меню.

5.3 Настройка внешнего вида консольного приложения

Написать и протестировать метод, очищающий содержимое консоли, изменяющий заголовок консоли, цвет фона и текста в консольном приложении. Метод должен вызываться при возврате в меню и при выборе каждого пункта меню кроме «Выход». Параметры метода: заголовок, цвет фона, цвет текста.

5.4 Перенаправление текстового потока в файл

Написать и протестировать метод, позволяющий записывать текст с консоли в файл, перенаправив стандартный потоковый вывод. Исходный поток требуется предварительно сохранить, а после сохранения файла — восстановить.

Имя файла должно быть введено пользователем. Метод должен вызываться при выборе пункта меню «Запись файла».

Признак конца ввода – строка end (она в файл записываться не должна).

5.5 Запуск консоли вместо оконного приложения

Создать оконное приложение, при запуске которого с аргументом командной строки -console программа запускается без графического интерфейса (в консольном режиме).

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать консольное приложение C#.

6.2 Выполнить все задания из п.5 в одном решении.

6.3 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 изменить цвет фона консольного приложения?

8.2 Как изменить цвет текста консольного приложения?

8.3 Как изменить стандартный потоковый ввод, перенаправив его на запись в файл?

8.4 Как перехватить нажатие произвольной клавиши в консольном приложении C#?

8.5 Как изменить заголовок окна в консольном приложении C#?