

Лекция 3

Основные структуры языка Java



Правила синтаксиса

Название класса пишется с заглавной и далее строчными символами. Если название класса состоит из нескольких слов, то слова пишутся слитно и каждое слово начинается с заглавной буквы:

```
public class MyNewClass { }
```

Название текстового файла должно совпадать с названием класса, который в нем записан:

```
MyNewClass.java
```

Правила синтаксиса (2)

Имена переменных пишутся строчными символами. Если имя состоит из двух слов, то второе слово и последующие пишутся с заглавной:

```
double airTemperature;
```

Правила синтаксиса (3)

Имена методов пишутся строчными символами. Если имя состоит из двух слов, то второе слово и последующие пишутся с заглавной:

```
public double getAirTemperature() { }
```

Первое слово как правило имеет смысл, говорящий о том, какое действие делает данный метод:

`set` — записывает данные в объект;

`get` — выдает данные из объекта.

Оператор условия if

В зависимости от результата вычисления логического выражения выполняется тот или иной блок операторов {...}.

Пример,

```
double t = 30;
```

```
if (t>20) {
```

```
    System.out.println("Теплая погода");
```

```
}
```

```
else {
```

```
    System.out.println("Холодная погода");
```

```
}
```

Операторы цикла

Для выполнения блока операторов несколько раз предназначены операторы цикла:

for

while

do ... while

Оператор цикла for

В цикле **for** блок операторов повторяется заданное количество раз, в зависимости от заданного условия.

Для работы оператора **for** необходимо указать:

- начальное значение целой переменной,
- условие прекращения цикла,
- выражение инкремента переменной

Оператор цикла for (2)

Например,

```
int i;
```

```
for(i=0; i<10; i++) {  
    System.out.printf("%d\n", i);  
}
```

Какие числа появятся на консоле?

Здесь $i++$ - это сокращенная запись от $i=i+1$

Оператор цикла for (3)

Другой пример,

```
int i;
```

```
int sum = 0;
```

```
for(i=0; i<10; i++) {
```

```
    sum = sum + i;
```

```
}
```

```
System.out.printf("Сумма равна %d\n", sum);
```

```
System.out.printf("После цикла i = %d\n", i);
```

Какие числа появятся на консоле?

Оператор while

Блок операторов выполняется пока выполняется логическое выражение.

Например,

```
int i = 0;
```

```
int sum = 0;
```

```
while(i<10) {
```

```
    sum += i;
```

```
    i++;
```

```
}
```

```
System.out.printf("Сумма равна %d\n", sum);
```

Какое число появится на консоле?

Оператор do ... while

Аналогичен оператору **while**, но условие проверяется после выполнения блока операторов.

Например,

```
int i = 0;  
do {  
    i++;  
}  
while(i<0);  
System.out.printf("После цикла i = %d", i);
```

Какое число появится на консоле?

Операторы, изменяющие последовательность действий

Оператор **continue**

Используется в циклах **for**, **while**, **do .. while** для досрочного перехода к началу следующего шага выполнения блока операторов

Оператор **break**

Используется для досрочного прекращения выполнения операторов **for**, **while**, **do .. while**

Управление передается оператору, идущему сразу за циклом.

Массивы данных

Массив — это индексированная совокупность переменных одного типа. Каждой переменной в массиве присвоен индекс, начиная с нуля.

Например, массив из трех элементов может быть создан явно:

```
double[ ] humidity = { 78, 89, 98 };
```

```
System.out.printf("%f", humidity[1]);
```

Массивы данных (2)

Какое число появится на консоле?

Массивы данных (3)

Массив может быть задан и неявно:

```
double[] humidity = new double[3];
```

Во этом случае по умолчанию всем элементам присваивается значение нуль.

Значения элементов массива можно изменить используя оператор присваивания:

```
humidity[0] = 21;
```

```
humidity[1] = 72;
```

```
humidity[2] = 38;
```

Использование массивов

```
// Объявление переменных
int i;
int N = 20;
double dz = 500.0;
double[ ] z = new double[N];
double[ ] T = new double[N];

// Инициализация массивов
for(i=0; i<N; i++) {
    z[i] = dz*i;
    T[i] = 298.15 - 0.0065*z[i];
}
```


Использование массивов (2)

// Печать всего содержимого массивов

```
for(i=0; i<N; i++) {  
    System.out.printf("%.0f\t%.1f\n", z[i], T[i]);  
}
```

// Печать данных от 0 до 2 км

```
for(i=0; i<N; i++) {  
    if(z[i]>2000) break;  
    System.out.printf("%.0f\t%.1f\n", z[i], T[i]);  
}
```

Использование массивов (3)

```
// Печать данных на высотах, кратных 1 км
for(i=0; i<N; i++) {
    if((z[i]%1000)>0.5) continue;
    System.out.printf("%.0f\t%.1f\n", z[i], T[i]);
}
```

Вопросы?

<http://www.chukin.ru/edu/java/>

chukin@rshu.ru

