C# Starter

Переменные и типы данных.

Автор курса

Александр Шевчук
MCT

MCID: 9230440

После урока обязательно

Повторите этот урок в видео формате на ITVDN.com

Доступ можно получить через руководство вашего учебного центра

Проверьте как Вы усвоили данный материал на TestProvider.com

Тема

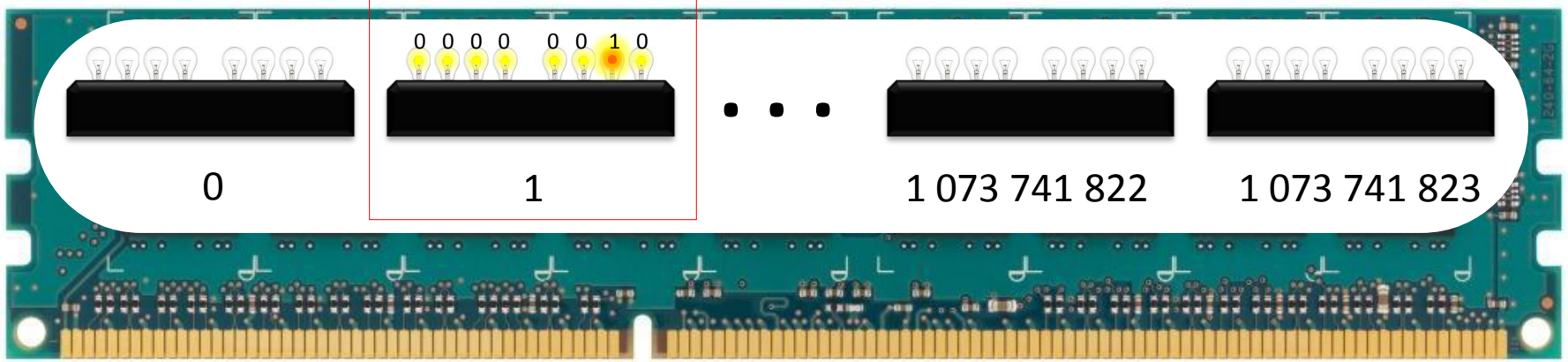
Переменные и типы данных.

Переменная

Variable

Переменная – это именованная область памяти, которая хранит в себе некоторое значение, которое можно изменить.

```
byte a = 2;  // 0000 0010 b
```

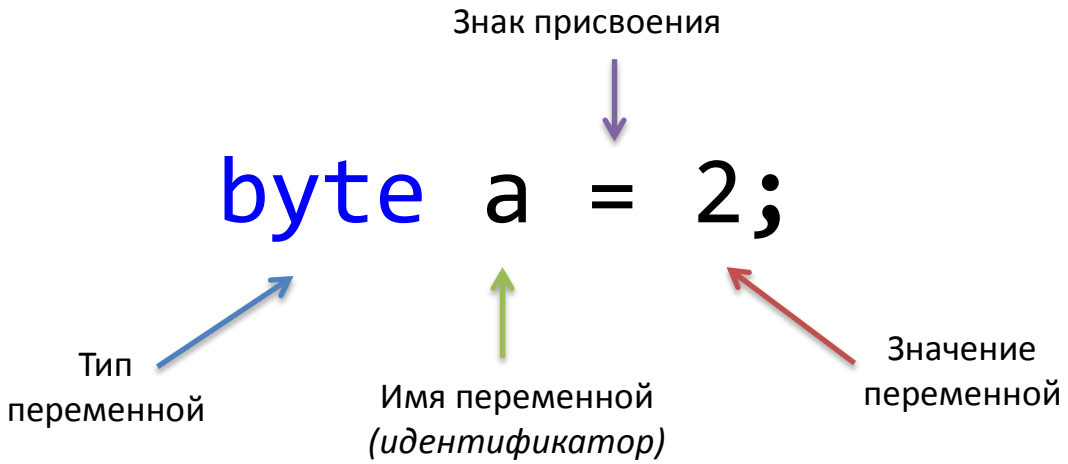


Переменная

Создание переменной

При создании переменной необходимо указать:

