Basic Java №6

`static` позволяет создать метод, к которому можно обратиться напрямую без создания экземпляра класса, в котором этот метод находится.

Условные операторы if, else if и else позволяют выполнять различные блоки кода в зависимости от выполнения или невыполнения определенных условий.

```
public class Main {
    public static void main(String[] args) {
        int number = 10;

        // Оператор if проверяет условие
        if (number > 0) {
            System.out.println("Число положительное.");
        }
        // Оператор else if проверяет другое условие, если
предыдущее условие if не выполнено
        else if (number < 0) {
            System.out.println("Число отрицательное.");
        }
        // Оператор else выполняется, если ни одно из условий if или
else if не выполнено
        else {
            System.out.println("Число равно нулю.");
        }
    }
}
```

Описание работы:

1. **if**: Оператор `if` проверяет заданное условие. Если условие истинно (`true`), то выполняется блок кода, заключенный в фигурные скобки `{}`.
2. **else if**: Оператор `else if` используется для проверки дополнительного условия, если предыдущее условие `if` оказалось ложным (`false`). Можно использовать несколько операторов `else if` для проверки различных условий.
3. **else**: Оператор `else` используется для выполнения блока кода, если все предыдущие условия `if` и `else if` оказались ложными.

Блок кода else выполняется в любом случае, если ни одно из предыдущих условий не было выполнено.

Условные операторы могут быть вложены друг в друга и независимы от внешних условий.

Оператор `switch` в языке Java используется для выполнения одного из нескольких блоков кода в зависимости от значения выражения. Он является альтернативой оператору `if-else` для случаев, когда нужно сравнивать одно и то же значение с несколькими возможными вариантами.

Структура оператора switch:

```java
switch (выражение) {
    case значение1:
        // Блок кода для случая значение1
        break;
    case значение2:
        // Блок кода для случая значение2
        break;
    // ...
    default:
        // Блок кода для случая по умолчанию
        break;
}
```

Описание работы:

1. **выражение**: Значение, которое будет сравниваться с каждым из значений `case`. Это значение может быть целым числом, строкой, перечислением и т.д.

2. **case значениеX**: Каждое значение `case` представляет собой возможное значение выражения. Когда значение выражения совпадает с `значениеX`, выполняется соответствующий блок кода.

3. **break**: Оператор `break` завершает выполнение текущего блока `case`. Без `break` выполнение продолжится до конца блока `switch`, включая другие блоки `case`.

4. **default**: Блок `default` выполняется, если ни одно из значений `case` не совпало с выражением. Он является необязательным, но полезным для обработки непредвиденных значений.

Пример использования оператора switch:

```java
public class Main {
    public static void main(String[] args) {
        int day = 3;

        switch (day) {
            case 1:
                System.out.println("Понедельник");
                break;
            case 2:
                System.out.println("Вторник");
                break;
            case 3:
                System.out.println("Среда");
                break;
            case 4:
                System.out.println("Четверг");
                break;
            case 5:
                System.out.println("Пятница");
                break;
            case 6:
                System.out.println("Суббота");
                break;
            case 7:
                System.out.println("Воскресенье");
                break;
            default:
                System.out.println("Неверный день недели");
                break;
        }
    }
}
```

В этом примере, переменная `day` сравнивается с каждым значением `case`. Когда `day` равно 3, выполняется блок кода для случая `case 3`, который выводит "Среда". Если ни одно из значений не совпало, выполняется блок кода `default`.

Операторы относительности (также называемые операторы сравнения) используются для сравнения значений в Java. Вот список операторов сравнения и примеры их использования:

1. **==**: Равно. Проверяет, равны ли два значения.
2. **!=**: Не равно. Проверяет, не равны ли два значения.
3. **>**: Больше. Проверяет, больше ли одно значение другого.
4. **>=**: Больше или равно. Проверяет, больше ли или равно одно значение другому.
5. **<**: Меньше. Проверяет, меньше ли одно значение другого.
6. **<=**: Меньше или равно. Проверяет, меньше ли или равно одно значение другому.

Логические операции позволяют работать с булевыми (логическими) значениями в программировании. В Java основными логическими операторами являются логическое умножение (AND), логическое сложение (OR) и логическое отрицание (NOT).

Логическое умножение (AND) - оператор `&&`: Оператор `&&` возвращает `true`, если оба операнда являются `true`.

Логическое сложение (OR) - оператор `||`: Оператор `||` возвращает `true`, если хотя бы один из операндов является `true`.

Логическое отрицание (NOT) - оператор `!`: Оператор `!` возвращает противоположное значение операнду.

Опасно изменять значения статических переменных, так как они изменятся во всех экземплярах класса, которые ссылаются на эти переменные.