



Язык программирования Java. Начальный уровень

Мультимедийный курс

автор: Васильев А.Н.

www.vasilev.kiev.ua

Киев 2017



Лекция 3. Управляющие инструкции



- Условный оператор if
- Оператор выбора switch
- Оператор цикла while
- Оператор цикла do-while
- Оператор цикла for
- Перехват исключений

*Значит, такое предложение:
сейчас мы нажимаем на
контакты и перемещаемся к
вам. Но если эта машинка не
срабатывает, тогда уж Вы с
нами переместитесь, куда мы
Вас переместим!
из к/ф "Кин-дза-дза"*

Условный оператор if



Синтаксис

```
if (условие) {  
    // команды  
}  
else {  
    // команды  
}
```

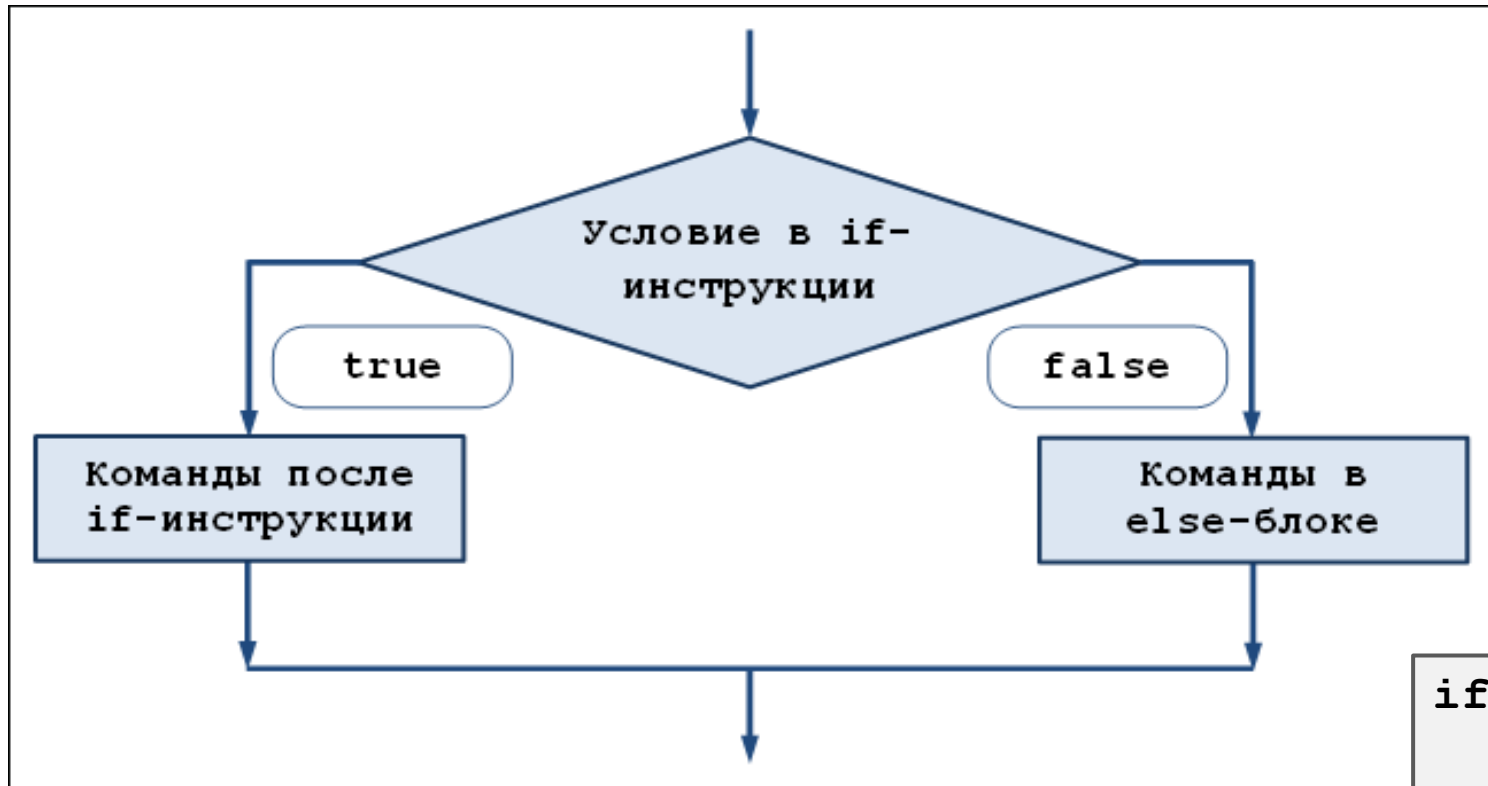
**Упрощенная форма
(отсутствует else-блок)**

```
if (условие) {  
    // команды  
}
```