- Имя переменной *(идентификатор)*
- Тип переменной
- Начальное значение *(необязательно)*



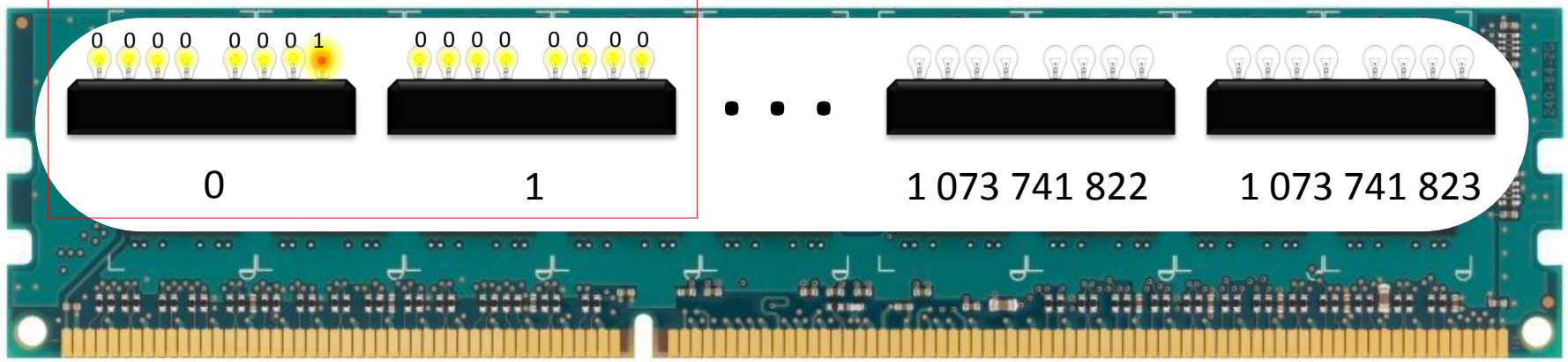
Инициализация переменной – это первое присвоение ей значения.

Переменная

Variable

Переменная – это именованная область памяти, которая хранит в себе некоторое значение, которое можно изменить.

```
short a = 256;          // 0000 0001  0000 0000 b - 0x100
```



