



Основы языка программирования Java

Компилируемые языки программирования

- **Компиляция** – преобразование текста программы, написанного на языке высокого уровня (C, C++, Pascal), в набор инструкций, которые может выполнять исполнитель
- Скомпилированная программа часто называется **байт-кодом**

Компилируемые языки программирования

Плюсы:

- Высокая скорость исполнения программы
- Отсутствие необходимости в дополнительном программном обеспечении для запуска программы

Минусы:

- Привязка к исполнителю

Компилируемые языки программирования



Интерпретируемые языки программирования

- **Интерпретация** – анализ текста программы, написанного на языке высокого уровня (JavaScript, PHP), и непосредственное исполнение обнаруженных инструкций
- Интерпретатор является исполнителем

Интерпретируемые языки программирования

Плюсы:

- Кроссплатформенность

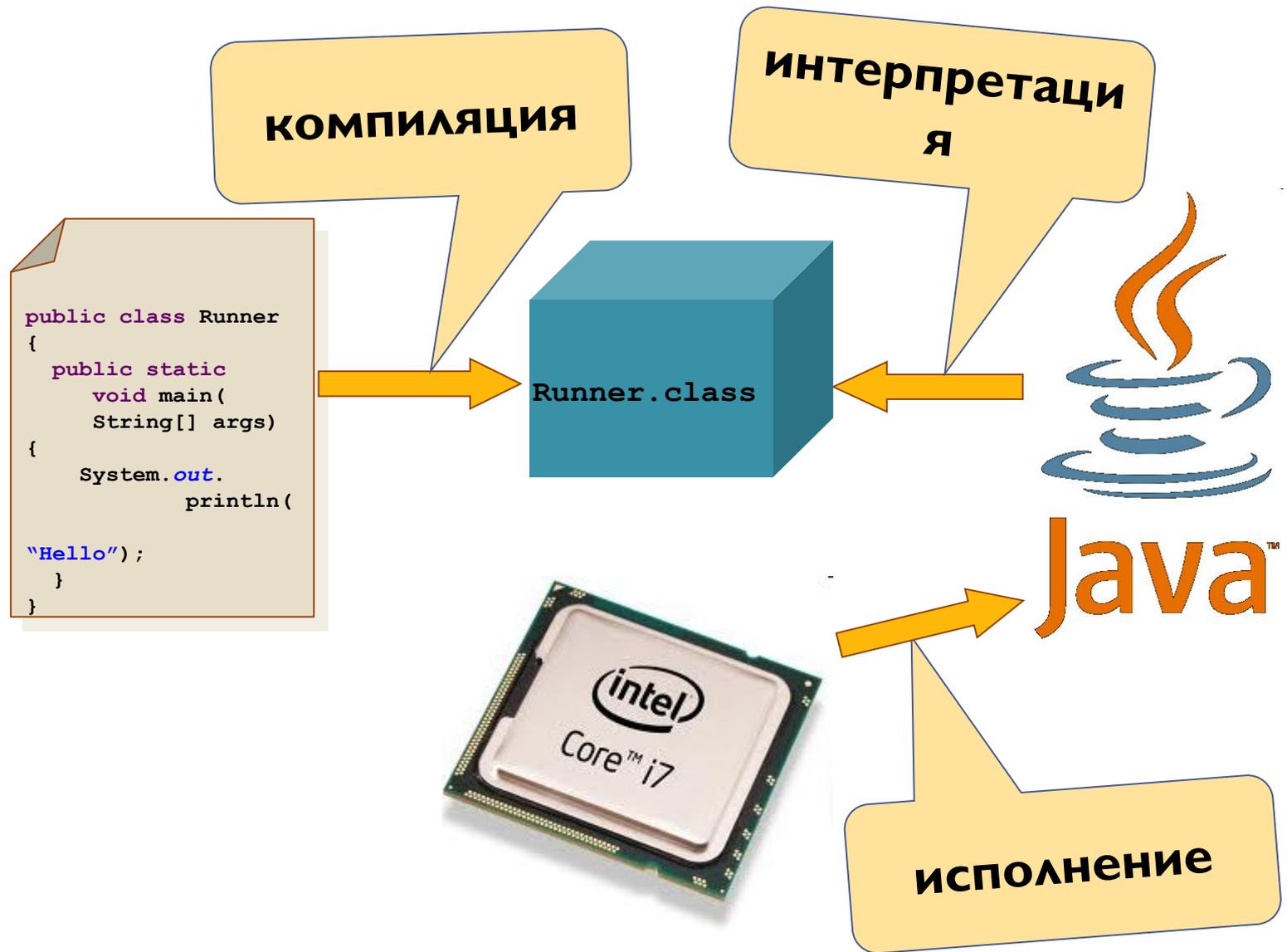
Минусы:

- Для запуска необходима дополнительная программа-интерпретатор
- Низкая скорость работы

Интерпретируемые языки программирования



Язык программирования Java



Область применения

- Консольные приложения
- Настольные приложения (AWT, Swing)
- Апплеты
- Мидлеты
- Сервлеты

Отличия от C++

- Отсутствие указателей и прямого доступа к памяти
- Механизм сборки мусора
- Отсутствуют деструкторы
- Неотделимость спецификации класса от реализации
- Отсутствие перегрузки операторов

Отличия от C++

- Отсутствие структур и объединений
- Отсутствуют шаблоны (templates), вместо них используют обобщения (generics)
- Отсутствуют беззнаковые целые числа
- Отсутствуют аргументы по умолчанию
- Не используются `goto` и `const`

Первая программа

```
// D:\java\proj\by\vsu\HelloWorld.java
```

```
package by.vsu;
```

```
public class HelloWorld {
```

```
    public static void main(String[]
```

```
        args) {
```

```
        System.out.println("Hello World");
```

```
    }
```

```
}
```

Компиляция программы

```
C:\Users\user> _
```

Компиляция программы

```
C:\Users\user> D:
```

```
D:> _
```

Компиляция программы

```
C:\Users\user> D:
```

```
D:> cd java\proj
```

```
D:\java\proj> _
```

Компиляция программы

```
C:\Users\user> D:
```

```
D:> cd java\proj
```

```
D:\java\proj> javac by\vsu\HelloWorld.java
```

```
D:\java\proj> _
```

Запуск программы

```
C:\Users\user> D:
```

```
D:> cd java\proj
```

```
D:\java\proj> javac by\vsu\HelloWorld.java
```

```
D:\java\proj> java by.vsu.HelloWorld
```

```
Hello World
```

```
D:\java\proj> _
```

Ввод данных

```
// ReadChar.java
```

```
public class ReadChar {  
    public static void main(String[]  
        args) {  
        int x;  
        try {  
            x = System.in.read();  
            char c = (char) x;  
        } catch(java.io.IOException e) {}  
    }  
}
```

Типы данных

Тип	Размер (бит)	Значения
boolean	8	true, false
byte	8	-128..127
short	16	-32 768..32 767
int	32	-2 147 483 648..2 147 483 647
long	64	-9 223 372 036 854 775 808.. 9 223 372 036 854 775 807
char	16	'\u0000'..' \uffff'
float	32	3.40282347E+38
double	64	1.797693134486231570E+308

Целочисленные литералы

- 23 – десятичное число
- 012 – восьмеричное число
- 0x7a – шестнадцатеричное число

тип литералов по умолчанию **int**

литерал типа **long** обозначается **L**

- 78L

Дробные литералы

- 1.234
- 0.123E-03

литералы типа **double**

литерал типа **float** обозначается **F**

- 2.34F

Символьные литералы

- `'a'`
- `'\123'`
- `'\u9ae8'`
- `'\n'`
- `'\t'`

Идентификаторы

- Не могут начинаться с цифры
- Не могут содержать знаки арифметических и логических операторов
- Не могут содержать символ **#**

Переменные

- **int a;**
- **char b = '#';**

Область видимости

```
// операторы (1)
while(a < 10) {
    // операторы (2)
    int x;
    // операторы (3)
    if(b > 0) {
        // операторы (4)
    }
    // операторы (5)
}
// операторы (6)
```

Арифметические операторы

+	Сложение	+ =	Сложение с присваиванием
-	Вычитание	- =	Вычитание с присваиванием
*	Умножение	* =	Умножение с присваиванием
/	Деление	/ =	Деление с присваиванием
%	Остаток от деления	% =	Остаток от деления с присваиванием
++	Инкремент	++	Декремент