Условный оператор if



Выполнение



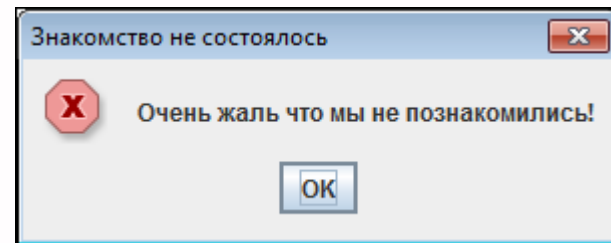
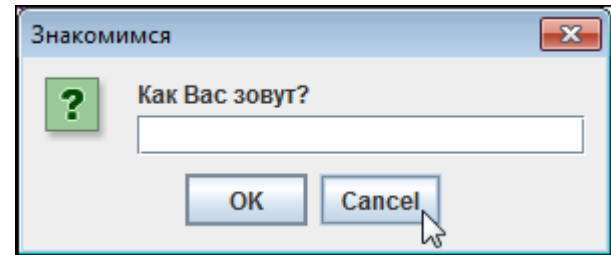
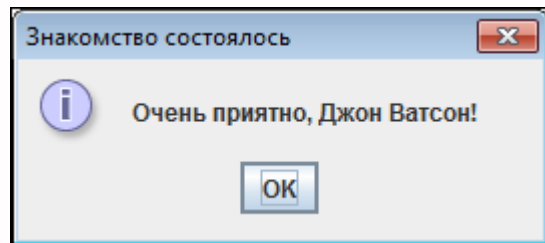
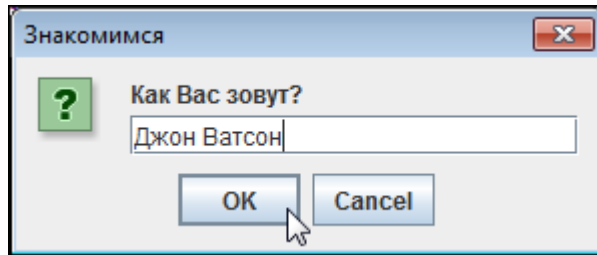
```
if(условие) {  
    // команды  
}  
else{  
    // команды  
}
```

Использование условного оператора



```
import static javax.swing.JOptionPane.*;
class UsingIfDemo{
    public static void main(String[] args){
        int icon;                // Тип пиктограммы
        String msg,title,name; // Сообщение, заголовок, имя
        name=showInputDialog(null,"Как Вас зовут?",
            "Знакомимся",QUESTION_MESSAGE);
        if (name==null){ // Если нет текста (пустая ссылка)
            icon=ERROR_MESSAGE;
            msg="Очень жаль что мы не познакомились!";
            title="Знакомство не состоялось";
        }
        else{ // Если ссылка не пустая
            icon=INFORMATION_MESSAGE;
            msg="Очень приятно, "+name+"!";
            title="Знакомство состоялось";
        }
        showMessageDialog(null,msg,title,icon);
    }
}
```

