

Реализация основных алгоритмических
структур на языке Java.
(пособие для учащихся)

Автор: Тюленев Евгений Анатольевич

Место работы: МБОУ "Боровихинская сош"

Должность: Учитель

Оглавление

1 Введение.....	3
2 Выражение — алгоритм.....	3
3 Определение переменных. Операция присваивания.....	4
4 Класс. Свойства и методы.	6
5 Консольный ввод.....	10
6 Управляющие операторы.....	11
6.1 Логические выражения.....	12
6.2 Условный оператор.....	14
6.3 Оператор выбора.....	16
6.4 Циклы.....	17
6.4.1 Цикл с предусловием.....	17
6.4.2 Цикл с постусловием.....	19
7 Массивы.....	20
7.1 Заполнение массивов.....	21
7.2 Обработка массива.....	23
8 Приложения.....	26
8.1 Необходимое программное обеспечение.....	26

1 Введение.

Исполнитель — это объект способный выполнять определенный набор команд.

Алгоритм — это последовательность инструкций, выполнение которой ведет к решению задачи.

Требования к алгоритму:

1. *Понятность*: инструкции алгоритма должны быть понятны исполнителю.
2. *Замкнутость*: в результате выполнения алгоритма должна быть решена именно данная задача.
3. *Дискретность*: выполнение очередной инструкции не начинается, пока не закончится выполнение текущей.
4. *Конечность*: выполнение инструкций алгоритма должно когда-то закончиться.

2 Выражение — алгоритм.

Для того, чтобы вычислить значение выражения $3(12-6)$ нужно действовать по алгоритму:

1. из 12 вычесть 6;
2. результат первого действия умножить на 3;
3. сообщить результат второго действия.

Программирование можно рассматривать как процесс составления выражений.

Требования к записи выражений:

1. расставлять знаки всех операций.
2. дробная черта заменяется делением.
3. у десятичной дроби целая от дробной части отделяется точкой.
4. если необходимо, то числитель и знаменатель заключается в скобки.

Обозначение арифметических операций:

+	сложение
-	вычитание
*	умножение
/	деление (для целых чисел - нацело)

%	нахождение остатка
++	увеличение на 1
--	уменьшение на 1

Например: $\frac{5,6 - 11(0,12 + 4)}{6,11 + 3,2^2}$

на языке программирования записывается так: $(5.6 - 11 * (0.12 + 4)) / (6.11 + 3.2 * 3.2)$.

Упражнения.

1. Запишите на языке программирования

а) $56,5(11 - 5,8)$

в) $(4,6 - 1)^4 - 87,3$

б) $\frac{23 - 1,123(3,5 + 6)}{5^3 - 6}$

г) $1,1 + \frac{89^2}{5,6} + 7$

2. Запишите на языке алгебраических формул:

а) $567.5 * 4 - 5 * 5$;

в) $(4 - 6.78) / (23 * 5)$;

б) $33.7 * (23 - 4.5 * 4.5)$;

г) $5.45 - 24 / 67 + 4.5$.

3. Вычислите.

а) $23/5 - 8 * 3$;

в) $37\%5 - 2 + 6$;

б) $6 - 78 / (4 + 2)$;

г) $(56 - 23\%7) + 76/4$;

4. Составьте выражения, и запишите их на языке программирования:

а) Найти площадь и периметр прямоугольника если его длина — 5м, а ширина 20,45см.

б) Автомобиль движется со скоростью 5.4 м/с. Сколько времени ему понадобится, чтобы проехать 55 км.

в) Сторона куба 3,4см. Сколько метров составляет объем куба.

г) С полуночи прошло 5 ч 34 м 20 с. Сколько секунд осталось до полудня.

3 Определение переменных. Операция присваивания.

Для хранения информации используют переменные.

Переменная — это область памяти.

Прежде чем использовать переменную, ее нужно определить.

Общий вид определения переменной:

```
тип имя1;
```

Обозначения основных типов:

```
byte — байт.
```

```
int – целое число;  
long – длинное целое;  
float – действительное число;  
double – действительное число двойной точности.
```

Если нужно определить несколько переменных одного типа, то их имена* перечисляют через запятую.

Примеры:

```
int a,b;//две переменных для целых чисел.  
float x,y,radius;//три переменных для действительных чисел.  
char asc,ch,c;//три переменных для символов.
```

Основной способ записи значения в переменную — *операция присваивания*.

