

Занятие 1

Чего не будет в курсе

- Docker
- Spring Security, Batch, Integration, Cloud...
- NoSQL
- Глубокое изучение Java Core
- Любые GUI технологии
- ...

Системы контроля версий

Системы контроля версий

Предназначены для управления исходным кодом. Решают следующие задачи

- Отслеживание истории изменений
- Разрешение конфликтов при одновременном редактировании
- Управление изменениями
- ...

Системы контроля версий

Обычно - серверное приложение. Стандартный подход включает в себя получение кода с сервера, внесение изменений, загрузка кода на сервер

- CVS
- SVN
- GIT
- Mercurial, etc.



Что такое Java?

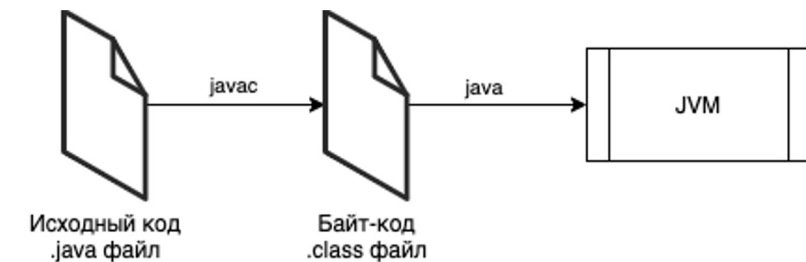
Что такое Java

Java - объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems, релиз первой версии - 21 января 1996 года

Основные особенности:

- Автоматическое управление памятью
- Богатая стандартная библиотека
- Длительный цикл обратной совместимости
- Кроссплатформенность
- ...

Как работают Java приложение



Исходный код - .java файл

Утилита javac - компилирует исходный код в байт-код

Утилита java - запускает байт-код на JVM

Основные понятия

- JVM. Виртуальная машина, которая умеет исполнять байт-код на конкретной ОС
- Байт-код. Унифицированное представление исходного кода, пригодного для выполнения на любой JVM
- `java` - утилита для запуска байт-кода
- `javac` - компилятор, превращающий исходный код в байт-код

JDK vs JRE

JRE - Java Runtime Environment. Минимальная реализация JVM для запуска Java приложений

JDK - Java Development Kit. Набор приложений для разработки и запуска Java приложений

Сильно упрощая:

JRE = утилита `java`

JDK = утилита `java` + компилятор `javac`

Реализации Java

OpenJDK, HotSpot, Azul, Corretto....

Azul

<https://www.azul.com/downloads-new/?package=jdk#zulu>



Среда разработки Java

Среда разработки - приложение, упрощающее написание исходного кода

- IntelliJ IDEA
- Eclipse
- NetBeans



Основы Java

Hello world

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello Ylab!");  
    }  
}
```

Hello world. v2

```
public class Main {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        String name = scanner.nextLine();  
        System.out.println("Hello " + name);  
    }  
}
```

Типы данных

- `byte` - 8 бит, целые числа -128 до 127
- `char` - 16 бит, Unicode символы. Поддерживает арифметические операции
- `short` - 16 бит, целые числа -2^{15} до $2^{15}-1$
- `int` - 32 бита, целые числа -2^{31} до $2^{31}-1$
- `long` - 64 бита, целые числа -2^{63} до $2^{63}-1$

- `float` - 32 бита, числа с плавающей точкой
- `double` - 64 бита, числа с плавающей точкой

- `boolean` - логический типа данных. Принимает значения `true` или `false`

- `String` - строка как последовательность символов

Операции с типами данных

Операция	Описание	Типы входных аргументов	Тип результата
<code>==, !=</code>	Сравнение на равенство	-	boolean
<code>>, >=, <, <=</code>	Сравнение	числа	boolean
<code>+, -, *, /, %</code>	Арифметические операции	числа	число
<code>!, &, &&, , </code>	Логические операции	boolean	boolean
<code>-</code>	Смена знака числа	число	число

Логические операции

A	!A
true	false
false	true

A	B	A & B
false	false	false
false	true	false
true	false	false
true	true	true

A	B	A B
false	false	false
false	true	true
true	false	true
true	true	true

&& и || вычисляют второй аргумент только при необходимости

`(b != 0) & (a % b == 0)` vs `(b != 0) && (a % b == 0)`

Переменные