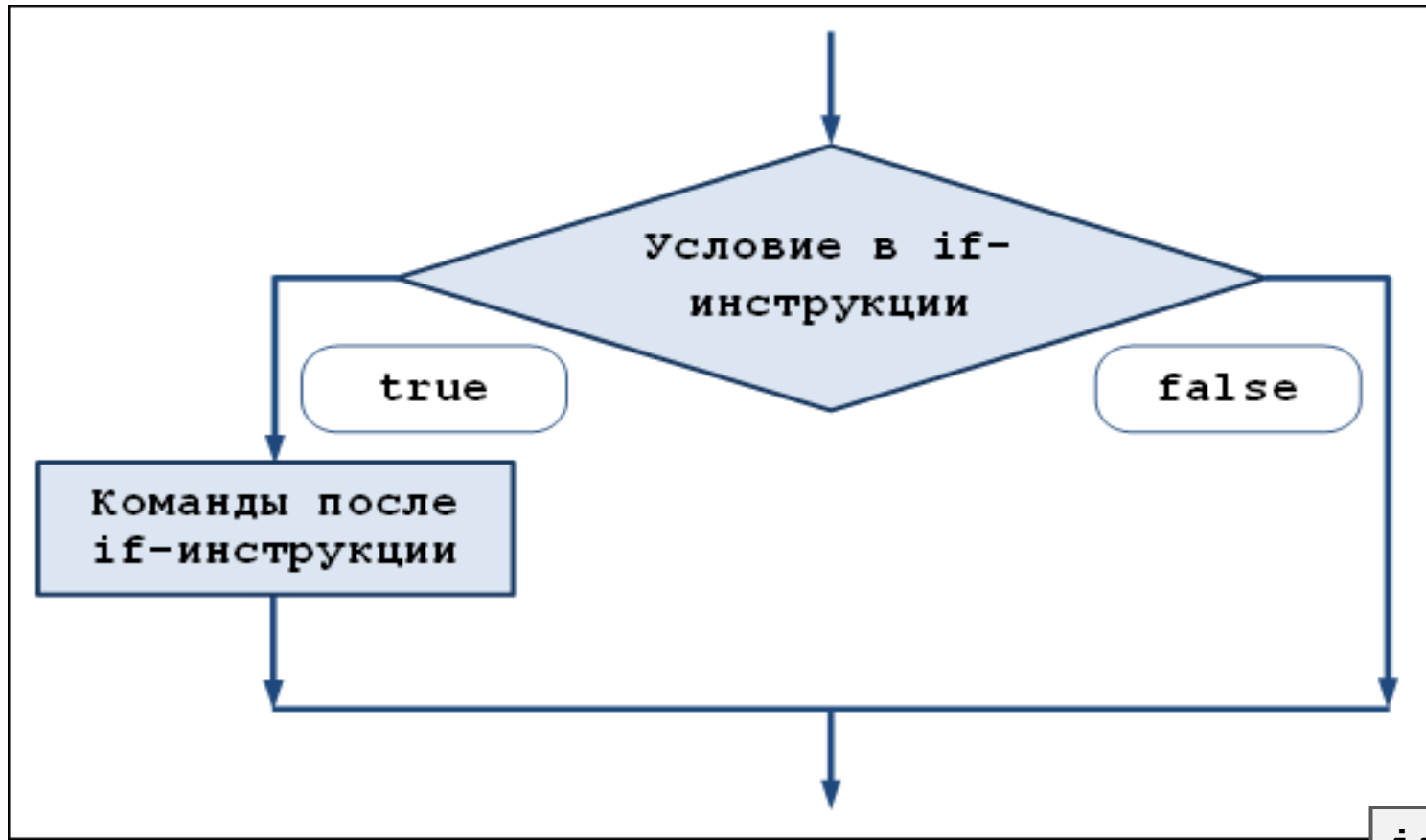
Условный оператор: выполнение программы



Условный оператор if



Выполнение (упрощенная форма)



```
if (условие) {  
    // команды  
}
```

© Васильев А.Н.

Упрощенная форма условного оператора

```
import static javax.swing.JOptionPane.*;
class AnotherIfDemo{
    public static void main(String[] args){
        int icon=ERROR_MESSAGE;    // Тип пиктограммы
        // Переменные с начальными значениями:
        String msg="Очень жаль что мы не познакомились!",
               title="Знакомство не состоялось",
               name;
        name=showInputDialog(null,"Как Вас зовут?",
                              "Знакомимся",QUESTION_MESSAGE);
        if(name!=null){    // Если ссылка не пустая
            // Новый тип пиктограммы:
            icon=INFORMATION_MESSAGE;
            // Новый текст сообщения:
            msg="Очень приятно, "+name+"!";
            // Новый заголовок:
            title="Знакомство состоялось";
        }
        showMessageDialog(null,msg,title,icon);
    }
}
```


Вложенные условные операторы

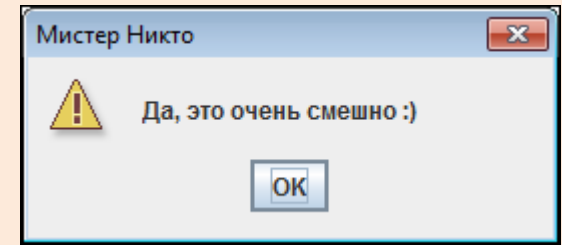
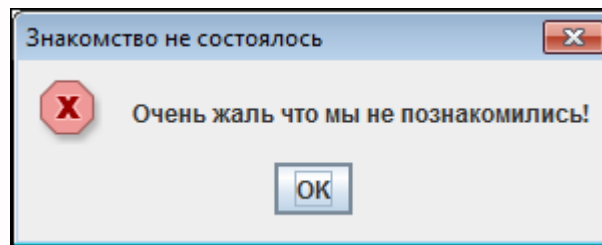
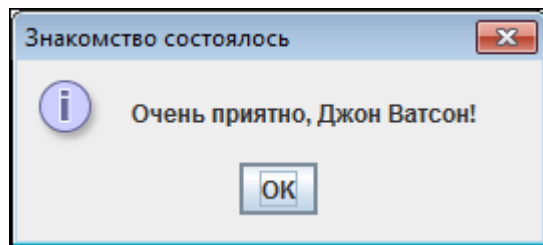
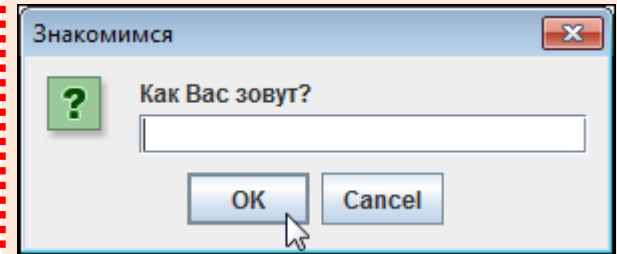
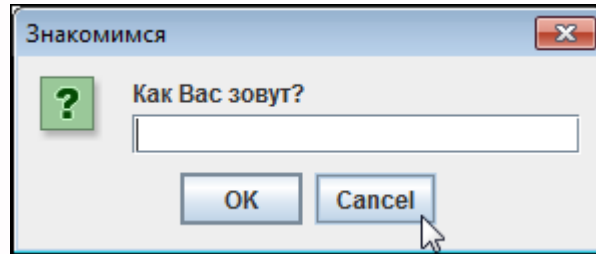
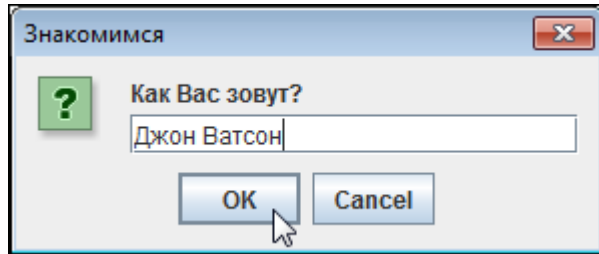


```
// Внешний условный оператор
if (условие) {
    // Внутренний условный оператор
    if (условие) {
        // команды
    }
    // Блок else внутреннего оператора
    else {
        // команды
    }
}
// Блок else внешнего оператора
} else {
    // команды
}
```



```
import static javax.swing.JOptionPane.*;
class NestedIfDemo{
    public static void main(String[] args){
        int icon;                // Тип пиктограммы
        String msg,title,name; // Сообщение, заголовок, имя
        name=showInputDialog(null,"Как Вас зовут?",
            "Знакомимся",QUESTION_MESSAGE);
        if(name!=null){ // Если ссылка не пустая
            if(name.equals("")){ // Если в поле нет текста
                icon=WARNING_MESSAGE;
                msg="Да, это очень смешно :) ";
                title="Мистер Никто";
            }
            else{ // Если в поле есть текст
                icon=INFORMATION_MESSAGE;
                msg="Очень приятно, "+name+"!";
                title="Знакомство состоялось";
            }
        }
        else{ // Если ссылка не пустая
            icon=ERROR_MESSAGE;
            msg="Очень жаль что мы не познакомились!";
            title="Знакомство не состоялось";
        }
        showMessageDialog(null,msg,title,icon);
    }
}
```

