

Теория и практика программирования

Шпилёв Пётр Валерьевич

Санкт-Петербургский государственный университет
Математико-механический факультет
Кафедра статистического моделирования

Лекция 1

Санкт-Петербург
2015 г.



Б. Пахомов. C-C++ и MS Visual C++ 2008 для начинающих, СПб.:БХВ-Петербург, 2009.



Герберт Шилдт. C# 4.0. Полное руководство, М.: ООО "И.Д. Вильямс 2011



Н. Культин. Microsoft Visual C в задачах и примерах, СПб.:БХВ-Петербург, 2009.



Ч. Петцольд. Программирование с использованием MS WindowsForms, М.: Русская Редакция; СПб.: Питер, 2006.



Ч. Петцольд. Программирование для Microsoft Windows на C#., М.: Русская Редакция, 2002



Эндрю Троелсен. Язык программирования C# 2010 и платформа .NET 4, М.:ООО "И.Д. Вильямс 2011.



Симон Робинсон и др. C# для профессионалов, издательство «Лори», 2003

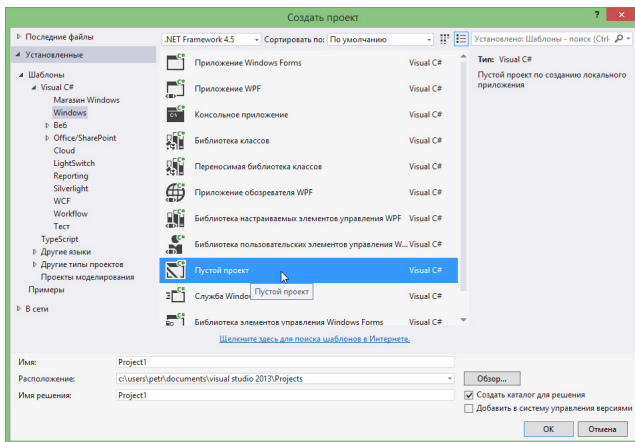
- Предыстория. Создание C#.
- Объектно-ориентированное программирование.
- Примеры.

- Предыстория. Создание C#.
- Объектно-ориентированное программирование.
- Примеры.

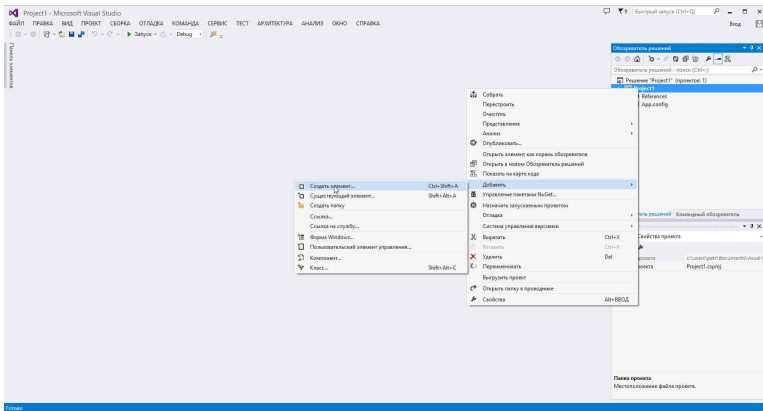
- Предыстория. Создание C#.
- Объектно-ориентированное программирование.
- Примеры.

```
/* Это простая программа на C#.  
Назовем ее Example.cs. */  
using System;  
class Example  
{  
    // Любая программа на C# начинается с вызова метода Main().  
    static void Main()  
    {  
        Console.WriteLine("Простая программа на C#.");  
    }  
}
```

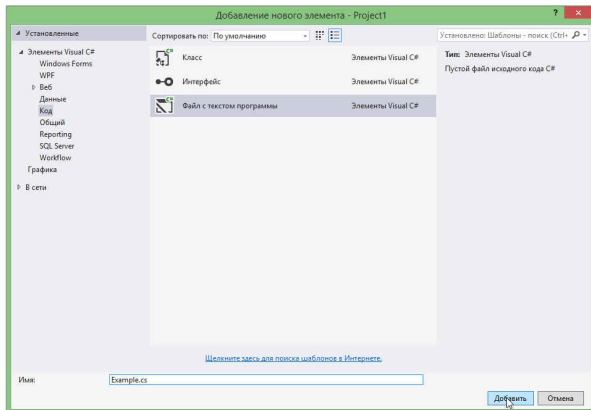
- Создайте новый (пустой) проект C#, выбрав команду Файл>Создать>Проект (File>New>Project). Затем выберите элемент Windows из списка Установленные шаблоны (Installed Templates) и далее — шаблон Пустой проект (Empty Project), как показано на рисунке. Щелкните на кнопке ОК, чтобы создать проект.



