Курс «C#. Программирование на языке высокого уровня»

Павловская Т.А.

Лекция 2. Состав языка и типы данных

Вводятся базовые для всего дальнейшего изложения понятия: из каких простейших «кирпичиков» состоят все тексты на языке программирования, что понимают под типом данных и какие встроенные типы данных есть в языке C#.

Состав языка

- **Символы**:
  - буквы: A-Z, a-z, _, буквы нац. алфавитов
  - цифры: 0-9, A-F
  - спец. символы: +, *, {, …
  - пробельные символы

- **Лексемы**:
  - константы 2 0.11 “Вася”
  - имена Vasia a _11
  - ключевые слова double do if
  - знаки операций + - =
  - разделители ; [ ] ,

- **Выражения**
  - выражение - правило вычисления значения: a + b
- **Операторы**
  - исполняемые: c = a + b;
  - описания: double a, b;

Константы (литералы) C#

Вид	Примеры
Булевские	*true* *false*
Целые дес.	8 199226 0Lu
шестн.	0xA 0x1B8 0X00FFL
Веществ. с тчк	5.7 .001f 35m
с порядком	0.2E6 .11e–3 5E10
Символьные	'A' '\x74' '\0' '\uA81B'
Строковые	"Здесь был Vasia"
	"\t3начение r=\xF5\n"
	"Здесь был \u0056\u0061"
	@"C:\temp\file1.txt"
Константа null	null

Имена (идентификаторы)

- имя должно начинаться с буквы или _;
- имя должно содержать только буквы, знак подчеркивания и цифры;
- прописные и строчные буквы различаются;
- длина имени практически не ограничена.
- имена не должны совпадать с ключевыми словами, однако допускается: @if, @float…
- в именах можно использовать управляющие последовательности Unicode

Примеры правильных имен:
Vasia, Вася, _13, \u00F2\u01DD, @while.

Примеры неправильных имен:
2late, Big gig, Б#г

Нотации

Понятные и согласованные между собой имена — основа хорошего стиля. Существует несколько *нотаций* — соглашений о правилах создания имен.

В C# для именования различных видов программных объектов чаще всего используются две нотации:

- *Нотация Паскаля* - каждое слово начинается с прописной буквы:
  - MaxLength, MyFuzzyShooshpanchik
- *Camel notation* - с прописной буквы начинается каждое слово, составляющее идентификатор, кроме первого:
  - maxLength, myFuzzyShooshpanchik

Ключевые слова, знаки операций, разделители

- *Ключевые слова* — идентификаторы, имеющие специальное значение для компилятора. Их можно использовать только в том смысле, в котором они определены.
  - Например, для оператора перехода определено слово goto.
- *Знак операции* — один или более символов, определяющих действие над операндами. Внутри знака операции пробелы не допускаются.
  - Например, сложение +, деление /, сложное присваивание %=.
- Операции делятся на *унарные* (с одним операндом), *бинарные* (с двумя) и тернарную (с тремя).

- *Разделители* используются для разделения или, наоборот, группирования элементов. Примеры разделителей: скобки, точка, запятая.

Ключевые слова C#

abstract as base bool break byte case catch
char checked class const continue decimal
default delegate do doubleelse enum event
explicit externfalse finally fixed float for
foreach goto if implicit in intinterface
internal is lock long namespacenewnull object
operator outoverride paramsprivate protected
publicreadonly refreturnsbyte sealedshort sizeof
stackalloc staticstringstructswitchthis throw
true trytypeofuint ulong uncheckedunsafe
ushortusing virtual void volatile while

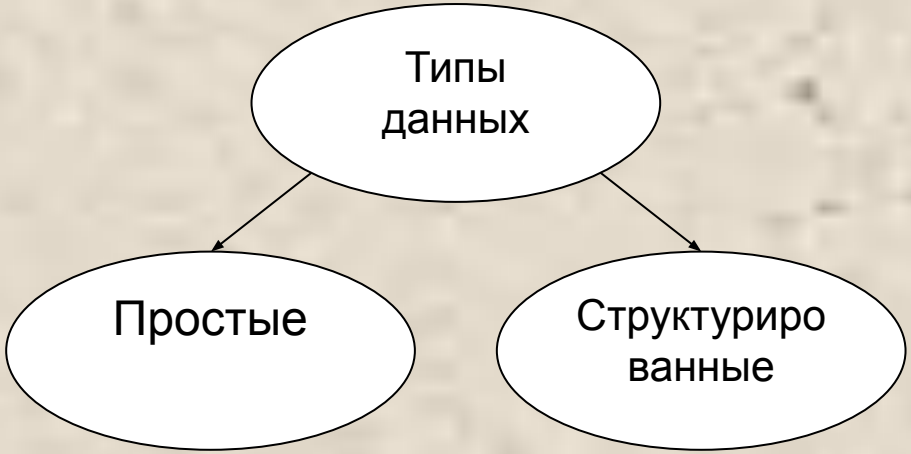
Типы данных

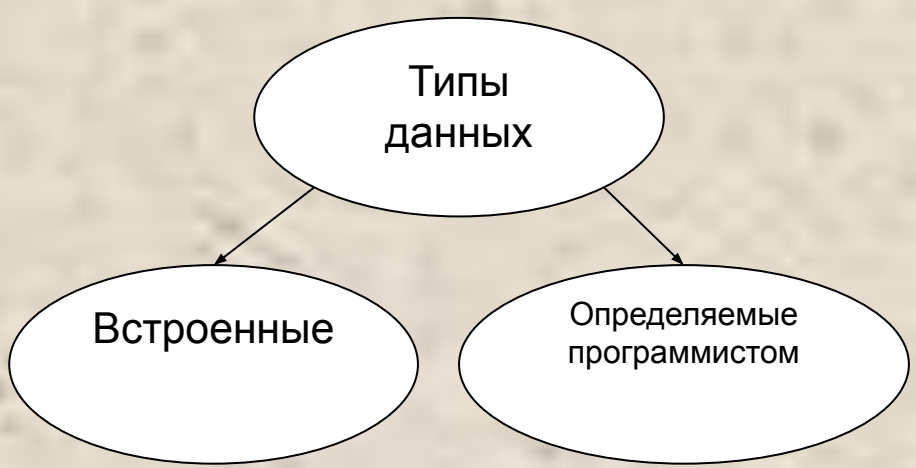
Концепция типа данных

Тип данных определяет:

- внутреннее представление данных => *множество их возможных значений*
- допустимые действия над данными => *операции и функции*

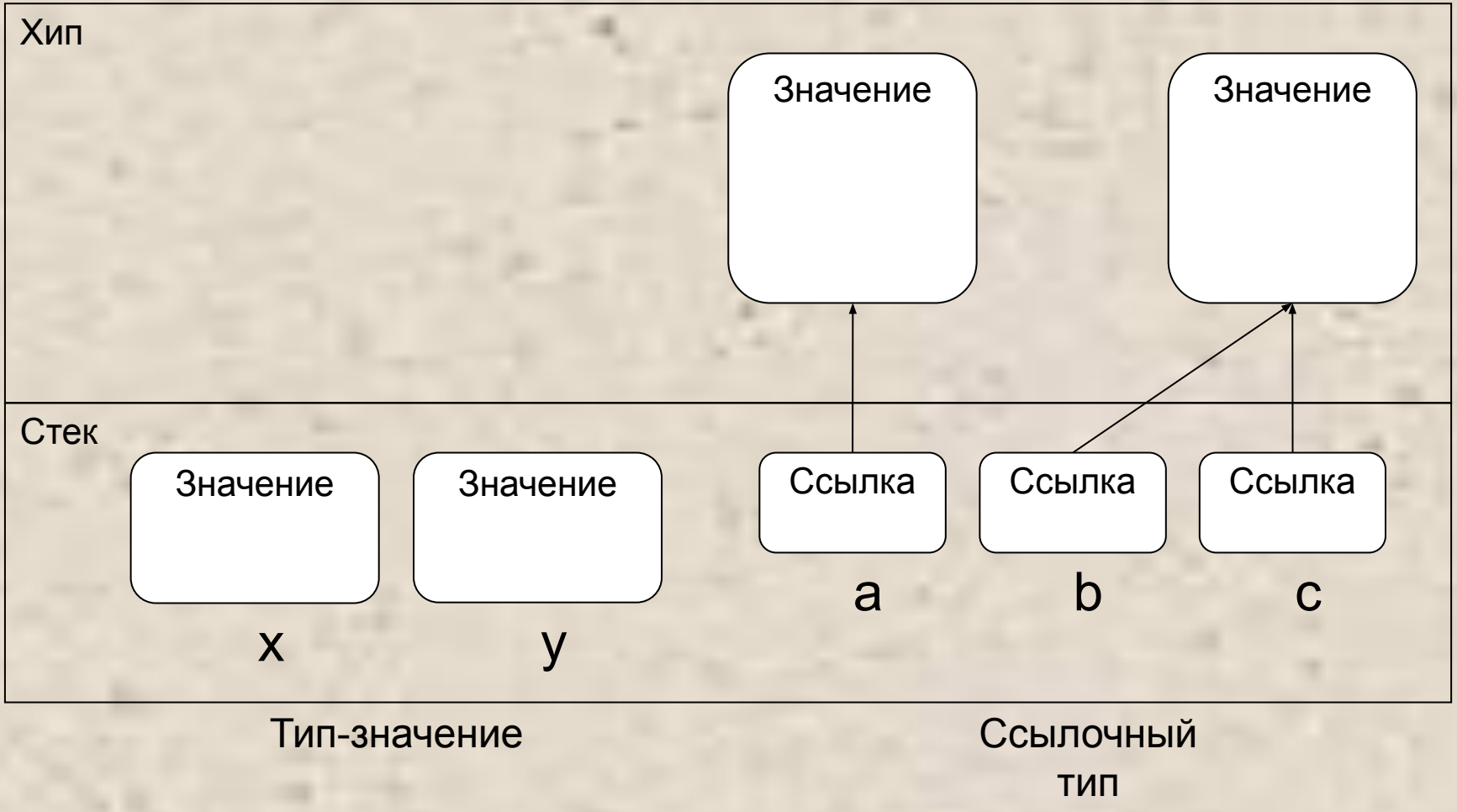
Различные классификации типов данных





Основная классификация типов C#

Хранение в памяти величин значимого и ссылочного типа



Встроенные типы данных C#

Логический и целые

Название	Ключевое слово	Тип .NET	Диапазон значений	Описание	Размер, бит
Булевский	bool	`Boolean`	`true, false`		
Целые	sbyte	`SByte`	–128 — 127	знаковое	8
	byte	`Byte`	0 — 255	беззнаковое	8
	short	`Int16`	–32768 —32767	знаковое	16
	ushort	`UInt16`	0 — 65535	беззнаковое	16
	int	`Int32`	$\approx(-210^9 — 210^9)$	знаковое	32
	uint	`UInt32`	$\approx(0 — 410^9)$	беззнаковое	32
	long	`Int64`	$\approx(-910^{18} — 910^{18})$	знаковое	64
	ulong	`UInt64`	$\approx(0— 1810^{18})$	беззнаковое	64

Остальные

Название	Ключевое слово	Тип .NET	Диапазон значений	Описание	Размер, бит
Символьный	char	`Char`	U+0000 — U+ffff	символ Unicode	16
Вещественные	float	`Single`	$1.5 \cdot 10^{-45}$ — $3.4 \cdot 10^{38}$	7 цифр	32
	double	`Double`	$5.0 \cdot 10^{-324}$ — $1.7 \cdot 10^{308}$	15-16 цифр	64
Финансовый	decimal	`Decimal`	$1.0 \cdot 10^{-28}$ — $7.9 \cdot 10^{28}$	28-29 цифр	128
Строковый	string	`String`	длина ограничена объемом доступной памяти	строка из символов Unicode	
object	object	`Object`	можно хранить все, что угодно	всеобщий предок	

Поля и методы встроенных типов

- Любой встроенный тип C# построен на основе стандартного класса библиотеки .NET. Это значит, что у встроенных типов данных C# есть *методы и поля*. С помощью них можно, например, получить:
  - double.MaxValue (или System.Double.MaxValue) — максимальное число типа double;
  - uint.MinValue (или System.UInt32.MinValue) — минимальное число типа uint.
- В вещественных классах есть элементы:
  - положительная бесконечность PositiveInfinity;
  - отрицательная бесконечность NegativeInfinity;
  - «не является числом»: NaN.