Вложенные условные операторы: выполнение программы



Оператор выбора switch



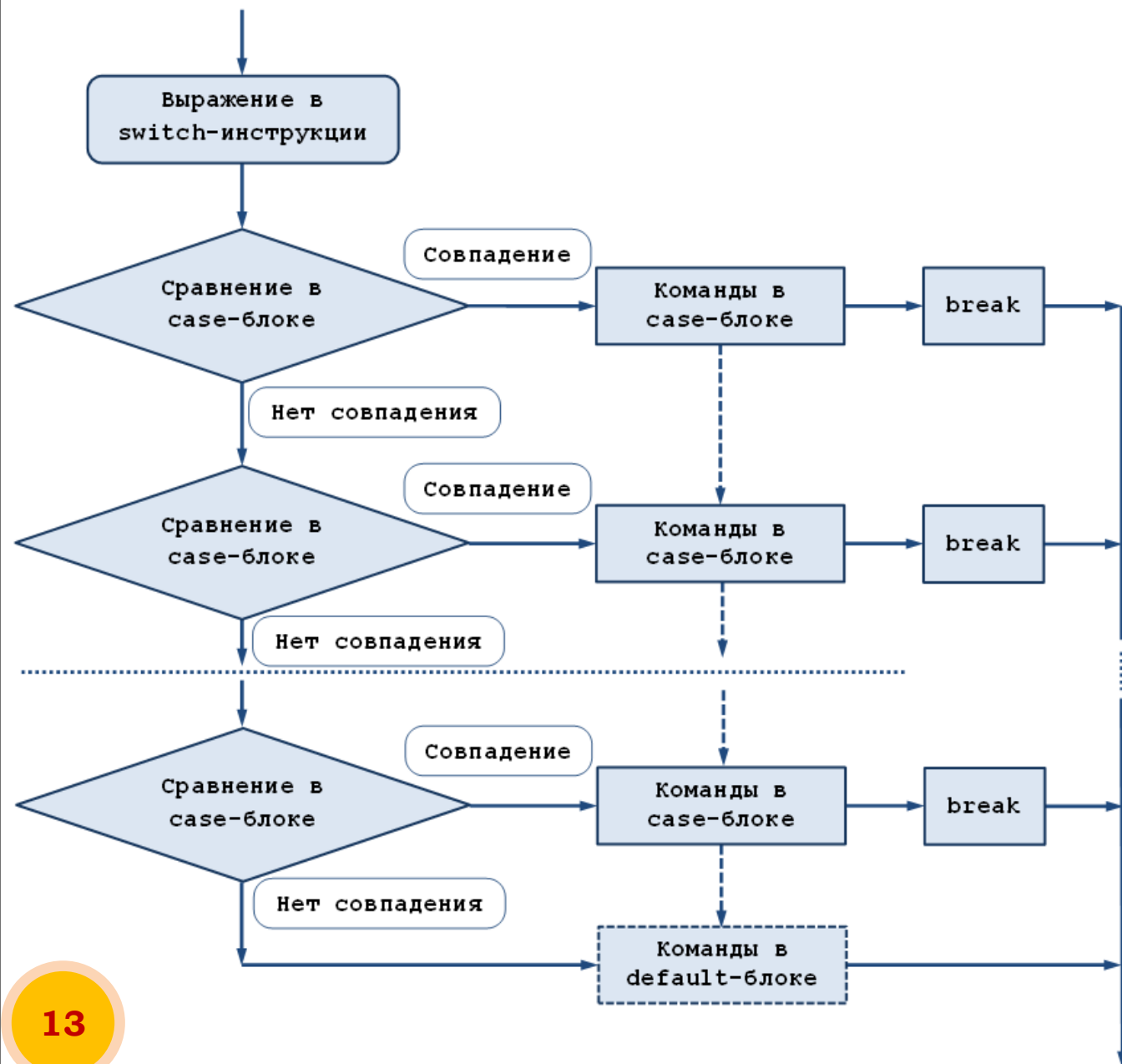
```
switch (выражение) {  
    case значение_1:  
        // команды  
        break;  
    case значение_2:  
        // команды  
        break;  
    ...  
    case значение_N:  
        // команды  
        break;  
    default:  
        // команды  
}
```

Синтаксис

Инструкцию break
можно не
использовать (но это
имеет последствия)

Блок default можно
не использовать

Оператор выбора switch: выполнение



```
switch(выражение) {
    case значение_1:
        // команды
        break;
    case значение_2:
        // команды
        break;
    ...
    case значение_N:
        // команды
        break;
    default:
        // команды
}
```