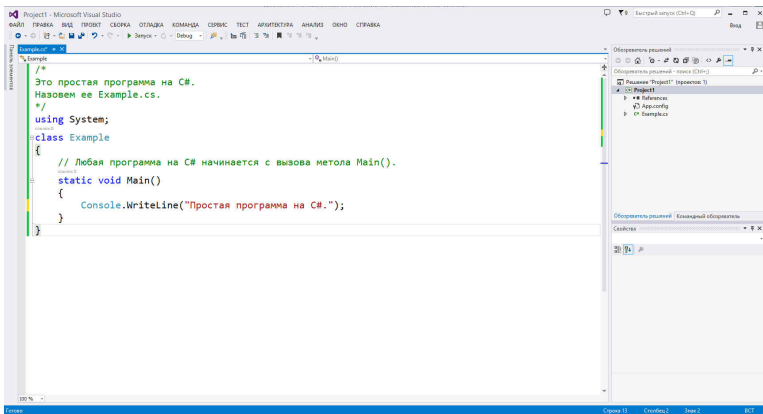
- На данном этапе проект пуст, и вам нужно ввести в него файл с исходным текстом программы на C#. Для этого щелкните правой кнопкой мыши на имени проекта (в данном случае — Project1) в окне Обозреватель решений, а затем выберите команду Добавить (Add) из контекстного меню. В итоге появится подменю, показанное на рисунке.



- Выберите команду Создать элемент (New Item), чтобы открыть диалоговое окно Добавление нового элемента (Add New Item). Выберите сначала элемент Код (Code) из списка Установленные шаблоны, а затем шаблон Файл с текстом программы (Code File) и измените имя файла на Example.cs, как показано на рисунке.

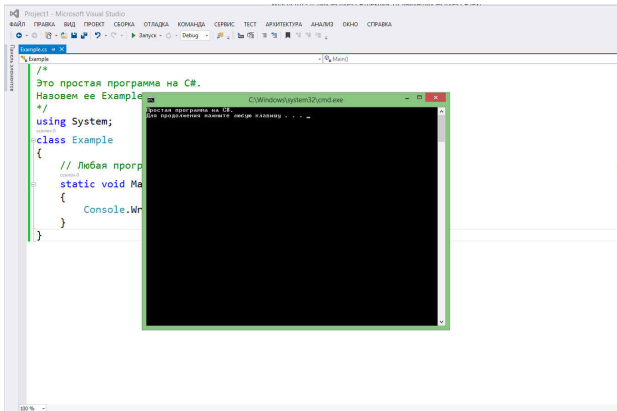


- Введите исходный текст программы в окне с меткой Example.cs, после чего сохраните этот текст в файле. По завершении ввода исходного текста программы экран будет выглядеть так, как показано на рисунке.



Лекция 1. Выполнение программы

- Скомпилируйте программу, выбрав команду Построение>Построить решение (Build>Build Solution).
- Выполните программу, выбрав команду Отладка>Запуск без отладки (Debug>Start Without Debugging). В результате выполнения программы откроется окно, показанное на рисунке.



- `/* Это простая программа на C#.
Назовем ее Example.cs. */`
- `using System;`
- `class Example`
- `// Любая программа на C# начинается с вызова метода Main().`
- `static void Main()`
- `Console.WriteLine("Простая программа на C#.");`

- `/* Это простая программа на C#.
Назовем ее Example.cs. */`
- `using System;`
- `class Example`
- `// Любая программа на C# начинается с вызова метода Main().`
- `static void Main()`
- `Console.WriteLine("Простая программа на C#.");`

- `/* Это простая программа на C#.
Назовем ее Example.cs. */`
- `using System;`
- `class Example`
- `// Любая программа на C# начинается с вызова метода Main().`
- `static void Main()`
- `Console.WriteLine("Простая программа на C#.");`

- `/* Это простая программа на C#.
Назовем ее Example.cs. */`
- `using System;`
- `class Example`
- `// Любая программа на C# начинается с вызова метода Main().`
- `static void Main()`
- `Console.WriteLine("Простая программа на C#.");`

- `/* Это простая программа на C#.
Назовем ее Example.cs. */`
- `using System;`
- `class Example`
- `// Любая программа на C# начинается с вызова метода Main().`
- `static void Main()`
- `Console.WriteLine("Простая программа на C#.");`

- `/* Это простая программа на C#.
Назовем ее Example.cs. */`
- `using System;`
- `class Example`
- `// Любая программа на C# начинается с вызова метода Main().`
- `static void Main()`
- `Console.WriteLine("Простая программа на C#.");`

```
// Эта программа демонстрирует применение переменных.  
using System;  
class Example2  
{  
    static void Main()  
    {  
        int x; // здесь объявляется переменная  
        int y; // здесь объявляется еще одна переменная  
        x = 100; // здесь переменной x присваивается значение 100  
        Console.WriteLine("x содержит" + x);  
        y = x / 2;  
        Console.Write("y содержит x / 2:");  
        Console.WriteLine(y);  
    }  
}
```

Обычно для объявления переменной служит следующий оператор:

- **тип имя_переменной;**

где **тип** — это конкретный тип объявляемой переменной, а **имя_переменной** — имя самой переменной. Помимо типа `int`, в `C#` поддерживается ряд других типов данных.

