

# ОСНОВЫ Java

Занятие 2 - Условные  
операторы



# Условные операторы

- Иногда требуется изменить последовательность выполнения команд в программе в зависимости от некоторых факторов. Манипуляции над последовательностью выполнения кода называют **Control flow**
- В Java существуют условные операторы **if**, **if-else** и **switch**
- В качестве условия может выступать любое выражение возвращающее **boolean** значение

# Оператор if

```
if (condition) {  
    code  
    code  
    ...  
}
```

```
int a = 10;
```

```
int b = 10;
```

```
if (a == b) {
```

```
    System.out.println( "a==b! " );
```

```
}
```

// Можно обойтись без скобок

```
if (a == b)
```

```
    System.out.println( "a==b! " );
```

# Оператор if-else

```
If (condition) {  
    code  
    code  
    ...  
} else {  
    code  
    code  
    ...  
}
```

# Логические переменные

- Тип переменных `boolean` позволяет хранить в себе значения `true` или `false`. Кроме того можно присваивать переменной результат логического выражения

```
boolean isEqual = (2*2) == 4;
```

- Такая переменная может быть использована в качестве выражения условия

```
boolean lessonIsOver = false;
```

```
if (lessonIsOver) {  
    System.out.println("Go home");  
}
```

# Конструкция else if

```
If (condition1) {  
    code  
    code  
    ...  
} else {  
    if (condition2) {  
        code  
        code  
        ...  
    }  
}
```



```
If (condition1) {  
    code  
    code  
    ...  
} else if (condition2) {  
    code  
    code  
    ...  
}
```

**Удобнее!**

# Логические операторы

Оператор	Описание	Пример
<code>!=</code>	Не равно	<code>a != b</code>
<code>&lt;=</code>	Меньше или равно	<code>a &lt; b</code>
<code>&gt;=</code>	Больше или равно	<code>a &gt; b</code>
<code>&gt;</code>	Больше	<code>a &gt; b</code>
<code>&lt;</code>	Меньше	<code>a &lt; b</code>
<code>&amp;&amp;</code>	И	<code>a &lt; b &amp;&amp; c &gt; d</code>
<code>  </code>	Или	<code>a &lt; b    c &gt; d</code>
<code>!</code>	Не	<code>!( a &gt; b)</code>
<code>==</code>	Равно	<code>a == b</code>



# Вложенный оператор if

- Вложенные операторы if используют для проверки нескольких условий отдельно
- Конструкция условного оператора else всегда относится к ближайшему оператору if

```
if(i == 10)
{
    if(j < 20) a = b;
    if(k > 100) c = d;
    else a = c;    // else относится к if(k > 100)
}
else a = d;    // else относится к if(i == 10)
```

# Тернарный оператор ?

- Наиболее продвинутый вариант использования ветвления программного кода - использование тернарного оператора ?

**condition ? expression\_1 : expression\_2**

- Если условие истинно, тогда выполняется выражение expression\_1. Если нет - expression\_2

```
int largerNum;  
int lowNum = 9;  
int highNum = 27;  
  
if(lowNum < highNum)  
{  
    largerNum = highNum;  
} else {  
    largerNum = lowNum;  
}
```

```
int lowNum = 9;  
int highNum = 27;  
int largerNum = lowNum < highNum ? highNum : lowNum;
```

# Оператор switch

- В тех случаях, когда нужно проверять очень много однородных условий, использовать большие и сложные конструкции else if нецелесообразно. В таких случаях используют оператор выбора switch-case
- Выбор осуществляется в зависимости от значения выражения, в качестве которой часто используют переменные типа int или String. Далее идет набор case, которые определяют, что произойдет в каждом случае
- Особый вид case - default. Случай «default» происходит тогда, когда остальные случаи не срабатывают
- Оператор break осуществляет выход из switch

```
switch (выражение) {  
    case значение1:  
        //блок кода 1;  
        break;  
    case значение2:  
        //блок кода 2;  
        break;  
    ...  
    case значениеN:  
        //блок кода N;  
        break;  
    default:  
        блок N+1;  
}
```

```
int day = 3;
String dayString;
switch (day) {
    case 1: dayString = "Monday";
            break;
    case 2: dayString = "Tuesday";
            break;
    case 3: dayString = "Wednesday";
            break;
    case 4: dayString = "Thursday";
            break;
    case 5: dayString = "Friday";
            break;
    case 6: dayString = "Saturday";
            break;
    case 7: dayString = "Sunday";
            break;
    default: dayString = "Error";
            break;
}
System.out.print(dayString);
```

```
int numDays = 0;

switch (month) {
case 1:
case 3:
case 5:
case 7:
case 8:
case 10:
case 12:
    numDays = 31;
    break;
case 4:
case 6:
case 9:
case 11:
    numDays = 30;
    break;
case 2:
    numDays = 28;
    break;
default:
    break;
}

System.out.println("Number of days = " + numDays);
```