Типы данных ОЗУ

Варианты хранения информации в ОЗУ

1 байт = 8 бит `byte` `sbyte` `bool`

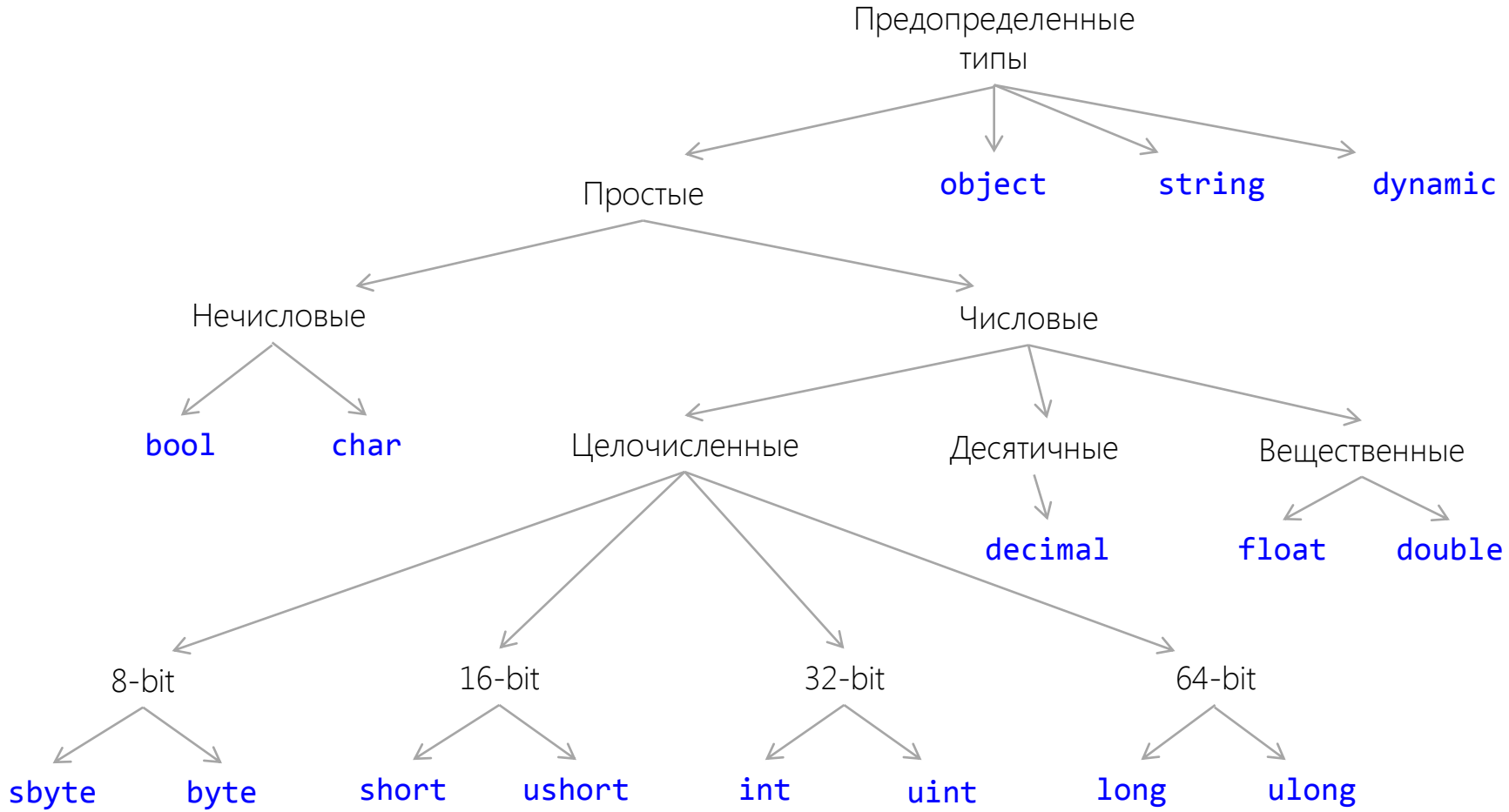
2 байта = 16 бит (Машинное слово) `short` `ushort` `char`

4 байта = 32 бита (Двойное машинное слово) `int` `uint` `float`

8 байт = 64 бита (Учетверённое машинное слово) `long` `ulong` `double`

Типы данных

Data type



Правила именования

Имена переменных должны быть понятны и передавать смысл хранимого значения.

```
int MyVariable = 2;
```

Первый символ идентификатора

Допустимые символы	*Недопустимые символы*
A-Z, a-z, _ , @	*0-9, . , , , ? , : , *,>,<,=,&,#,!,$ и т.п.*

Остальные символы идентификатора

Допустимые символы	*Недопустимые символы*
A-Z, a-z, _ , 0-9	*@, . , , , ? , : , *,>,<,=,&,#,!,$ и т.п.*

Идентификаторы (имена переменных) могут записываться с помощью символов Unicode, указанных с использованием синтаксиса \uXXXX, где XXXX – четырехзначный шестнадцатеричный код Unicode символа. Например: \u005fmyVar , где \u005f – Unicode-код для знака подчеркивания. Такой подход не рекомендован.