Операторы сравнения

<	Меньше	>	Больше
<=	Меньше или равно	>=	Больше или равно
==	Равно	!=	Не равно

Логические операторы

&	И
&&	сокращённое И
 	ИЛИ
 	сокращённое ИЛИ
!	НЕ

Пример логических операторов

```
int a = 1, b = 0;
```

```
int c = 1, d = 1;
```

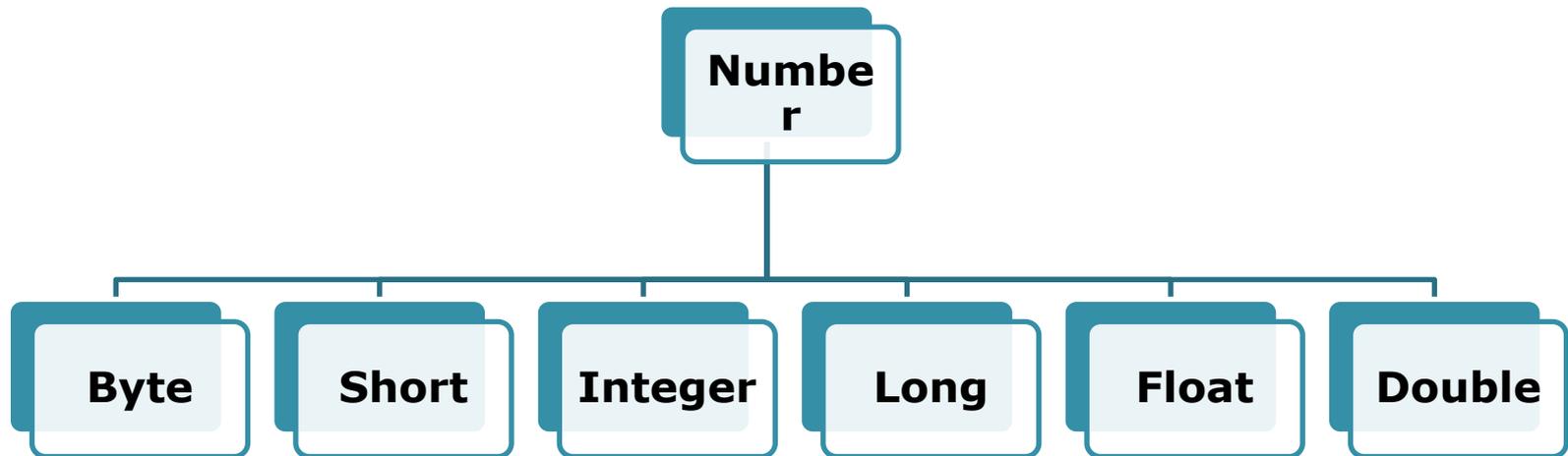
```
a == b & c == d++ // d = 2
```

```
a == b && c == d++ // d =  
1
```

Классы-оболочки

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

Иерархия классов-оболочек



Математические константы

- **Math.** ***π***
- **Math.** ***e***

Математические методы

- **Math.*abs*(x);**
- **Math.*sqrt*(x);**
- **Math.*cbrt*(x);**
- **Math.*pow*(x, y);**
- **Math.*hypot*(x, y);**

Математические методы

- **Math.*cos*(x);**
- **Math.*sin*(x);**
- **Math.*tan*(x);**

Математические методы

- **Math.acos(x);**
- **Math.asin(x);**
- **Math.atan(x);**
- **Math.atan2(y, x);**

Математические методы

- ***Math.toDegrees(x);***
- ***Math.toRadians(x);***

Математические методы

- **Math.*cosh*(x);**
- **Math.*sinh*(x);**
- **Math.*tanh*(x);**

Математические методы

- ***Math.exp(x);***
- ***Math.log(x);***
- ***Math.log10(x);***

Математические методы

- **Math.*ceil*(x);**
- **Math.*floor*(x);**
- **Math.*round*(x);**

Математические методы

- **`Math.random();`**
- **`Math.min(x, y);`**
- **`Math.max(x, y);`**