Замечание

В `C#` внедрено средство, называемое неявно типизированной переменной. Неявно типизированными являются такие переменные, тип которых автоматически определяется компилятором.

```
/* Эта программа демонстрирует отличия между типами данных int и
double. */
using System;
class Example3
{
    static void Main()
    {
        int ivar; // объявить целочисленную переменную
        double dvar; // объявить переменную с плавающей точкой
        ivar = 100; // присвоить переменной ivar значение 100
        dvar = 100.0; // присвоить переменной dvar значение 100.0
        Console.WriteLine("Исходное значение ivar:" + ivar);
        Console.WriteLine("Исходное значение dvar:" + dvar);
        Console.WriteLine(); // вывести пустую строку
        // Разделить значения обеих переменных на 3.
        ivar = ivar / 3;
        dvar = dvar / 3.0;
        Console.WriteLine("Значение ivar после деления:" + ivar);
        Console.WriteLine("Значение dvar после деления:" + dvar);
    }
}
```

- Условный оператор: *if(условие) оператор*;
- Оператор цикла: *for(инициализация; условие; итерация) оператор*;
Например: `for(int i = 0; i < 3; i++) {...}`
- Тернарный оператор `"?:"`: *Выражение 1 ? Выражение 2 : Выражение 3*;
используется вместо определенных видов конструкций *if-then-else*
Например: `abs_val = val < 0 ? -val : val; // абсолютное значение переменной val`

Замечание

В C# точка с запятой обозначает конец оператора. Составные элементы оператора можно располагать в отдельных строках. Например, следующий фрагмент кода считается в C# вполне допустимым:

```
Console.WriteLine("Это длинная строка вывода" +  
x + y + z +  
"дополнительный вывод");
```

- Условный оператор: *if*(условие) оператор;
- Оператор цикла: *for*(инициализация; условие; итерация) оператор;
Например: `for(int i = 0; i < 3; i++) {...}`
- Тернарный оператор `"?:"`: *Выражение 1 ? Выражение 2 : Выражение 3*;
используется вместо определенных видов конструкций *if-then-else*
Например: `abs_val = val < 0 ? -val : val; // абсолютное значение`
`переменной val`

Замечание

В C# точка с запятой обозначает конец оператора. Составные элементы оператора можно располагать в отдельных строках. Например, следующий фрагмент кода считается в C# вполне допустимым:

```
Console.WriteLine("Это длинная строка вывода" +  
x + y + z +  
"дополнительный вывод");
```

- Условный оператор: *if*(условие) оператор;
- Оператор цикла: *for*(инициализация; условие; итерация) оператор;
Например: `for(int i = 0; i < 3; i++) {...}`
- Тернарный оператор `"?:"`: *Выражение 1 ? Выражение 2 : Выражение 3*;
используется вместо определенных видов конструкций *if-then-else*
Например: `abs_val = val < 0 ? -val : val; // абсолютное значение
переменной val`

Замечание

В C# точка с запятой обозначает конец оператора. Составные элементы оператора можно располагать в отдельных строках. Например, следующий фрагмент кода считается в C# вполне допустимым:

```
Console.WriteLine("Это длинная строка вывода" +  
x + y + z +  
"дополнительный вывод");
```

Таблица 1: Ключевые слова, зарезервированные в языке C#

abstract	as	base	bool	break
byte	case	catch	char	checked
class	const	continue	decimal	default
delegate	do	double	else	enum
event	explicit	extern	false	finally
fixed	float	for	foreach	goto
if	implicit	in	int	interface
internal	is	lock	long	namespace
new	null	object	operator	out
override	params	private	protected	public
readonly	ref	return	sbyte	sealed
short	sizeof	stackalloc	static	string
struct	switch	this	throw	true
try	typeof	uint	ulong	unchecked
unsafe	ushort	using	virtual	volatile
void	while			

Таблица 2: Контекстные ключевые слова C#

add	dynamic	from	get	global
group	into	join	let	orderby
partial	remove	select	set	value
var	where	yield		