Переменные

Правила именования

1) В идентификаторах допустимо использовать символы алфавита и нижнего подчеркивания:

```
myVariable, my_Variable, _MyVariable
```

2) Использование цифр недопустимо только на первой позиции:

```
myVariable1, my1Variable, 1MyVariable
```

3) Нельзя использовать в качестве идентификаторов зарезервированные ключевые слова:

```
decimal, false, extern, intMyVar
```

4) Использование символа @ допустимо только на первой позиции:

```
@myVariable, my@Variable
```

5) Язык C# чувствителен к регистру, поэтому если вы напишите их в разном регистре – это будут различные переменные:

```
myVariable, MyVariable, myvariable
```

Ключевые слова

Keywords

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

Ключевые слова – это предварительно определенные зарезервированные идентификаторы, имеющие специальные значения для компилятора.

Ключевые слова нельзя использовать в программе в качестве идентификаторов, если только они не содержат префикс @.

Символ @, в идентификаторе переменной, указывает компилятору, что это – идентификатор, а не ключевое слово C# или его команда.

Ключевые слова не могут быть использованы в качестве идентификаторов.

MSDN: http://msdn.microsoft.com/ru-ru/library/x53a06bb.aspx

Контекстные ключевые слова

Contextual Keywords

Контекстные ключевые слова могут быть использованы в качестве идентификаторов.

`add`	`get`	`orderby`	`value`
`alias`	`global`	`partial` (type)	`var`
`ascending`	`group`	`partial` (method)	`where` (generic)
`descending`	`into`	`remove`	`where` (query)
`dynamic`	`join`	`select`	`yield`
`from`	`let`	`set`	

MSDN: http://msdn.microsoft.com/ru-ru/library/x53a06bb.aspx

Соглашения по именованию

Спецификация языка C # рекомендует придерживаться определенных правил (casing conventions) при создании идентификаторов.

Стиль	Описание	Пример
Pascal casing	каждое слово в идентификаторе начинается с большой буквы	MyMethod, Remove
Camel casing	каждое слово, исключая первое, в идентификаторе начинается с большой буквы	myCount, totalDiscount
Uppercase	идентификатор состоит из букв написанных в верхнем регистре	IO, XML

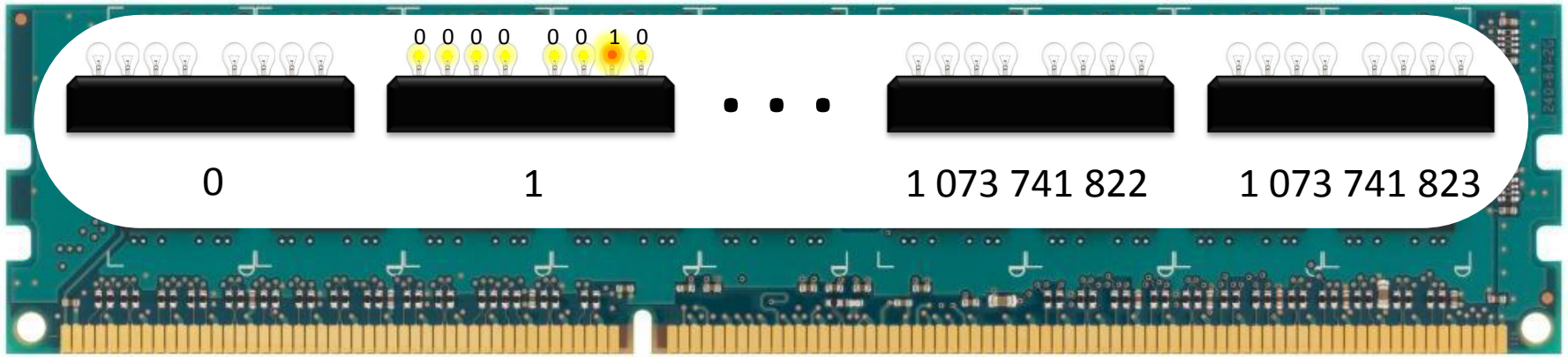
Не рекомендуется пользоваться венгерской нотацией и начинать идентификаторы с символа нижнего подчеркивания.

Константа

Constant

Константа – это именованная область памяти, которая хранит в себе некоторое значение, которое нельзя изменить.

```
const byte a = 2;  // 0000 0010 b
```



Попытка присвоить константе новое значение, приводит к ошибке уровня компиляции.

