

# Лабораторная работа №6

## Разработка консольных приложений

### 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 <https://metanit.com/sharp/tutorial> – 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 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.01.02».

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).

Для реализации:

1) объявить в классе `Program` поля:

```
[DllImport("kernel32.dll", EntryPoint = "GetStdHandle",
SetLastError = true, CharSet = CharSet.Auto, CallingConvention =
CallingConvention.StdCall)]
public static extern IntPtr GetStdHandle(int nStdHandle);
```

```
[DllImport("kernel32.dll", EntryPoint = "AllocConsole",
SetLastError = true, CharSet = CharSet.Auto, CallingConvention =
CallingConvention.StdCall)]
public static extern int AllocConsole();
```

2) добавить в `Main` аргументы командной строки

3) проверять, если есть аргумент `-console`, то запускать консольное приложение вместо вывода формы на экран:

```
AllocConsole();
Console.WriteLine("test");
Console.ReadLine();
```

## 6 Порядок выполнения работы

6.1 Выполнить все задания из п.3 в одном решении `LabWork6`. Возможные ошибки требуется обрабатывать. Выполнить форматирование и рефакторинг кода.

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

# 8Контрольные 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#?

# 9Приложение

## Настройка окна

```
Console.SetWindowPosition(x, y); // установка положения окна
Console.SetWindowSize(x,y); // установка размеров окна
Console.Title = "text"; // смена заголовка окна
```

## Очистка экрана:

```
Console.Clear();
```

## Смена цветов в консольном приложении:

```
Console.ForegroundColor = ConsoleColor.НазваниеЦвета; // цвет шрифта
Console.BackgroundColor = ConsoleColor.НазваниеЦвета; // цвет фона текста
Console.BackgroundColor = ConsoleColor.НазваниеЦвета; Console.Clear(); // цвет фона окна
Console.ResetColor(); // сброс цветов на стандартные
```

## Перенаправление текста из консоли (например, в файл)

```
// Для работы с потоками в разделе using подключить System.IO.
StreamWriter writer = new StreamWriter(имя файла); // создание потока для записи в файл
writer.AutoFlush = true; // данные будут сброшены из буфера в поток
Console.SetOut(writer); // установка потокового вывода консоли на поток writer
Console.Out.WriteLine(текст); // направление текста в потоковый вывод
Console.Out.Close(); // закрытие передачи потокового вывода
```

## Звуки:

```
// звуковой сигнал
Console.Beep();
// проигрывание звука с указанной частотой указанное количество миллисекунд
Console.Beep(частота звука, длительность);
```

## Положение курсора:

```
Console.SetCursorPosition(x, y); // установка положения курсора в координату (x, y) окна
Console.CursorTop; // верхняя граница окна
Console.CursorLeft; // левая граница окна
```

## Получение информации о нажатой клавише:

```
// info - объект "Информация о нажатой клавише"
ConsoleKeyInfo info = Console.ReadKey();
// проверка, нажат ли ALT одновременно с текущей нажатой клавишей
(info.Modifiers & ConsoleModifiers.Alt) != 0
// проверка, что была нажата стрелка вверх
info.Key == ConsoleKey.UpArrow;
```