Замечание

Несмотря на то что зарезервированные ключевые слова нельзя использовать в качестве идентификаторов, в C# разрешается применять ключевое слово с предшествующим знаком @ в качестве допустимого идентификатора. Например, @for — действительный идентификатор.

```
using System;
class IdTest
{
    static void Main()
    {
        int @if; // применение ключевого слова if
                // в качестве идентификатора
        for(@if = 0; @if < 10; @if++)
            Console.WriteLine("@if равно" + @if);
    }
}
```

Таблица 3: Типы значений в C#

Тип	Значение
bool	Логический, предоставляет два значения: "истина" или "ложь"
byte	8-разрядный целочисленный без знака
char	Символьный
decimal	Десятичный (для финансовых расчетов)
double	С плавающей точкой двойной точности
float	С плавающей точкой одинарной точности
int	Целочисленный
long	Длинный целочисленный
sbyte	8-разрядный целочисленный со знаком
short	Короткий целочисленный
uint	Целочисленный без знака
ulong	Длинный целочисленный без знака
ushort	Короткий целочисленный без знака

Замечание

Помимо простых типов, в C# определены еще три категории типов значений: перечисления, структуры и обнуляемые типы.

Таблица 4: Целочисленные типы

Тип	Разрядность в битах	Диапазон представления чисел
byte	8	0-255
sbyte	8	-128-127
short	16	-32 768-32 767
ushort	16	0-65 535
int	32	-2 147 483 648-2 147 483 647
uint	32	0-4 294 967 295
long	64	-9 223 372 036 854 775 808-9 223 372 036 854 775 807
ulong	64	0-18 446 744 073 709 551 615

Замечание

Тип char также является целочисленным.

- Тип `decimal` предназначен для применения в финансовых расчетах. Этот тип имеет разрядность 128 бит для представления числовых значений в пределах от $1E-28$ до $7,9E+28$.

// Использовать тип decimal для расчета скидки.

```
using System;
```

```
class UseDecimal
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        decimal price;
```

```
        decimal discount;
```

```
        decimal discounted_price;
```

```
        // Рассчитать цену со скидкой.
```

```
        price = 19.95m;
```

```
        discount = 0.15m; // норма скидки составляет 15%
```

```
        discounted_price = price - ( price * discount);
```

```
        Console.WriteLine("Цена со скидкой:" + discounted_price);
```

```
    }
```

```
}
```

Упражнение 1.1

Представьте себе, к примеру, что некий богатей желает облагодетельствовать деньгами кафедру статистического моделирования и намерен внести первоначальный взнос в конце года 1. Если процентная ставка равна 10% и если меценат намерен обеспечивать кафедре по 100 тыс. дол. (с учетом 5-% роста заработной платы) ежегодно на бессро́чную перспективу, напишите программу для расчета суммы, которую ему нужно отложить для этой цели сегодня.

Упражнение 1.1

Представьте себе, к примеру, что некий богатей желает облагодетельствовать деньгами кафедру статистического моделирования и намерен внести первоначальный взнос в конце года 1. Если процентная ставка равна 10% и если меценат намерен обеспечивать кафедре по 100 тыс. дол. (с учетом 5-% роста заработной платы) ежегодно на бессрочную перспективу, напишите программу для расчета суммы, которую ему нужно отложить для этой цели сегодня.

Упражнение 1.2

Дополните предыдущую программу возможностью расчета аннуитета (актива, который приносит фиксированный доход в течение конечного числа лет).

Упражнение 1.1

Представьте себе, к примеру, что некий богатей желает облагодетельствовать деньгами кафедру статистического моделирования и намерен внести первоначальный взнос в конце года 1. Если процентная ставка равна 10% и если меценат намерен обеспечивать кафедре по 100 тыс. дол. (с учетом 5-% роста заработной платы) ежегодно на бессрочную перспективу, напишите программу для расчета суммы, которую ему нужно отложить для этой цели сегодня.

Упражнение 1.2

Дополните предыдущую программу возможностью расчета аннуитета (актива, который приносит фиксированный доход в течение конечного числа лет).

Упражнение 1.3

Предположим, работники кафедры хотят получать зарплату от мецената не ежегодно, а ежемесячно ($1/12$ от 100 тыс с учетом 5% ежегодного роста) какова сумма в этом случае?

Упражнение 1.4

С помощью написанной программы выполните следующие расчеты:

- а) Новый автомобиль стоит 10 тыс. дол. Если процентная ставка равна 5%, сколько вы должны отложить сейчас, чтобы накопить эту сумму за пять лет?
- б) Вы должны платить за обучение по 12 тыс. дол. в конце каждого года в течение следующих шести лет. Если процентная ставка равна 8%, сколько вам нужно отложить сегодня, чтобы покрыть эту сумму?
- в) Вы инвестировали 60 476 дол. под 8% годовых. Сколько у вас останется к концу года 6 после внесения платы за обучение, о которой говорилось выше?