

Переменные и константы

Для хранения данных в программе применяются **переменные**. Переменная представляет именованную область памяти, в которой хранится значение определенного типа. Переменная имеет тип, имя и значение. Тип определяет, какого рода информацию может хранить переменная.

Перед использованием любую переменную надо определить. Синтаксис определения переменной выглядит следующим образом:

```
1  тип имя_переменной;
```

Вначале идет тип переменной, потом ее имя. В качестве имени переменной может выступать любое произвольное название, которое удовлетворяет следующим требованиям:

- имя может содержать любые цифры, буквы и символ подчеркивания, при этом первый символ в имени должен быть буквой или символом подчеркивания
- в имени не должно быть знаков пунктуации и пробелов
- имя не может быть ключевым словом языка C#. Таких слов не так много, и при работе в Visual Studio среда разработки подсвечивает ключевые слова синим цветом.

Хотя имя переменной может быть любым, но следует давать переменным описательные имена, которые будут говорить об их предназначении.

Например, определим простейшую переменную:

```
1  string name;
```

В данном случае определена переменная `name`, которая имеет тип **string**. то есть переменная представляет строку. Поскольку определение переменной представляет собой инструкцию, то после него ставится точка с запятой.

При этом следует учитывать, что C# является регистрозависимым языком, поэтому следующие два определения переменных будут представлять две разные переменные:

```
1  string name;  
2  string Name;
```

После определения переменной можно присвоить некоторое значение:

```
1  string name;  
2  name = "Tom";
```

Так как переменная `name` представляет тип `string`, то есть строку, то мы можем присвоить ей строку в двойных кавычках. Причем переменной можно присвоить только то значение, которое соответствует ее типу.

В дальнейшем с помощью имени переменной мы сможем обращаться к той области памяти, в которой хранится ее значение.

Также мы можем сразу при определении присвоить переменной значение. Данный прием называется инициализацией:

```
1 string name = "Tom";
```

Отличительной чертой переменных является то, что в программе можно многократно менять их значение. Например, создадим небольшую программу, в которой определим переменную, поменяем ее значение и выведем его на консоль:

```
1 string name = "Tom"; // определяем переменную и инициализируем ее
2
3 Console.WriteLine(name); // Tom
4
5 name = "Bob"; // меняем значение переменной
6 Console.WriteLine(name); // Bob
```

Консольный вывод программы:

```
Tom
Bob
```

Константы

Отличительной особенностью переменных является то, что мы можем изменить их значение в процессе работы программы. Но, кроме того, в C# есть константы. **Константа** должна быть обязательно инициализирована при определении, и после определения значение константы не может быть изменено

Константы предназначены для описания таких значений, которые не должны изменяться в программе. Для определения констант используется ключевое слово **const**, которое указывается перед типом константы:

```
1 const string NAME = "Tom"; // определяем константу
```

Так, в данном случае определена константа `NAME`, которая хранит строку "Tom". Нередко для название констант используется верхний регистр, но это не более чем условность.

При использовании констант надо помнить, что объявить мы их можем только один раз и что к моменту компиляции они должны быть определены. Так, в следующем случае мы получим ошибку, так как константе не присвоено начальное значение:

```
1 const string NAME; // ! Ошибка - константа NAME не инициализирована
```

Кроме того, мы ее не сможем изменить в процессе работы программы:

```
1  const string NAME = "Tom";    // определяем константу
2  NAME = "Bob";                // !Ошибка - у константы нельзя изменить значение
```

Таким образом, если нам надо хранить в программе некоторые данные, но их не следует изменить, они определяются в виде констант. Если же их можно изменять, то они определяются в виде переменных.