Программа: оператор выбора



```
import static javax.swing.JOptionPane.*;
class SwitchDemo{
    public static void main(String[] args){
        int number;
        String name;
        number=Integer.parseInt(
            showInputDialog("Введите число:")
        );
        switch(number){    // Оператор выбора
            case 1:
                name="Единица";
                break;
            case 2:
                name="Двойка";
                break;
            case 3:
                name="Тройка";
                break;
            default:
                name="Неизвестное число";
        }
        showMessageDialog(null,name);
    }
}
```



Оператор выбора: выполнение программы



Input

Введите число:

1

OK Cancel



Message

Единица

OK

Input

Введите число:

2

OK Cancel



Message

Двойка

OK

Input

Введите число:

3

OK Cancel



Message

Тройка

OK

Input

Введите число:

5

OK Cancel



Message

Неизвестное число

OK



```
import static javax.swing.JOptionPane.*;
class AnotherSwitchDemo{
    public static void main(String[] args){
        int number;
        String txt="";    // Обратите внимание
        number=Integer.parseInt(
            showInputDialog("Введите число от 1 до 9:")
        );
        switch(number) {    // Оператор выбора
            case 1:
            case 9:
                txt="Вы ввели нечетное, \n но не простое число.";
                break;
            case 2:
            case 3:
            case 5:
            case 7:
                txt="Вы ввели простое число.";
                break;
            case 4:
            case 8:
                txt="Вы ввели число - степень двойки.";
                break;
            case 6:
                txt="Вы ввели 6 - совершенное число.";
        }
        showMessageDialog(null,txt);
    }
}
```


Оператор с пустыми блоками: выполнение программы



Input

Введите число от 1 до 9:

9

OK Cancel



Message

Вы ввели нечетное, но не простое число.

OK

Input

Введите число от 1 до 9:

5

OK Cancel



Message

Вы ввели простое число.

OK

Input

Введите число от 1 до 9:

8

OK Cancel



Message

Вы ввели число - степень двойки.

OK

Input

Введите число от 1 до 9:

6

OK Cancel



Message

Вы ввели 6 - совершенное число.

OK

Оператор цикла while

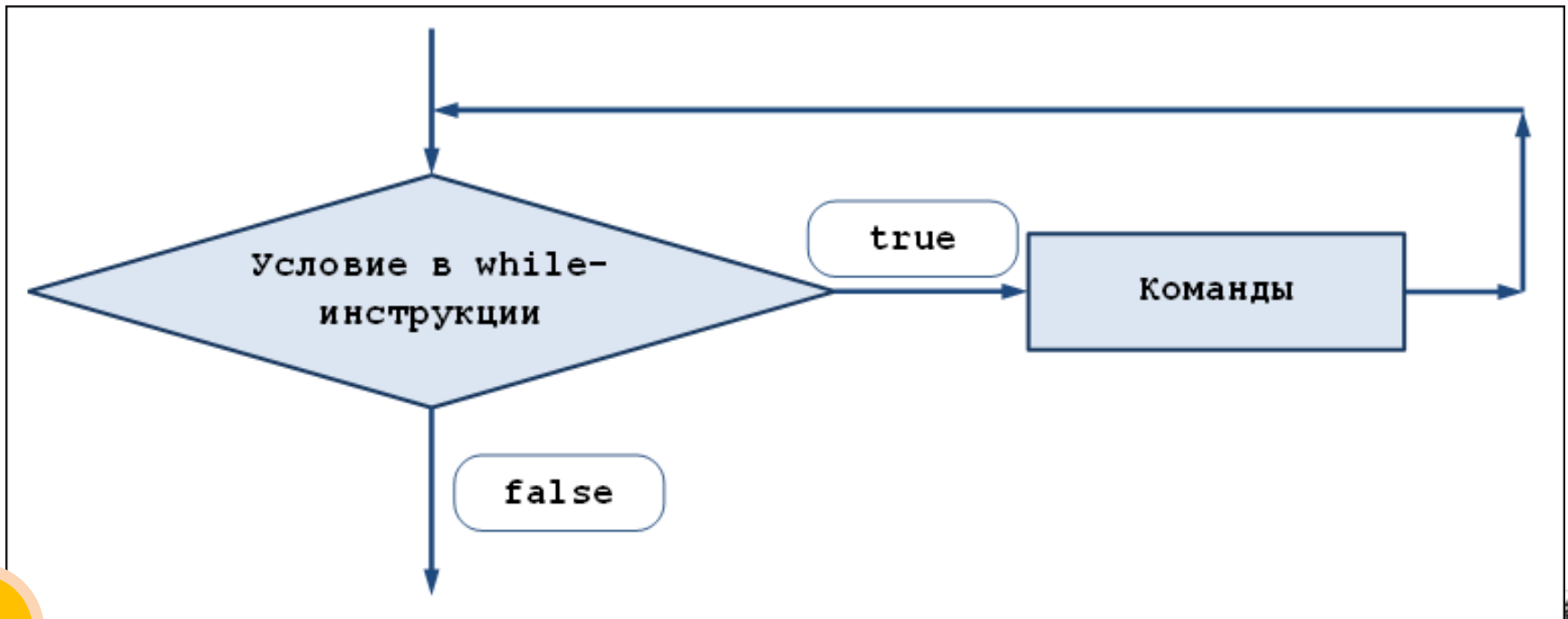


```
while (условие) {  
    // команды  
}
```