Синтаксис операции присваивания:

```
имя = выражение;
```

Например:

```
result = 2 * ( 56.4 - 4 );//в result будет записано 104.8
```

Найти площадь круга S и длину окружности C , если радиус 4,67см.

Переменные S и C должны быть действительного типа. Расчетные формулы $S = \pi \cdot r^2$; $C = 2 \cdot \pi \cdot r$, таким образом:

```
float s,c;  
s = 3.14 * 4.67 * 4.67;  
p = 2 * 3.14 * 4.67;
```

Упражнения.

5. Определите:

- а) две переменных для целых чисел;
- б) три переменных для действительных чисел и одну для символа;
- в) пять переменных для символов и две для действительных чисел;
- г) две для целых чисел, одну для действительного числа и три для символов.

6. Для хранения какой информации определены переменные:

* Имя переменной не может начинаться с цифры, в именах нельзя использовать пробел, знаки препинания, скобки и многие другие символы. Чтобы не ошибиться, используйте в именах только буквы латинского алфавита.

- a) int x,y; float a,b,c;
- б) char v,n,d; int d,b;
- в) float x,t,b; int l,m,n; char d,r,t,rr,yy;
- г) float x,y,z; int a,b,c,d; float f,l;

7. Что будет записано в переменные, и какого типа они должны быть:

- a) $tt = 6 * 7 - 11$;
- б) $result = 5 \% 2 + 7$;
- в) $y = 5,67 - 8/7$;
- г) $* r = 34 / 2 + 5$;

8. Составьте выражение, ведущее к решению задачи, запишите фрагмент программы с определением необходимых переменных:

- a) Найти скорость V , если расстояние 2345 км, пройдено за 10 ч. (движение равномерное и прямолинейное);
- б) Определить сколько минут MM содержится в 20 часах 125 секундах;
- в) Найдите остаток r и неполное частное q при делении числа 23465 на 234;
- г) За карандаш и тетрадь заплатили 15 рублей. Сколько стоит карандаш, если цена тетради составляет 70% стоимости покупки.

4 Класс. Свойства и методы.

Мир состоит из объектов. Для удобства объекты объединяют в *классы* по определенным признакам — *свойствам* и способностям выполнять какие либо действия — *методам*.

Например: животные, люди, машины, геометрические фигуры, ...

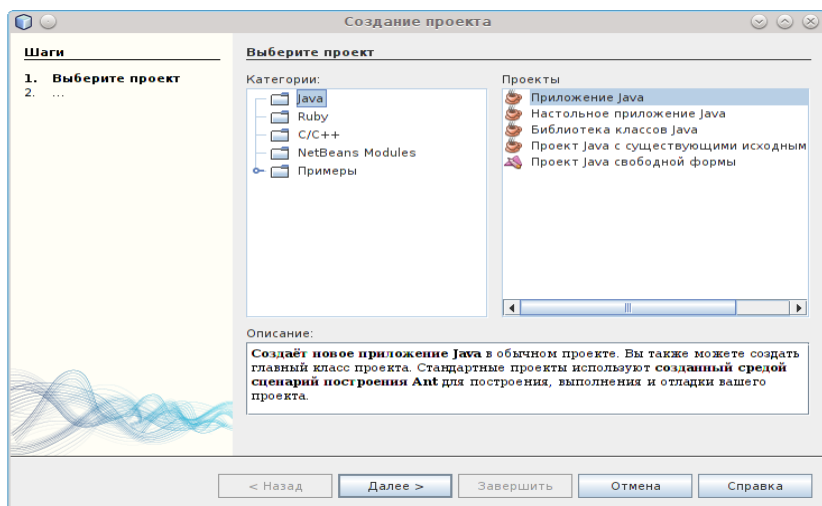
Объектно-ориентированное программирование сводится к описанию класса, объекты которого смогут решить поставленную задачу.

Существует множество языков ориентированных на ОПП. Вот самые популярные: C++, Java, C#, Python, Ruby и другие.