```
int a = 1; // a == 1
```

```
double c = -1.5; // c == -1.5
```

```
int b = a + 1; // b == 2
```

```
boolean d = b > 1; // true
```

```
a = a + 1; // a == 2
```

```
a+=1; // a == 3
```

Live Coding Section

Задача. Объявить 2 целочисленные переменные. Присвоить им значения, вывести в консоль сумму, разность, произведение

Задача. Считать из консоли 2 целочисленных значения. Вывести в консоль сумму, разность, произведение

Условный оператор if

Выполняет один код, если определенное логическое выражение имеет значение true и другой в противном случае

```
if (выражение) {  
    // выполняем если выражение  
true  
}
```

```
if (выражение) {  
    // если выражение true  
} else {  
    // выполняем если false  
}
```

```
if (выражение) {  
    // если выражение выражение  
true  
} else if (выражение2) {  
    // выполняем если выражение2  
true  
} else if (выражение3) {  
    // выполняем если выражение3 true  
} else {  
    // выполняем если ни одно условие не  
    // выполняется  
}
```

Условный оператор switch

Изучить и разобрать самостоятельно

Циклы. While, do-while

Выполняет определенный код, пока условие имеет значение true

```
while (выражение) {  
    // выполняем пока выражение  
true  
}
```

```
do {  
    // выполняем пока выражение  
true  
} while (выражение);
```

```
int a = 5;  
while (a >= 0) {  
    System.out.println(a);  
    a = a - 1;  
}
```

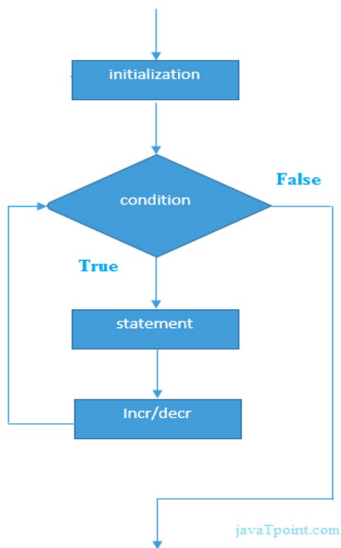
Циклы. for

Выполняет код определенное количество раз

```
for (инициализация; условие; модификация) {  
    // повторяемый код  
}
```

- Инициализация. Блок выполняется один раз, при старте цикла
- Условие. Проверяется, и если оно имеет значение `true` - выполняется код
- Модификация. Выполняется после каждого выполнения кода

Циклы. for



Каждый блок является опциональным (но так лучше не делать)
Если отсутствует инициализация/модификация - ничего не выполняется

Если отсутствует условие - оно считается true

```
for (;;) {  
    // бесконечный цикл  
}
```

```
for (int i = 1; i <= 10; i++) {  
    // обработка чисел от 1 до 10  
}
```

Циклы. Метки, break, continue

break - позволяет выйти из цикла раньше времени

continue - позволяет перейти к следующей итерации цикла раньше времени

Метка - позволяет выходить из вложенных циклов/переходить к следующей итерации

```
while (true) {  
    // что то делаем  
    if (условие) {  
        // тут понимаем,  
        //что надо выйти  
        break;  
    }  
    // продолжаем  
    работу  
}
```

```
while (условие1) {  
    // что то делаем  
    if (условие2) {  
        // тут понимаем,  
        //что надо идти к  
        // следующей итерации  
        continue;  
    }  
    // продолжаем работу  
}
```

```
label:  
while (условие1) {  
    // что то делаем  
    while (условие2) {  
        //тут понимаем,  
        //что надо выйти  
        //из всех циклов  
        break label;  
    }  
    // продолжаем работу  
}
```

Пара слов про строки

```
String s = "Hello";
```

```
s = s + " World"; // s= "Hello World";
```

```
System.out.println(s); // вывод строки в консоль
```

```
int n = s.length(); // возвращает длину строки. n == 11
```

```
char c = s.charAt(2); // возвращает символ на позиции 2 (начиная с 0)
```

Live Coding Section

Пользователь вводит число N .

Напишите программу, которая выводит на экран числа от 1 до N . При этом вместо чисел, кратных трем, программа должна выводить слово «Fizz», а вместо чисел, кратных пяти — слово «Buzz». Если число кратно и 3, и 5, то программа должна выводить слово «FizzBuzz».

Обзор домашнего задания