Синтаксис

Выполнение



Программа: оператор цикла while



```
class WhileDemo{
    public static void main(String[] args){
        // Верхняя граница суммы, индексная переменная,
        // значение суммы:
        int n=10,k=1,s=0;
        // Отображение верхней границы суммы:
        System.out.print("Сумма чисел от 1 до "+n+": ");
        // Оператор цикла:
        while (k<=n) {
            s+=k;    // Добавляем слагаемое в сумму
            k++;     // Новое значение индексной переменной
        }
        // Отображение результата:
        System.out.println(s);
    }
}
```

Программа: оператор цикла while



```
class WhileDemo{
    public static void main(String[] args){
        // Верхняя граница суммы и значение суммы:
        int n=10,s=0;
        // Отображение верхней границы суммы:
        System.out.print("Сумма чисел от 1 до "+n+": ");
        // Оператор цикла:
        while (n>0) {
            s+=n;    // Добавляем слагаемое в сумму
            n--;     // Новое значение переменной
        }
        // Отображение результата:
        System.out.println(s);
    }
}
```

Оператор цикла do-while

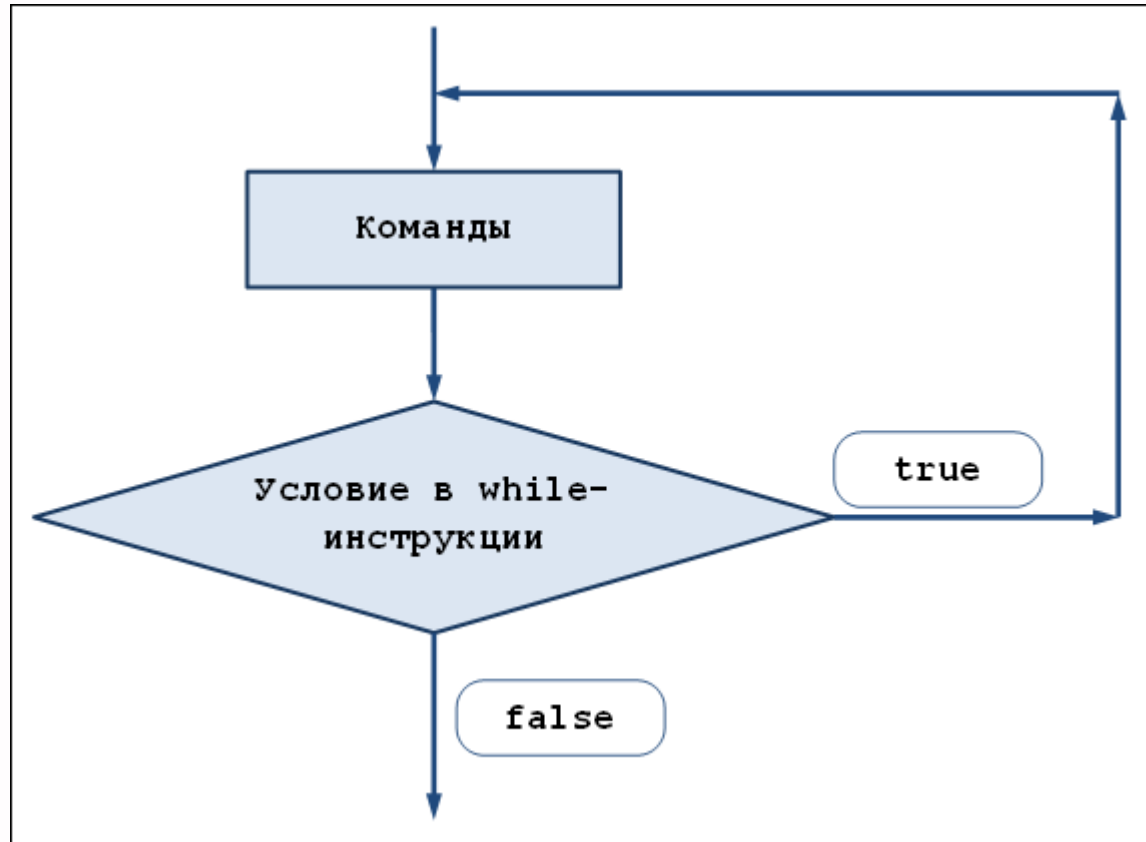


Синтаксис



```
do{  
    // команды  
}while (условие) ;
```

Выполнение



Программа: оператор цикла do-while



```
class DoWhileDemo{
    public static void main(String[] args){
        // Верхняя граница суммы, индексная переменная,
        // значение суммы:
        int n=10,k=1,s=0;
        // Отображение верхней границы суммы:
        System.out.print("Сумма чисел от 1 до "+n+": ");
        // Оператор цикла:
        do{
            s+=k;    // Добавляем слагаемое в сумму
            k++;     // Новое значение индексной переменной
        }while(k<=n) ;
        // Отображение результата:
        System.out.println(s) ;
    }
}
```