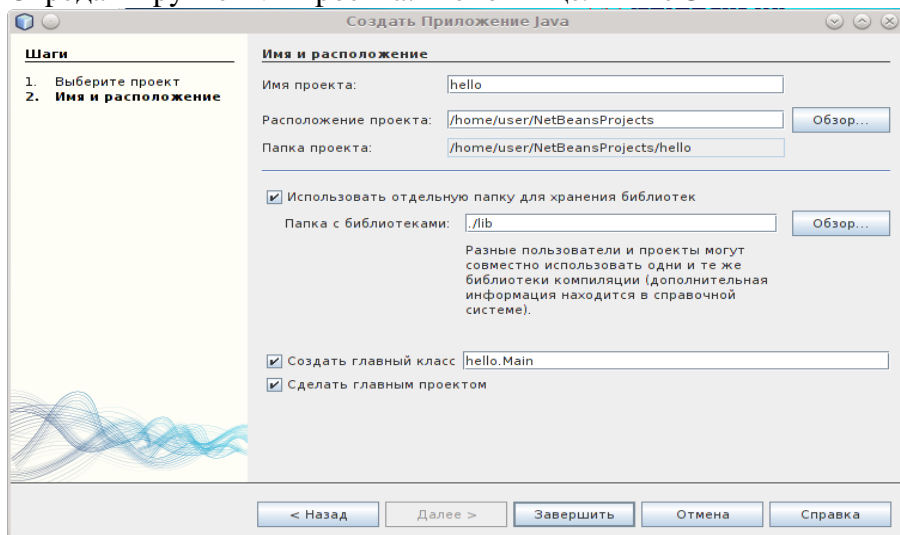
В этом пособии используется Java, который является самым популярным в настоящее время. На этом языке создаются приложения для всех популярных операционных систем, мобильных устройств. Для того, чтобы проверять примеры и решать задачи рекомендуется установить JRE/JDK + IDE NetBeans или Eclipse. Как это сделать рассказывается в приложении 1.

Создайте первое приложение на java в среде NetBeans:

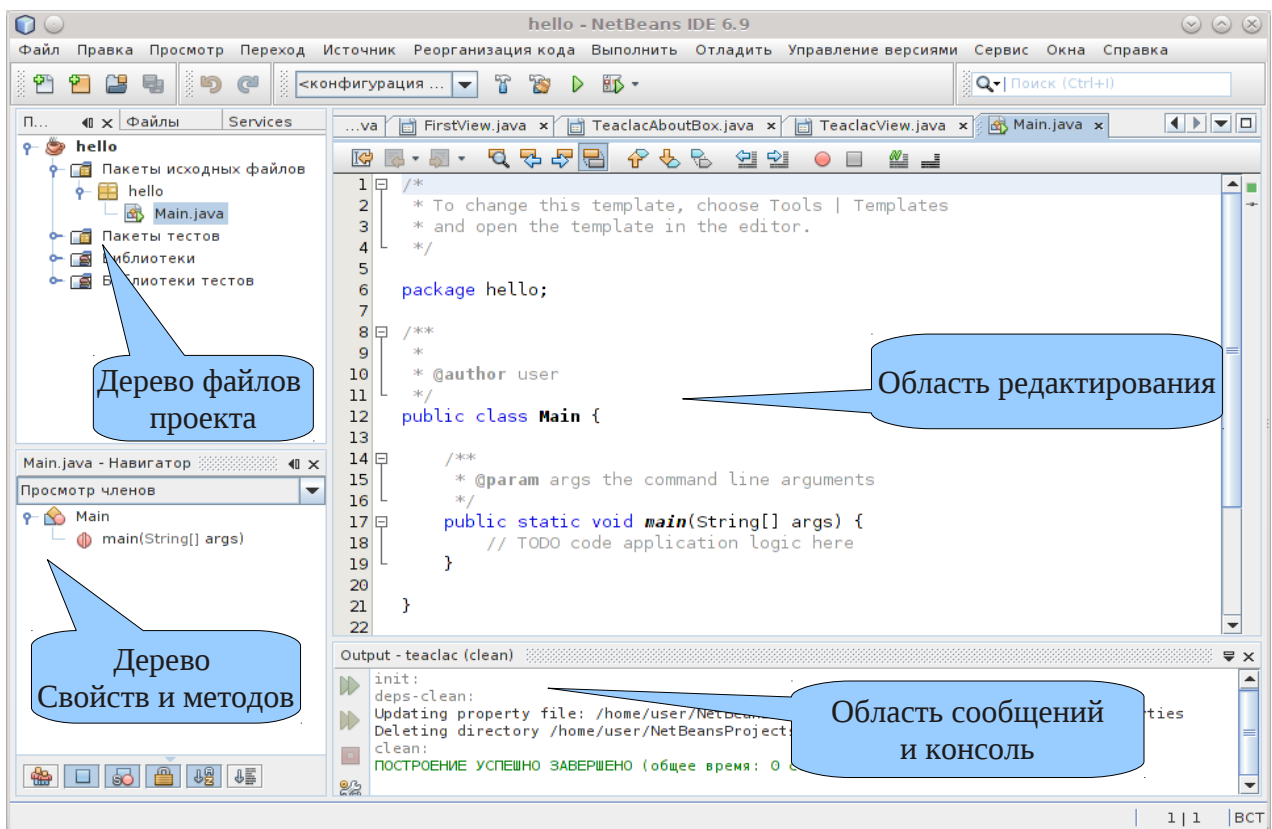
1. Запустите NetBeans
2. Выберите файл + создать проект. Категории Java, проекты Приложение Java



