09.12.20г\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

19ИСиП 2Д \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Основы алгоритмизации и программирования

ТЕМА Решение задач, создание программы «SharpDeveoper»

СОЗДАНИЯ ПРОГРАММЫ

Изначально создание программы можно разделить на следующие этапы:

1. Разработка алгоритма работы программы.
2. Дизайн пользовательского интерфейса. Размеры и свойства окна программы, расположение элементов управление внутри окна (окошки, кнопочки, менюшки и другие) и так далее.
3. Написание логики программы. Это именно та часть, которая реагирует на события от нажатия кнопок, события операционной системы и выводит результат своей работы на экран.

[**Back to top**](http://profhelp.com.ua/articles/how-to-write-custom-application?page=17#top)

2 . УСТАНОВКА СРЕДЫ РАЗРАБОТКИ

Так как большинство из нас не гениальные мальчики и девочки, родившиеся с клавиатурой в руках и не впитали языки программирования с молоком матери, нам понадобится программа более функциональная и наглядная чем блокнот. Эта программа называется интегрированной средой разработки (IDE - Integrated Development Eniroenment). Таких сред разработки много - это [Microsoft Visual Studio](http://www.microsoft.com/visualstudio/), [SharpDevelop](http://www.icsharpcode.net/), [Netbeans](http://netbeans.org/), и другие. В данной статье мы будем писать программу под Windows, используя язык програмирования C#, поэтому остановим свой выбор на среде разработки [SharpDevelop](http://www.icsharpcode.net/), которая является бесплатным продуктом с открытым исходным кодом. [SharpDevelop](http://www.icsharpcode.net/) содержит в себе все средства, которые нужны для написания полноценной программы: Средство визуального проектирования пользовательского интерфейса, автодополнение кода и многое другое.

Устанавливать будем последнюю версию, которая доступна для [загрузки](http://www.icsharpcode.net/OpenSource/SD/Download/) на сайте разработчика.

[Lightbox](http://profhelp.com.ua/sites/default/files/how-to-write-program/sharp-develop-1.png)



Устанавливаем программу, думаю здесь не должно возникнуть сложностей.

После установки запускаем среду разработки.



[**Back to top**](http://profhelp.com.ua/articles/how-to-write-custom-application?page=17#top)

3 . СОЗДАНИЕ ПРОГРАММЫ

Нажимаем кнопку **New solution**, или меню **File - New - Solution**.

В открывшемся диалоговом окне выбираем: **C# - Windows Applications - Windows Application - Название програмы - Create (Создать)**



В закладке **Projects** открываем файл MainForm.Designer, как показано на рисунке.



В закладке **Tools** выбираемм элемент  управления **Button** (Кнопка), перетаскиваем ее на форму. Теперь у yас форма содержит кнопку.



Давайте ее переименуем, и назначим ей действие - при нажатии кнопки будет закрываться программа.

Переименовуем кнопку: В окне Properties находим свойство Text, Меняем значение **Button1** на **Close**



**НАЗНАЧАЕМ ОБРАБОТЧИК СОБЫТИЯ НАЖАТИЯ КНОПКИ**

Для этого делаем двойное нажатие на кнопке, программа автоматически создаст необходимую функцию **Button1Click**. Внутри этой функции пишем: **this.Close();**



Запускаем программу:



При нажатии кнопки **Close**, окно программы закроется, в след за этим завершит работу и сама программа.

**СДЕЛАЕМ ЧТО-НИБУДЬ ПОСЛОЖНЕЕ, ЧЕМ ПРОСТО ВЫВОД ТЕКСТА НА ЭКРАН**

В большинстве учебников для начинающих программистов первая программа выводит простое сообщение "Hello world!". Мне кажется это немного тупо, так как для того, чтоб заинтересовать начинающего программиста, нужно показать нечто большее и полезное, нежели просто окно с текстом.

Для того чтоб оценить возможности языка программирования C# и .Net Framework, давайте в нашей программе реализуем возможность выполнения простейших арифметических операций. Мне кажется это будет гораздо полезней.

Итак, давайте создадим следующие элементы интерфейса программы

* Ччетыре кнопки: "+", "-", "\*", "/"
* Три текстовых поля: два для выполнения арифметических операций и один для результата.
* Надписи (Labels)
* Кнопка закрытия окна осталась из примера, рассмотренного ранее

Добавлять элементы интерфейса программы весьма просто - перетягиваем их из панели инструментов в то место окна, где должен находиться элемент.

**Обратите внимание**, что имя элементов назначаются автоматически и, если не давать им осмысленные имена, и при достаточно боьшом их количестве, очень легко потеряться в хаосе, состоящем из элементов с названиями button1, button2, button3, textBox1, textBox2, textBox3 и так далее. Сразу после создания элемента, не ленитесь, **давайте ему осмысленое имя** - дальше, при создании обработчиков событий, **имена функций будут генерироваться автоматически на основании имени элемента управления**. Думаю никто не сомневается в том, что имя функции **ButtonAddClick()**гораздо более осмысленно, чем **Button1Click()**

**НАЗНАЧИМ ОБРАБОТЧИКИ СОБЫТИЙ НАЖАТИЯ КНОПОК**

Выбераем по очереди каждую кнопку. Даблкликом создаем обработчик нажатия кнопки(Click). Если нужно добавить обработчик другого события - следует открыть вкладку События (значек в виде молнии)  и там уже выбрать необходимое событие.

Пишем код обработчика

Приведу текст для операции сложения. По аналогии я думаю вы догадаетесь как сделать вычитание и другие виды действий с числами, в плоть до возведения в степень и так далее:

[?](http://profhelp.com.ua/articles/how-to-write-custom-application?page=17)

|  |  |
| --- | --- |
| 123456 | void ButtonAdditionClick(object sender, EventArgs e){    textBoxResult.Text =        (Convert.ToInt32(textBoxOperand1.Text.ToString()) +        Convert.ToInt32(textBoxOperand2.Text.ToString())).ToString();} |

**Проверяем работу программы**

Контрольные вопросы :

1 На каком язые идет написание кода?

2 Как называется программа для создания програмы?

3 Из каких частей состоит составление программы?

Преподователь \_\_\_\_\_\_\_\_\_\_\_\_\_\_Дузаев И.К.