Оператор цикла for



Синтаксис

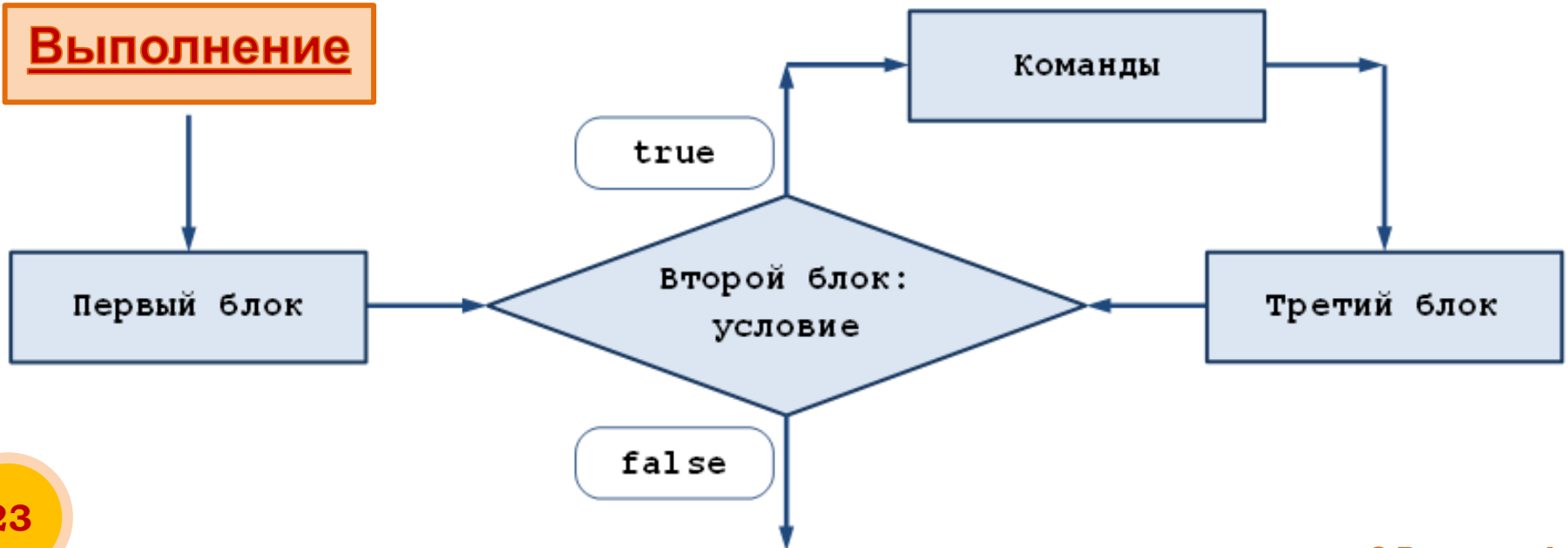
Первый блок

Второй блок

Третий блок

```
for (инициализация ; условие ; инкремент / декремент) {  
    // команды  
}
```

Выполнение



Программа: оператор цикла for



```
class ForDemo{
    public static void main(String[] args){
        // Верхняя граница суммы, индексная переменная,
        // значение суммы:
        int n=10,k,s=0;
        // Отображение верхней границы суммы:
        System.out.print("Сумма чисел от 1 до "+n+": ");
        // Оператор цикла:
        for (k=1;k<=n;k++) {
            s+=k;    // Добавляем слагаемое в сумму
        }
        // Отображение результата:
        System.out.println(s);
    }
}
```


Программа: оператор цикла for



```
class ForDemo{
    public static void main(String[] args){
        // Верхняя граница суммы и значение суммы
        // индексная переменная объявлена
        // в операторе цикла):
        int n=10,s=0;
        // Отображение верхней границы суммы:
        System.out.print("Сумма чисел от 1 до "+n+": ");
        // Оператор цикла:
        for(int k=1;k<=n;k++){ // Обратите внимание!!!
            s+=k;    // Добавляем слагаемое в сумму
        }
        // Отображение результата:
        System.out.println(s);
    }
}
```

Программа: оператор цикла for



```
class ForDemo{
    public static void main(String[] args){
        // Верхняя граница суммы, индексная переменная,
        // значение суммы:
        int n=10,k,s;
        // Отображение верхней границы суммы:
        System.out.print("Сумма чисел от 1 до "+n+": ");
        // Оператор цикла:
        for (k=1,s=0;k<=n;s+=k,k++) ;
        // Отображение результата:
        System.out.println(s);
    }
}
```

Программа: оператор цикла for



```
class ForDemo{
    public static void main(String[] args){
        // Верхняя граница суммы, индексная переменная,
        // значение суммы:
        int n=10,k=1,s=0;
        // Отображение верхней границы суммы:
        System.out.print("Сумма чисел от 1 до "+n+": ");
        // Оператор цикла:
        for (;k<=n;){
            s+=k; // Прибавление слагаемого к сумме
            k++;  // Новое значение индексной переменной
        }
        // Отображение результата:
        System.out.println(s);
    }
}
```

Программа: оператор цикла for



```
class ForDemo{
    public static void main(String[] args){
        // Верхняя граница суммы, индексная переменная,
        // значение суммы:
        int n=10,k=1,s=0;
        // Отображение верхней границы суммы:
        System.out.print("Сумма чисел от 1 до "+n+": ");
        // Оператор цикла:
        for(;;){
            s+=k; // Прибавление слагаемого к сумме
            k++; // Новое значение индексной переменной
            // Условный оператор:
            if(k>n) break;
        }
        // Отображение результата:
        System.out.println(s);
    }
}
```

- Инструкция **break** завершает выполнение оператора цикла
- Инструкция **continue** завершает выполнение текущей итерации (цикла).

Перехват исключений



Общая схема (упрощенный вариант)

```
try{  
    // команды  
}catch (Exception e) {  
    // команды  
}
```



Контролируемый код в try-блоке



Код в catch-блоке выполняется при обработке ошибки

Выполнение

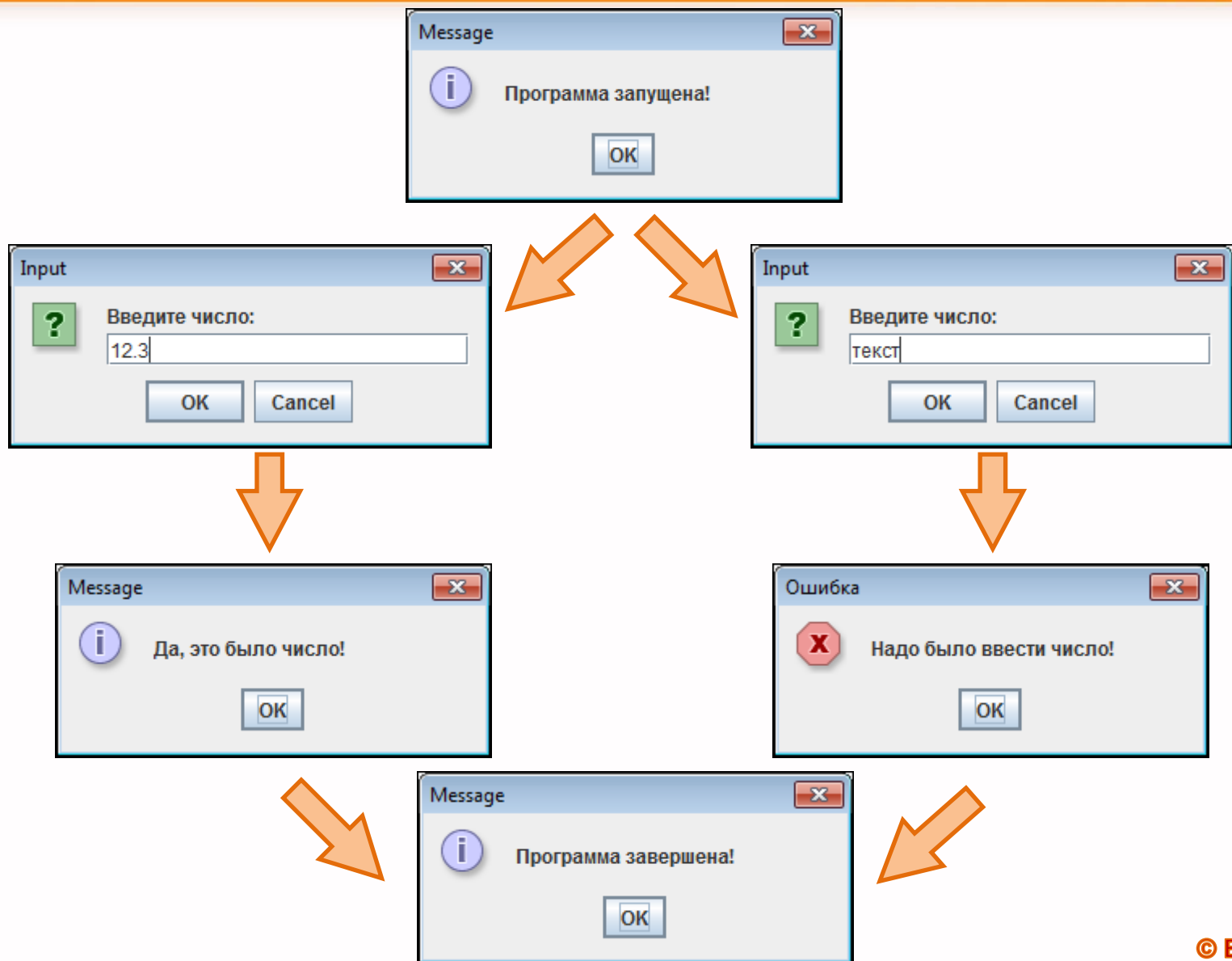
- Если при выполнении кода в try-блоке ошибок не возникает, то код в catch-блоке игнорируется
- Если при выполнении кода в try-блоке возникает ошибка, то выполнение кода в блоке прекращается и начинает выполняться код в catch-блоке

Перехват исключения



```
import static javax.swing.JOptionPane.*;
class TryCatchDemo{
    public static void main(String[] args){
        showMessageDialog(null,"Программа запущена!");
        try{ // Контролируемый код
            Double.parseDouble(showInputDialog("Введите
число:")) ;
            showMessageDialog(null,"Да, это было число!");
        }
        catch(Exception e){ // Блок обработки ошибок
            showMessageDialog(null,"Надо было ввести число!",
                "Ошибка",ERROR_MESSAGE);
        }
        showMessageDialog(null,"Программа завершена!");
    }
}
```

Перехват исключения: результат выполнения



Домашнее задание



- Напишите программу, которая проверяет, делится ли введенное пользователем число на 3 и одновременно на 7.
- Напишите программу, которая вычисляет сумму квадратов натуральных чисел (учесть, что $1^2 + 2^2 + 3^2 + \dots + n^2 = \frac{n(n+1)(2n+1)}{6}$). Попробуйте использовать разные операторы цикла.
- Напишите программу, которая по номеру дня в неделе выводит его название (например, 1 соответствует понедельнику, 2 - вторнику, и так далее).