3. Отредактируйте имя проекта: "hello" и щелкните ЗАВЕРШИТЬ



4. Если все правильно, то вы увидите вот такое окно:



В области редактирования содержится определение класса Main:

```
public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }
}
```

В этом классе определен только один метод `main` да и то пустой.

TODO code application logic here (создайте код приложения здесь) указывает место где нужно написать главную часть программы. Именно с метода **main** начинается

выполнение любой программы Java.

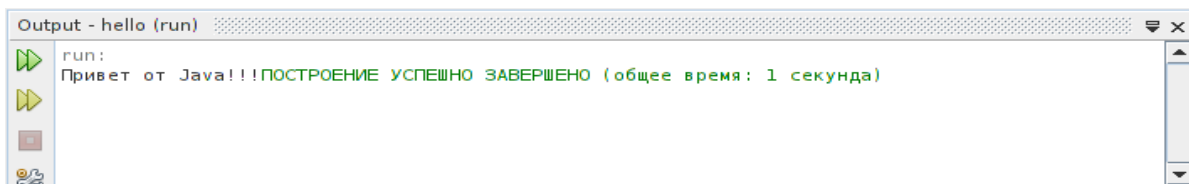
Смело удаляйте эту фразу и вместо нее пишите:

```
System.out.println("Привет от Java!!!");
```

После этого запустите приложение: кнопка



Если все правильно, то в области сообщений и консоли вы увидите:



Первое приложение готово.

"... вы должны суметь где-то создать текст программы, успешно его скомпилировать, загрузить, запустить на выполнение и разобраться, куда будет отправлен результат. Как только вы овладеете этим, все остальное окажется относительно просто".

(Б. Керниган, Д. Ритчи "Язык программирования Си").

Разберем подробно оператор который мы написали:

System — класс System

out — обращаемся к объекту out который является свойством класса System.

println("Привет от Java!!!") - вызываем метод println объекта out, которому передаем строку "Привет от Java!!!" которую он и выведет на консоль.

А, теперь более практически значимый пример.

Пример 1. Перевести температуру из градусов по Фаренгейту в градусы по Цельсию.

Известно, что для перевода необходимо из значения по Фаренгейту вычесть 32 и результат умножить на 5/9.

Будем изменять код метода main, поэтому остальное опускаем.

```
public static void main(String[] args) {  
    double f, c;  
    f=25;  
    c=(f-32)*(5/9);  
}
```

```
System.out.println("F="+f+" C="+c);  
}
```

Упражнения.

Напишите программы решения задач:

9. Вычислить значения выражений (вывод осуществить в виде формул):

а) $5a + 4b$

в) $4,6(x - y)^2 + 9$

б) $x^3 - 4x + y$

г) $\frac{4x - 7}{5}$

10. Найти площадь треугольника:

а) по стороне и соответствующей высоте;

в) Найти последнюю цифру целого числа.

б) по трем сторонам;

г) Найти гипотенузу зная катеты.

11. Найти количество теплоты необходимое для нагревания 5 литров воды от 20 до 80 градусов Цельсия.

5 Консольный ввод.

В примерах предыдущего пункта значения исходных данных задавались в самой программе. Это не удобно, потому что приходится каждый раз компилировать программу, а значит иметь компилятор java, знать язык Java, уметь компилировать Java-программы. Рядовому пользователю это совершенно ни к чему.

Поэтому нужно предоставить возможность вводить исходные данные в процессе выполнения программы.

Организация консольного ввода на языке Java не простая задача.

Первое что нужно сделать — подключить пакеты util делается это командой компилятору:
`import java.util.*;`

Она должна быть записана перед методом main.

Затем создать объект класса Scanner связав его с потоком ввода.

После этого вызвать один из нужных методов:

`next()` - возвращает строку

`nextInt()` - возвращает очередное целое

`nextLong()` - возвращает очередное целое двойной точности

`nextFloat()` - возвращает очередное действительное

`nextDouble()` - возвращает очередное действительное двойной точности.