Преобразование значения типа

Casting или Type conversion

Кастинг – это преобразование значения переменной одного типа в значение другого типа.

Явный кастинг (explicit)

Преобразования выполняются только в случае явного указания (в круглых, скобках) типа, в который необходимо преобразовать.

Неявный кастинг (implicit)

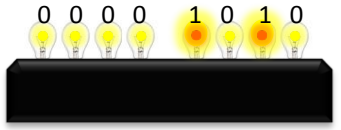
Преобразования выполняются автоматически без потери точности и урезания части исходного значения числа.

Неявный кастинг

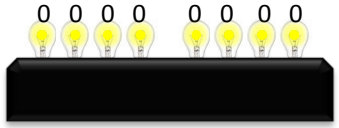
Безопасный

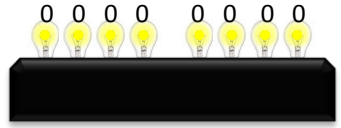
Неявное преобразование типа (безопасное) – преобразование значения меньшего типа в значение большего или целого в вещественное.

```
byte a = 10;
```

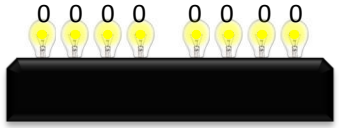


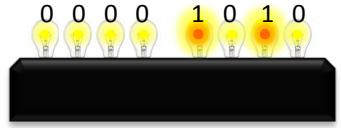

```
short b = 0;
```






```
b = a;
```





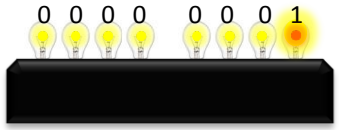
Ячейка заполняется нулевыми значениями или, если число было отрицательным – единицами.

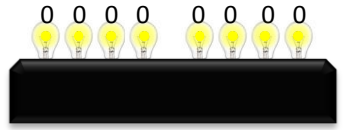
Явный кастинг

Опасный

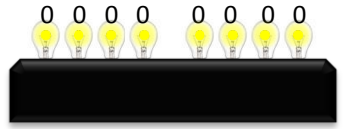
Явное преобразование типа (опасное) – преобразование значения большего типа в значение меньшего или вещественного в целое.

```
short a = 256;
```



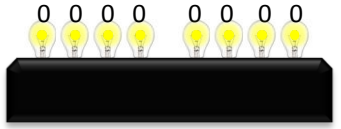



```
byte b = 0;
```




```
b = (byte)a;
```

Оператор явного преобразования значения типа.



Явный кастинг считается опасным, так как может произойти потеря точности или урезание числа.

Строки

Форматированный вывод

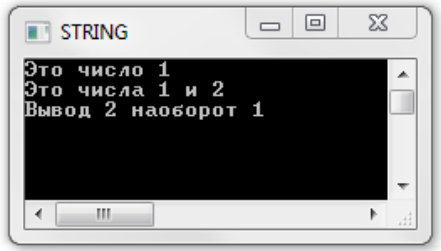
Маркер подстановки

```
Console.WriteLine("Это число {0}", 1);
```

Элемент подстановки

Позиция элемента подстановки, начинается с нуля (0,1,2,3,… и т.д.), указание большей позиции приведет к ошибке.

```
Console.WriteLine("Это числа {0} и {1}", 1, 2);
Console.WriteLine("Вывод {1} наоборот {0}", 1, 2);
```



Смотрите наши уроки в видео формате

ITVDN.com

Посмотрите этот урок в видео формате на образовательном портале ITVDN.com для закрепления пройденного материала.

Все курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics

ITVDN
IT VIDEO DEVELOPERS NETWORK

Проверка знаний

TestProvider.com

TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на TestProvider.com

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.

TestProvider

Q&A

Информационный видеосервис для разработчиков программного обеспечения

ITVDN

IT VIDEO DEVELOPERS NETWORK