



ОНЛАЙН-ОБРАЗОВАНИЕ

## 04 – Java Data Types (Часть 1)

Дмитрий Коган  
Петрелевич Сергей



Как меня слышно и видно?



Если нет – напишите, если слышите – смайлик в чат.



Цели :

- Разберёмся с идентификаторами
- Узнаем разницу между примитивами и ссылочными типами данных
- Посмотрим на примеры



# Темы экзамена

- ☐ Java Basics
- ☐ **Working with Java Data Types**
- ☐ Using Operators and Decision Constructs
- ☐ Creating and Using Arrays
- ☐ Using Loop Constructs
- ☐ Working with Methods and Encapsulation
- ☐ Working with Inheritance
- ☐ Handling Exceptions
- ☐ Working with Selected classes from the Java API

# Подтемы экзамена

## Working with Java Data Types

- **Declare and initialize variables** (including casting of primitive data types)
- **Differentiate between object reference variables and primitive variables**
- Know how to read or write to object fields
- Explain an object's lifecycle (creation, "dereference by reassignment" and garbage collection)
- Develop code that uses wrapper classes such as Boolean, Double and Integer



Идентификаторы

# Идентификаторы

[Где про это почитать](#)

Что там написано

An identifier is an unlimited-length sequence of Java letters and Java digits, the first of which must be a Java letter

Некоторые примеры идентификаторов:

String  
i3  
αρετη  
MAX\_VALUE  
isLetterOrDigit

Всего три правила:

- Первый символ: буква, \$ или \_ .
- Может содержать цифры.
- Не может быть зарезервированным словом



# Идентификаторы

Properties of valid identifiers	Properties of invalid identifiers
<p>Unlimited length</p> <p>Starts with a letter ( a–z, upper- or lowercase), a currency sign, or an underscore</p> <p>Can use a digit (not at the starting position)</p> <p>Can use an underscore (at any position)</p> <p>Can use a currency sign (at any position): ¤, \$, £, ¢, ¥, and others</p>	<p>Same spelling as a Java reserved word or keyword (see table 2.8)</p> <p>Uses special characters: !, @, #, %, ^, &amp;, *, (, ), ', :, ;, [, /, \, }</p> <p>Starts with a Java digit (0–9)</p>
Examples of valid identifiers	Examples of invalid identifiers
<p>customerValueObject</p> <p>\$rate, £Value, _sine</p> <p>happy2Help, nullValue</p> <p>Constant</p>	<p>7world (identifier can't start with a digit)</p> <p>%value (identifier can't use special char %)</p> <p>Digital!, books@manning (identifier can't use special char ! or @)</p> <p>null, true, false, goto (identifier can't have the same name as a Java keyword or reserved word)</p>

# Ключевые слова

abstract	boolean	break	byte	case	catch
char	class	const	continue	default	do
double	else	extends	final	finally	float
for	goto	if	implements	import	instanceof
int	interface	long	native	new	package
private	protected	public	return	short	static
strictfp	super	switch	synchronized	this	throw
throws	transient	try	void	volatile	while
assert	enum				

# Примеры

Посмотрим примеры:

IdentifierExample.java

# Конвенции

- **Классы и интерфейсы**  
(существительные, с большой буквы):

Dog  
Account  
PrintWriter

- **Константы (final static):**

MIN\_HEIGHT

- **Методы**  
(глаголы, с маленькой буквы):

getBalance  
doCalculation  
setCustomerName

- **Переменные**  
(с маленькой буквы):

buttonWidth  
accountBalance  
myString



# Типы данных

# Какие бывают типы данных

[Где про это почитать](#)

Java – строго типизированный язык, т.е. типы переменных определены во время компиляции.

А в java 10, стало можно писать так:  
`var x = "ddd"`

Это динамическая типизация?

## Возможные типы данных

*Примитивные*

*Ссылочные*

# Примитивные типы

Целые типы (integral):

- byte 8-bit
- short 16-bit
- int 32-bit
- long 64-bit
- char 16-bit (без знаковый UTF-16)

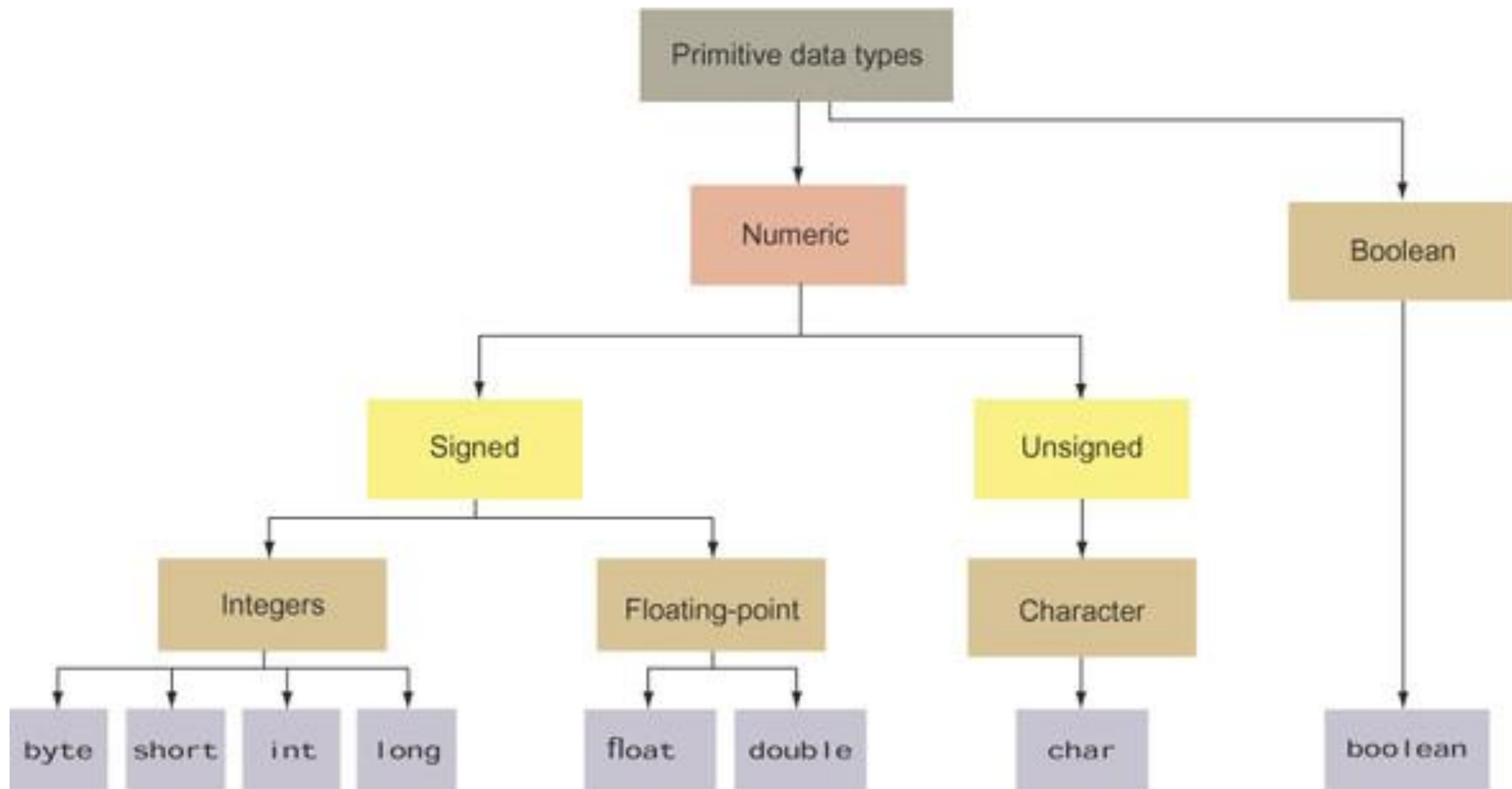
Типы с плавающей точкой (floating-point):

- float 32-bit
- double 64-bit

boolean – особый случай:

The boolean type has exactly two values: true and false.

# Примитивы





# Максимальное и минимальное значения

Значения находятся в константах, например Integer.MAX\_VALUE

byte:	-128....127
short:	-32_768...32_767;
int:	-2_147_483_648...2_147_483_647
long:	-9_223_372_036_854_775_808...9_223_372_036_854_775_807
float:	1.4e-45 ... 3.4028235e38
double:	4.9e-324 ... 1.7976931348623157e308
char:	\u0000 ... \uFFFF

Что такое “e+x” ?

Это  $10^x$

# Деньги

Какой тип данных возьмем для денег?

# Деньги

Самый плохой вариант: `double`

Возможный вариант: `long`

(помним, что может быть 0.01, а может 0.001)

Самый популярный вариант: `BigDecimal`

`Big*` типы хороши и для хранения очень-очень больших чисел.

# Значения по умолчанию

Тут все просто: или 0 или false.

# Пример

Посмотрим пример:

*Primitive.java*

# Важно помнить

Про примитивные типы важно помнить:

- смешивание дробных и целых чисел
- сравнение дробных чисел.
- переполнение

**Обязательно делайте тесты на эти случаи!**



Вопросы?



## Ссылочные типы

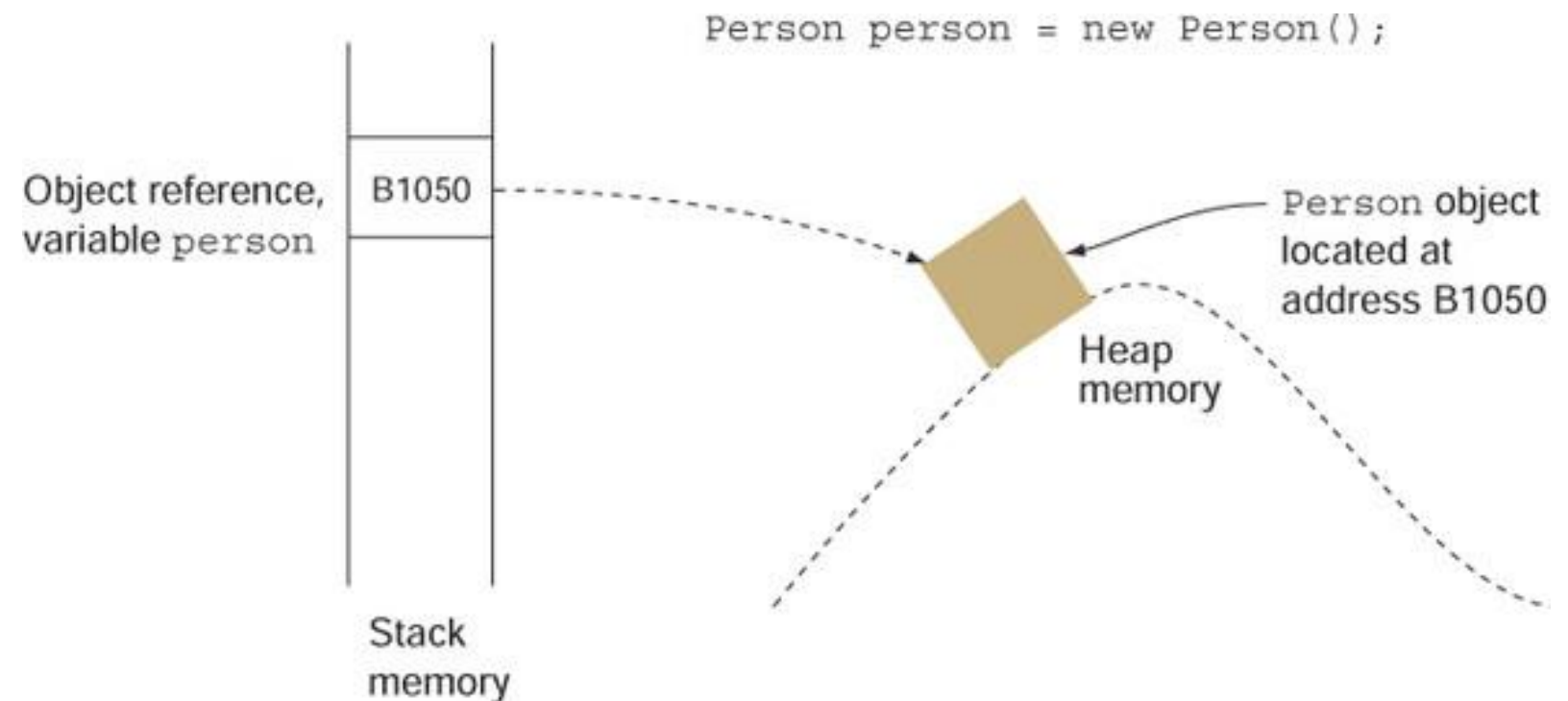
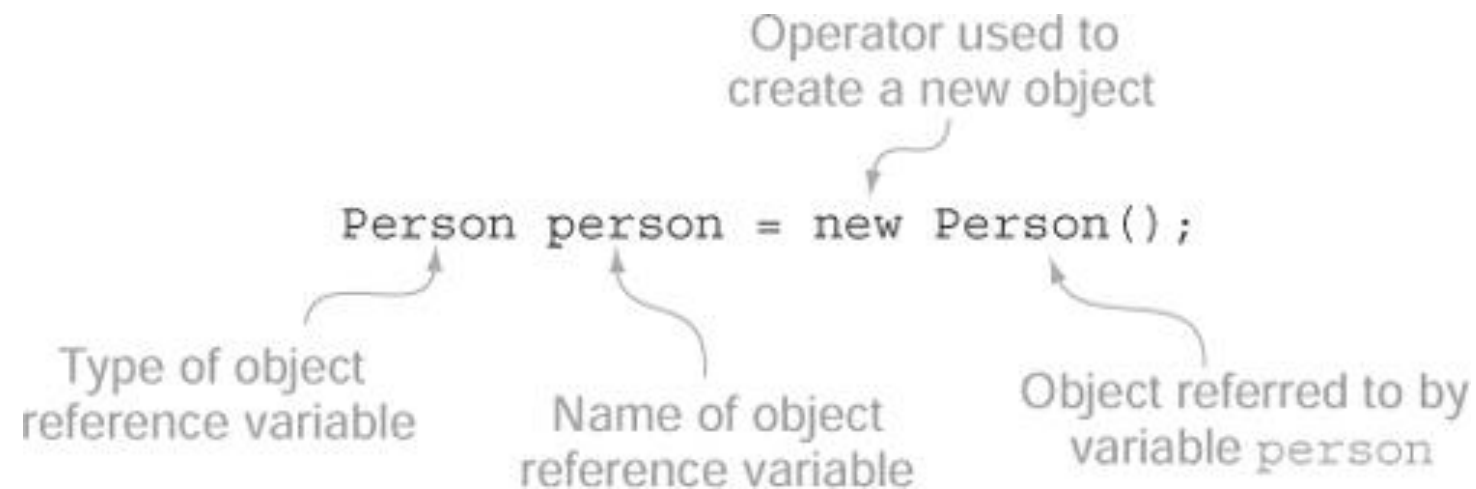


# Ссылочные типы

Выделяют следующие ссылочные типы:

- Class types
- Interface type
- Type variables
- Array types

# Расположение в памяти



# Ссылки и объекты



A leash without a dog.



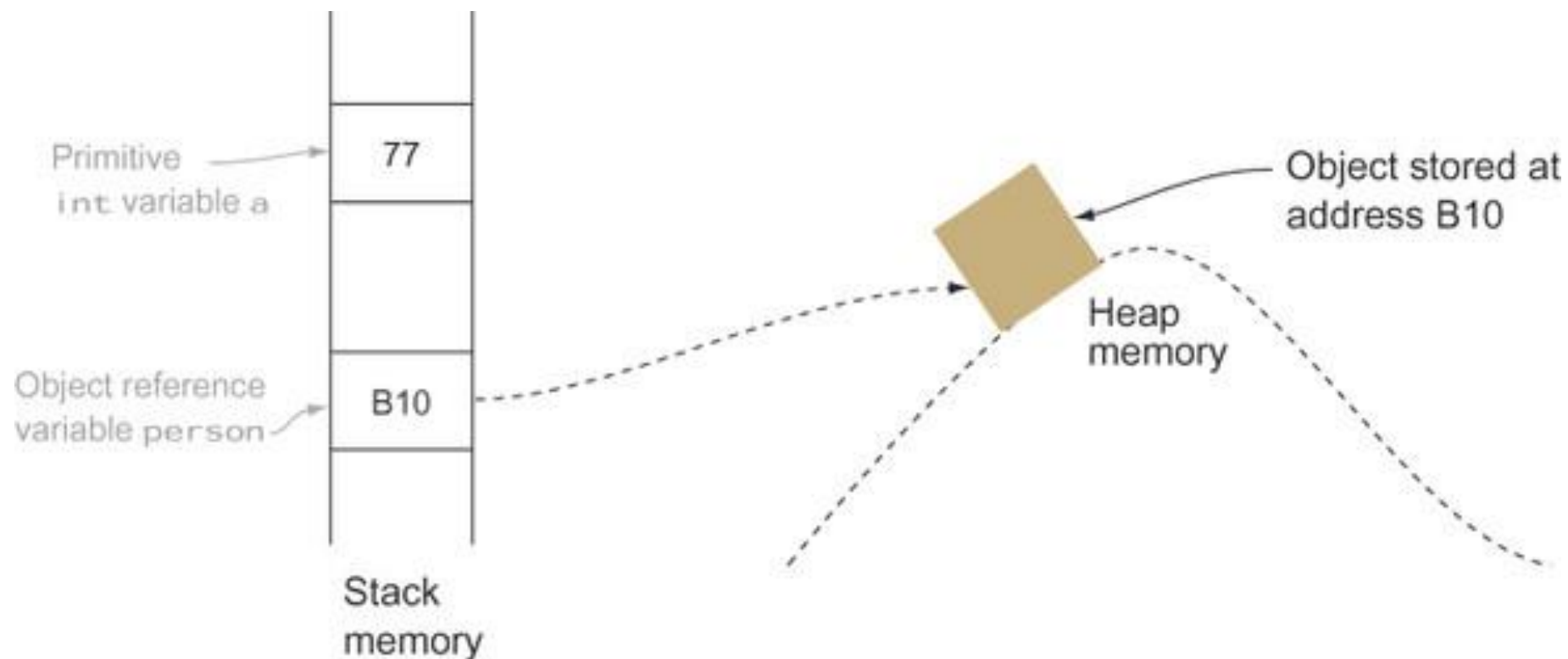
A dog without a leash.



Several leashes may be tethered to one dog.

# Отличия от примитивов

```
int a = 77;  
Person person = new Person();
```



# Boxing - unboxing

Чтобы примитивные типы данных можно было использовать в коллекциях, generic-ах т.п. существуют “boxed” классы.

boolean	Boolean
byte	Byte
short	Short
char	Character
int	Integer
long	Long
float	Float
double	Double

# Пример

Посмотрим примеры:

*Boxing.java*

*пакет bytes*

*GcDemo.java*



Вопросы?

---

**Домашнее задание**

Тест





Пожалуйста, пройдите опрос



Спасибо  
за внимание!

Никакого  
примитива в  
жизни! 😊