Пример. Найти площадь прямоугольника по его ширине и высоте.

```
import java.util.*;
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    double a,b,s;
    System.out.println("Введите длину и ширину через пробел:");
    a=sc.nextDouble();
    b=sc.nextDouble();
    s=a*b;
    System.out.println("Площадь равна:"+s);
}
```

Упражнения.

12. Напишите программы, организовав динамический ввод:

- | | |
|---|---------------------------------|
| а) Конвертации долларов в рубли по
текущему курсу. | в) Перевода миль в километры. |
| б) Перевода километров в версты. | г) Перевода футов в сантиметры. |

6 Управляющие операторы.

Для того, чтобы указать исполнителю на необходимость принятия решения о выполнении какого-либо действия или его повторения служат *управляющие операторы*.

К основным управляющим операторам относятся:

1. Условный оператор.
2. Оператор выбора.
3. Оператор цикла.

В управляющих операторах используют *логические выражения*.

6.1 Логические выражения.

Логическое выражение состоит из *высказываний* и логических операций над ними.

Любое элементарное высказывание или логическое выражение может принимать одно из двух значений: *истина* или *ложь*. С точки зрения программиста — 1 или 0 соответственно.

Самые распространенные высказывания - это *равенства и неравенства, их системы и совокупности*.

Например:

$5 > 6$, $7 < 90$, $2 * 3 < 6$, $45 = 45$

$$\begin{cases} 5 > x \\ y \geq 12 \end{cases}$$

$$\begin{cases} 3x < 4 \\ m \leq 2 \end{cases}$$

Основные логические операции и их *таблицы истинности*.

Каждая строка *таблицы истинности* показывает результат логической операции при указанных значения *операндов*.

1. Операция "не"

A	!A
1	0
0	1

2. Операция "и"

A	B	A & B
0	0	0
1	0	0
0	1	0
1	1	1

3. Операция "или"

A	B	A B
0	0	0
1	0	1
0	1	1
1	1	1

Примеры вычисления значений логических выражений:

$$!(4 > 5) = !0 = 1$$

$$!((45 > 6) \&\& (9 < 11)) = !(1 \&\& 1) = !1 = 0$$

Логические выражения с *переменными*.

Если в равенстве или неравенстве один из операндов — *переменная*, то значение выражения нельзя найти до тех пор, пока не задано значение этой *переменной*.

Например:

$$(x > 2) \parallel (4 < x) = ?$$

при $x=5$, $(5 > 2) \parallel (4 < 5) = 1 \parallel 1 = 1$

Контрольные вопросы.

1. Что такое логическое выражение?
2. Что, чаще всего, выступает в роли высказывания? Назовите основные логические операции.
3. Запишите, по памяти, таблицы истинности основных логических операций.

Упражнения.

13. Запишите логические выражения на языке Java:

а)
$$\begin{cases} 5 - x \geq 1 \\ x < 0 \end{cases}$$

б)
$$\begin{cases} 6x - 1 > 7 \\ x < 5 \end{cases}$$

в)
$$\begin{cases} 4,5a - 1 \leq 4 \\ 6x > 11 \\ y < 56 \end{cases}$$

г)
$$\begin{cases} x^2 - 1 \geq 1 \\ \begin{cases} y > 2x \\ x < 2 \end{cases} \end{cases}$$

14. Найдите значение логического выражения, при данных значениях переменных.

- а) $!(5x > 11)$, при $x = 1; 34$;
б) $(y + x < 45) \&\& !(y > x)$, при $x = 2; y = 17$;
в) $!((d > 2) \&\& (d + 1 < 6))$, при $d = 1; 5; 0$;
г) $(a + b + c \leq 0) \parallel !(a + b > c)$, при $a = -1; b = 2; -3$.

15. Петя и Коля выше Васи, но Коля ниже Пети. Каково значение логического выражения: $(K > V) \parallel (V > P)$, где первой буквой имени обозначен рост.

6.2 Условный оператор.

Перед вычислением значения выражения $\frac{1}{x}$ нужно убедиться в том, что значение x не равно 0, иначе вычислить нельзя. И таких ситуаций на практике очень много.

Чтобы указать исполнителю на необходимость проверки условия существует *условный оператор*.

Синтаксис условного оператора:



Пример:

Найти значение выражения $\frac{5a-b}{a-b}$.

Очевидно, что значение выражения можно вычислить только при условии, что $a-b \neq 0$.

```
import java.util.*;
public static void main(String[]args){
    Scanner sc = new Scanner(System.in);
    double a,b,res;
    a=sc.nextDouble(); b=sc.nextDouble();

    if ((a - b)!=0)
    {
        res = (5*a-b)/(a-b);
        System.out.println("результат:"+res);
    }
    else
        System.out.println("Деление на 0!!!\n");

    sc.close();
}
```

Бывает, что в качестве оператора1 или оператора2 необходимо выполнить несколько операторов, то они берутся в операторные скобки:

```
{ - скобка отрывается
} - скобка закрывается
```

Такая конструкция называется *блок*.

Если при неверном условии требуется просто перейти к выполнению операторов после условного, то else часть можно упустить. В этом случае имеет место *сокращенный условный оператор*.

Например, найти наименьшее среди трех целых чисел.

```
Import java.util.*
public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    int a,b,c,min;
    a=sc.nextInt();b=sc.nextInt();c=sc.nextInt();
```

```

min=a;
if (min>b) min=b;
if(min>c) min = c;
System.out.println("min="+min);
sc.close();
}

```

Для организации вычислений полезно использовать класс математических функций *Math*:

метод	Результат.
sqrt(x)	$\sqrt{(x)}$
sin(x)	sin(x)
cos(x)	cos(x)
tan(x)	tg(x)
exp(x)	e^x
abs(x)	x

Например, выражение $\frac{\sin x}{\sqrt{x+4}}$ будет записано так: Math.sin(x)/Math.sqrt(x+4).

Контрольные вопросы.

1. Запишите общий вид сокращенного и полного условного оператора.
 2. Объясните: как компьютер выполняет условный оператор?
 3. Перечислите основные математические функции.
-

Упражнения.

16. Что будет храниться в переменной x после выполнения кода:

- а) x=7; if(x>3) {x--;} else {x=x+1;}
 б) x=34;x=x+5; if(x<=40) {x=0;} else {x++;}
 в) x=11;y=5; if(x>2*y) {x=x+y;} else {x=3;}
 г) * x=5; (if(x++>=6){x=0;}

17. Найти значение выражения:

- а) $\frac{r-m}{r+1}$
 б) $\sqrt{4x-5}$

в) $\frac{2}{x} - \sqrt{x-1}$

г) $r + \sqrt{\frac{4x-2}{x}}$

18. Найти наибольшее среди четырех чисел.

19. Определить принадлежит ли точка координатной прямой, заданному интервалу.

а) $[a;b]$

в) $(a;b]$

б) $[a;b)$

г) $(a;b)$

20. Делится ли данное число на 2?

21. Какое из трех данных чисел лежит между двумя другими.

22. Принадлежит ли точка $(x;y)$ графику функции $y=2x^2-5$

6.3 Оператор выбора

Иногда необходимо выполнять операторы в зависимости от нескольких различных значений выражения. В этом случае удобнее использовать *оператор выбора*.

Синтаксис оператора:

```
switch (выражение)
{
    case зн1: оператор 1;
        break;
    case зн2: оператор 2;
        break;
    ...
    case знn: оператор n;
        break;
    default: оператор;
}
```

Пример: Вывести название целого числа из отрезка $[1;5]$.

```
import java.util.*;
public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    int n;
    System.out.println("введите целое число:");
    sc.nextInt();
    switch(n)
    {
        case 1: System.out.println("один\n");
            break;
        case 2: System.out.println("два\n");
            break;
    }
```



```

        case 3: System.out.println("три\n");
            break;
        case 4: System.out.println("четыре\n");
            break;
        case 5: System.out.println("пять\n");
            break;
        default: System.out.println("Не знакомое
число!!!");
    }
    sc.close();
}

```

6.4 Циклы.

Цикл — это управляющая конструкция, организующая повторение выполнения операторов, пока верно условие.

Различают циклы *с предусловием* и *постусловием*.

Чаще всего используют циклы с предусловием.

6.4.1 Цикл с предусловием.

Основной оператор:

```
while (условие) оператор;
```

Пример:

Вывести на консоль все числа последовательности Фибоначчи, не превосходящие заданное число.

Последовательность Фибоначчи:

1 1 2 3 5 7 12 ...

Начиная с третьего каждый элемент равен сумме двух предыдущих.

```

import java.util.*;
public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    int a,b,c,n;
    a=b=1;
    System.out.println("Введите целое число:");
    n=sc.nextInt();
}

```

```

while ( b<n)
{
    c=a+b;
    a=b;
    b=c;
}
System.out.println("Искомое число:"+a);
sc.close();
}

```

Оператор for:

for(инициализация; условия; изменение)оператор;

Пример:

Вывести на экран n первых неотрицательных чисел кратных трем.

Последовательность: 0 3 6 9 ...

```

import java.util.*;
public static void main(int argc,char **argv)
{
    Scanner sc = new Scanner(System.in);
    int i,n;
    System.out.print("Введите n:");
    n=sc.nextInt();
    for(i=0 ; i<n; i++) System.out.print(" ",3*i);
    System.out.println();
    sc.close();
}

```

Упражнения.

23. Найти значение выражения

а) $1*2*3*...*n$;

в) $x+2*x+3*x+...+n*x$;

б) $1+2+3+...+n$;

г) $\frac{1}{2x} + \frac{2}{3x} + \dots + \frac{n}{(n+1)x}$

24. Вывести на консоль первые n нечетных чисел.

25. Вывести на консоль значения функции из данного интервала, с указанным шагом:

а) $y=3x-1$ $[1;5]$, шаг=1

б) $y=2\sin(x)$ $(-\pi;\pi)$, шаг=0,1

26. Вывести все цифры целого числа.

27. Записать цифры числа в обратном порядке.

28. Найти сумму первых n элементов последовательности, если $a_i = \begin{cases} 2i-1 & i \text{ четное} \\ i-2 & i \text{ нечетное} \end{cases}$

29. Для последовательности Фибоначчи найти

а) сумму элементов не превосходящих

в) сумму первых n элементов

данного целого положительного числа;

г) количество четных элементов меньших

б) первые n элементов;

данного целого положительного числа.

6.4.2 Цикл с постусловием.

Синтаксис оператора:

```
do оператор while(условие);
```

Пример: Добиться ввода четного числа

```
import java.util.*;
public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    int a;
    do
    {
        System.out.print("Введите четное число!");
        a = sc.nextInt();
    }
    while (a%2 !=0);
    System.out.println("Наконец-то!");
    sc.close();
}
```

Упражнения.

30. Добиться ввода задуманного числа.

31. Трехзначное число — пароль. Добиться ввода пароля не более чем с 4 попыток.

32. *Реализовать игру "Угадай число".

7 Массивы.

Для хранения многих однотипных данных удобно использовать составную переменную — массив.

Массив — это последовательность однотипных элементарных переменных.

Вся последовательность имеет имя, а конкретная элементарная переменная имеет номер.

В языке Java нумерация элементов массива ведется с 0.

Пример:

массив **a[11]** из 11 целых чисел

0	1	2	3	4	5	6	7	8	9	10
345	12	56	98	11	5	98	4	11	8	6

a[7]==4

Определение массива

```
тип_элементов имя[];
```

Выделение памяти под массив:

```
имя = тип_элементов[количество];
```

Например:

```
int a[];
a=new int[50]; //массив из 50-ти целочисленных переменных

float x[];
x = new float[10]; //массив из 10-ти действительных переменных1
```

Виды задач на массивы:

1. Заполнение массива;
2. Обработка массива.

7.1 Заполнение массивов.

По формуле n -го элемента.

Заполнить массив из 12 целых чисел первыми четными. (формула четного числа $2*n$, где n — номер четного числа)

```
public static void main(String[] args)
{
    int a[]=new int[12];
    int i;
    String s;
    s="";
    for(i=0;i<12;i++) a[i]=2*i;
    for(i=0;i<12;i++) s=s+" "+a[i];
    System.out.println(s);
}
```

Обратите внимание, вместо многократного вызова метода println в цикле формируется строка. А метод println вызывается один раз. По скорости работы такой подход предпочтительнее.

По рекуррентному соотношению

Рекуррентное соотношение — это зависимость очередного элемента от предыдущих.

Заполнить массив из 8 действительных чисел, если $a[0]=1,1$ $a[i]=2*a[i-1]+1$

```
public static void main(String[] args)
{
    double a[]=new double[8];
    int i;
    String s;
    s="";
    a[0]=1.1;
    for(i=1;i<8;i++) a[i]=2*a[i-1]+1;
    for(i=0;i<8;i++) s=s+" "+a[i];
    System.out.println(s);
}
```

Из потока ввода.

Пример

Заполнить массив `int a[5]` со стандартного потока ввода (клавиатура).

```
import java.util.*
public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    int a[]=new int[5];
    int i;
    String s;
    s="";
    a[0]=1.1;
    for(i=0;i<5;i++) a[i]=sc.nextInt();
    for(i=0;i<5;i++) s=s+" "+a[i];
    System.out.println(s);
}
```

Случайными числами.

Для получения случайного числа служит класс `Random`. Он содержит метод

`nextInt(int n)` - случайное целое число из `[0;n]`.

Пример.

Заполнить массив `int a[10]` случайными числами из отрезка `[10;99]`

```
public static void main(String[] args)
{
    int a[]=new int[10];
    int i;
    Random rand =new Random();
    String s;
    s="";
```

```

    for(i=0;i<10;i++) a[i]=rand.nextInt(90)+9;
    for(i=0;i<10;i++) s=s+" "+a[i];
    System.out.println(s);
}

```

Упражнения.

33. Заполнить массив:

а) int a[17] числами кратными 5;

б) int b[9] первыми элементами последовательности Фибоначчи.

в) float x[7] с клавиатуры.

г) int m[10] случайными трехзначными числами.

34. Заполнить массив int a[20] элементами последовательности $a_i = \begin{cases} i+1 & i < 3 \\ 2*a_{i-1} & i \geq 3 \end{cases}$

35. Заполнить массив int a[14] элементами последовательности:

а) 1 0 1 0 1 0 ...

в) 1 2 3 2 3 4 3 4 5 ...

б) 1 2 3 1 2 3 1 2 3 ...

г) * 1 0 1 1 0 1 1 1 ...

7.2 Обработка массива

Поиск элементов.

Пример

Найти минимальный элемент в массиве двузначных чисел int a[12]

```

public static void main(String[] args)
{
    int a[]=new int[12];
    int i,min;
    Random rand =new Random();
    String s;
    s="";
    for(i=0;i<12;i++) a[i]=rand.nextInt(90)+9;
    for(i=0;i<12;i++) s=s+" "+a[i];
    System.out.println(s);
    min=a[0];
    for(i=1;i<12;i++) {
        if(a[i]<min) min =a[i];
    }
    System.out.println("min="+min);
}

```

```
}
```

Поиск и замена.

Пример.

В массиве int a[19] трехзначных чисел заменить все четные числа нулями.

```
public static void main(String[] args)
{
    int a[]=new int[19];
    int i;
    Random rand =new Random();
    String s;
    s="";
    for(i=0;i<19;i++) a[i]=rand.nextInt(900)+99;
    for(i=0;i<19;i++) s=s+" "+a[i];
    System.out.println(s);

    for(i=1;i<19;i++) {
        if((a[i]%2)==0) a[i] = 0;
    }
    s=""
    for(i=0;i<19;i++) s=s+" "+a[i];
    System.out.println(s);
}
```

Изменение порядка элементов.

Пример.

Упорядочить массив целых неотрицательных чисел меньших 50 int a[10] в порядке возрастания.

```
public static void main(String[] args){
    int a[]=new int[10];
    int i,j,b;
    Random rand =new Random();
    String s;
```



```

s="";
for(i=0;i<10;i++) a[i]=rand.nextInt(50);
for(i=0;i<10;i++) s=s+" "+a[i];
System.out.println(s);
for(i=0;i<10;i++)
    for(j=9;j>i;j--)
        if ( a[j]<a[j-1] ) {b=a[j];a[j]=a[j-1];a[j-1]=b;}

s="";
for(i=0;i<10;i++) s=s+" "+a[i];
System.out.println(s);
}

```

Упражнения.

36. Дан массив двузначных чисел int a[10]. Найти

- а) Номера четных элементов;
- б) Элементы большие 50;
- в) Элементы, оканчивающиеся на 5.
- г) Элементы, сумма цифр которых больше 10.

37. Дан массив трехзначных чисел int a[15]. Определить какие из них не могут входить в ip адрес и заменить их нулями.

38. В массиве целых неотрицательных чисел меньших 20 int a[30] найти:

- а) минимальный.
- б) максимальный
- в) сумму минимального и максимального.
- г) * первый элемент больший минимального.

8 Приложения.

8.1 Необходимое программное обеспечение.

Для работы наиболее подходит IDE от разработчика: NetBeans. Все необходимое можно скачать отсюда: <http://www.oracle.com/technetwork/articles/javase/index-jsp-138363.html>.

Скачиваете, запускаете установку и через некоторое время получаете полноценную IDE от разработчика.

Еще возможно понадобится установить JRE. Ее также можно скачать с сайта разработчика: