



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ

*Институт Принтмедиа и информационных технологий  
Кафедра Информатики и информационных технологий*

направление подготовки

09.03.02 «Информационные системы и технологии»

## ЛАБОРАТОРНАЯ РАБОТА № 11

Дисциплина: Введение в программирование.

Выполнил(а):

студент(ка) группы 191-726

Синельникова К.Т.

\_\_\_\_\_  
(Дата)

\_\_\_\_\_  
(Подпись)

Проверил: асс. Кононенко К.М.

\_\_\_\_\_  
(Дата)

\_\_\_\_\_  
(Подпись)

Замечания: \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_

Москва

2019

## Оглавление

<b>Теория .....</b>	<b>3</b>
<b>Задания .....</b>	<b>6</b>
<b>Блок-схемы.....</b>	<b>7</b>
<b>Код программы.....</b>	<b>13</b>
<b>Результат программы.....</b>	<b>16</b>

## Теория

### Оператор for

Оператор for выполняет оператор или блок операторов, пока определенное логическое выражение равно значению true.

В любой момент в блоке операторов for вы можете прервать цикл с помощью оператора break или перейти к следующей итерации в цикле с помощью оператора continue. Также можно выйти из цикла for с помощью операторов goto, return или throw.

### Структура оператора for

Оператор for определяет разделы *инициализатора*, *условия* и *итератора*.

Все три раздела добавляются по желанию. Тело цикла является оператором или блоком операторов.

#### Раздел инициализатора

Операторы в разделе *инициализатора* выполняются только один раз перед входом в цикл. Раздел *инициализатора* представляет собой один из следующих объектов:

- Объявление и инициализация локальной переменной цикла, к которой невозможно получить доступ вне цикла.
- Ноль или более выражений операторов из следующего списка, разделенные запятыми:
  - оператор присваивания
  - вызов метода
  - префиксное или постфиксное выражение приращения, такое как ++i или i++
  - префиксное или постфиксное выражение декремента, такое как --i или i--
  - создание объекта с помощью оператора new
  - выражение await

## Раздел условия

Раздел *условия*, если он определен, должен быть логическим выражением. Это выражение оценивается перед каждой итерацией цикла. Если раздел *условия* отсутствует или логическое выражение имеет значение true, выполняется следующая итерация цикла. В противном случае выполняется выход из цикла.

## Раздел итератора

Раздел *итератора* определяет, что происходит после каждой итерации тела цикла. Раздел *итератора* содержит ноль или более следующих выражений оператора, разделенных запятыми:

- оператор присваивания
  - вызов метода
  - префиксное или постфиксное выражение приращения, такое как ++i или i++
  - префиксное или постфиксное выражение декремента, такое как --i или i--
  - создание объекта с помощью оператора new
- выражение await

## Оператор while

Оператор while выполняет оператор или блок операторов, пока определенное логическое выражение равно значению true. Так как это выражение оценивается перед каждым выполнением цикла, цикл while выполняется ноль или несколько раз. Это отличает его от цикла do, который выполняется от одного до нескольких раз.

В любой точке блока операторов while можно разорвать цикл с помощью оператора break.

Можно перейти непосредственно к оценке выражения while, воспользовавшись оператором continue. Если значение выражения оценивается

как `true`, выполнение продолжается с первого оператора цикла. В противном случае выполнение продолжается с первого оператора после цикла.

Также можно выйти из цикла `while` с помощью операторов `goto`, `return` или `throw`.

### Задания

1. Даны целые положительные числа  $A$  и  $B$  ( $A < B$ ). Вывести все целые числа от  $A$  до  $B$  включительно; при этом каждое число должно выводиться столько раз, каково его значение (например, число 3 выводится 3 раза).
2. Даны положительные числа  $A$  и  $B$  ( $A > B$ ). На отрезке длины  $A$  размещено максимально возможное количество отрезков длины  $B$  (без наложений). Не используя операции умножения и деления, найти длину незанятой части отрезка  $A$ .
3. Дано целое число  $N$  ( $> 1$ ). Вывести наименьшее из целых чисел  $K$ , для которых сумма  $1 + 2 + \dots + K$  будет больше или равна  $N$ , и саму эту сумму.
4. Начальный вклад в банке равен 1000 руб. Через каждый месяц размер вклада увеличивается на  $P$  процентов от имеющейся суммы ( $P$  — вещественное число,  $0 < P < 25$ ). По данному  $P$  определить, через сколько месяцев размер вклада превысит 1100 руб., и вывести найденное количество месяцев  $K$  (целое число) и итоговый размер вклада  $S$  (вещественное число).
5. Даны целые положительные числа  $A$  и  $B$ . Найти их наибольший общий делитель (НОД), используя алгоритм Евклида.
6. Дано целое число  $N$  ( $> 1$ ), являющееся числом Фибоначчи:  $N = F_K$ . Найти целое число  $K$  — порядковый номер числа Фибоначчи  $N$ .

## Блок-схемы

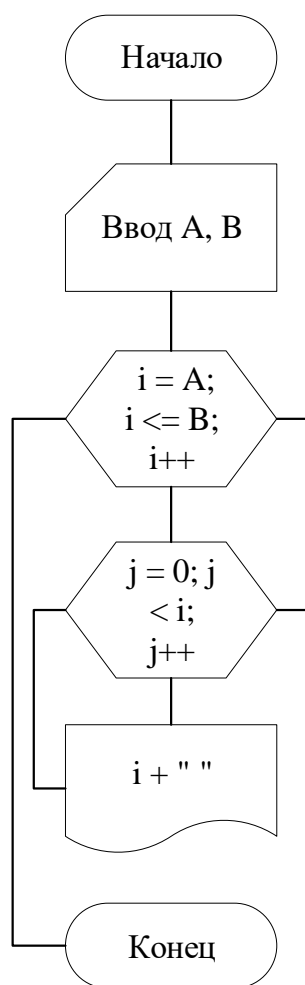


Рисунок 1 — Блок-схема к заданию 1

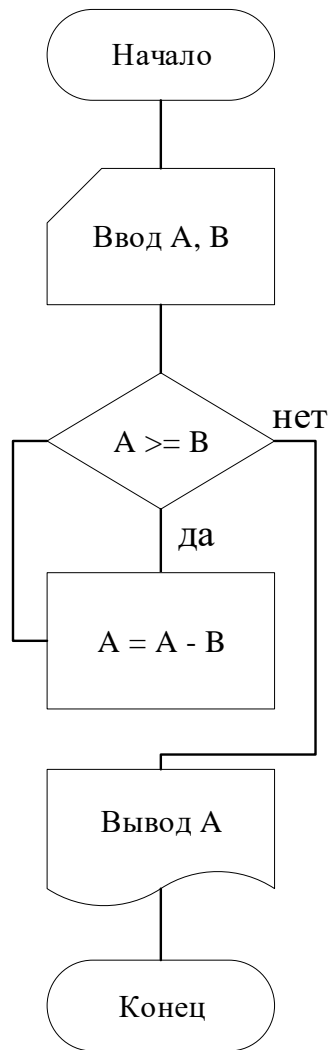


Рисунок 2 — Блок-схема к заданию 2



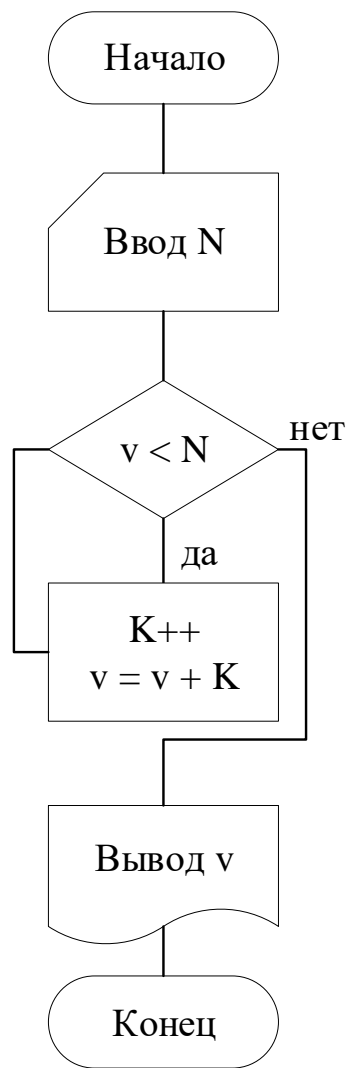


Рисунок 3 — Блок-схема к заданию 3

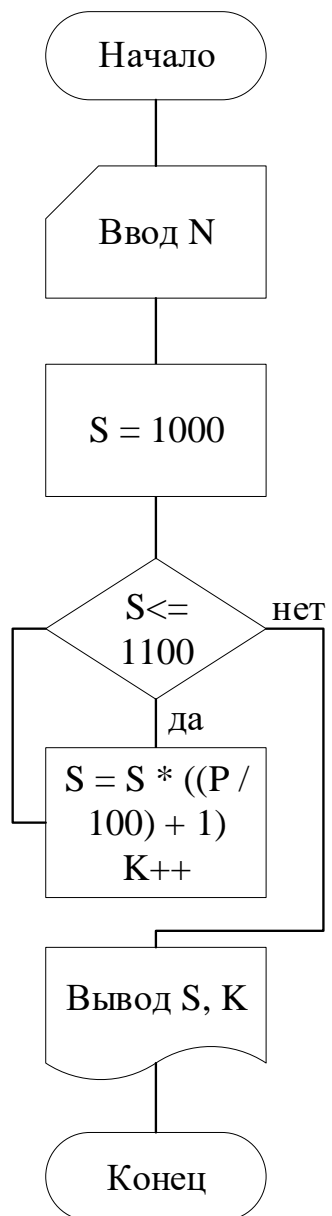


Рисунок 4 — Блок-схема к заданию 4

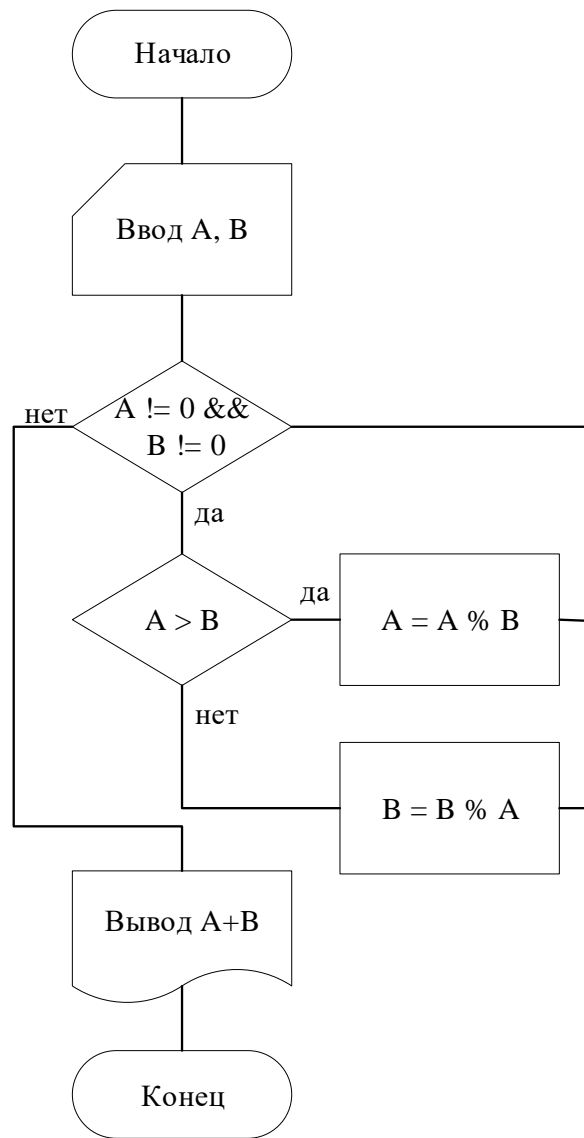


Рисунок 5 — Блок-схема к заданию 5

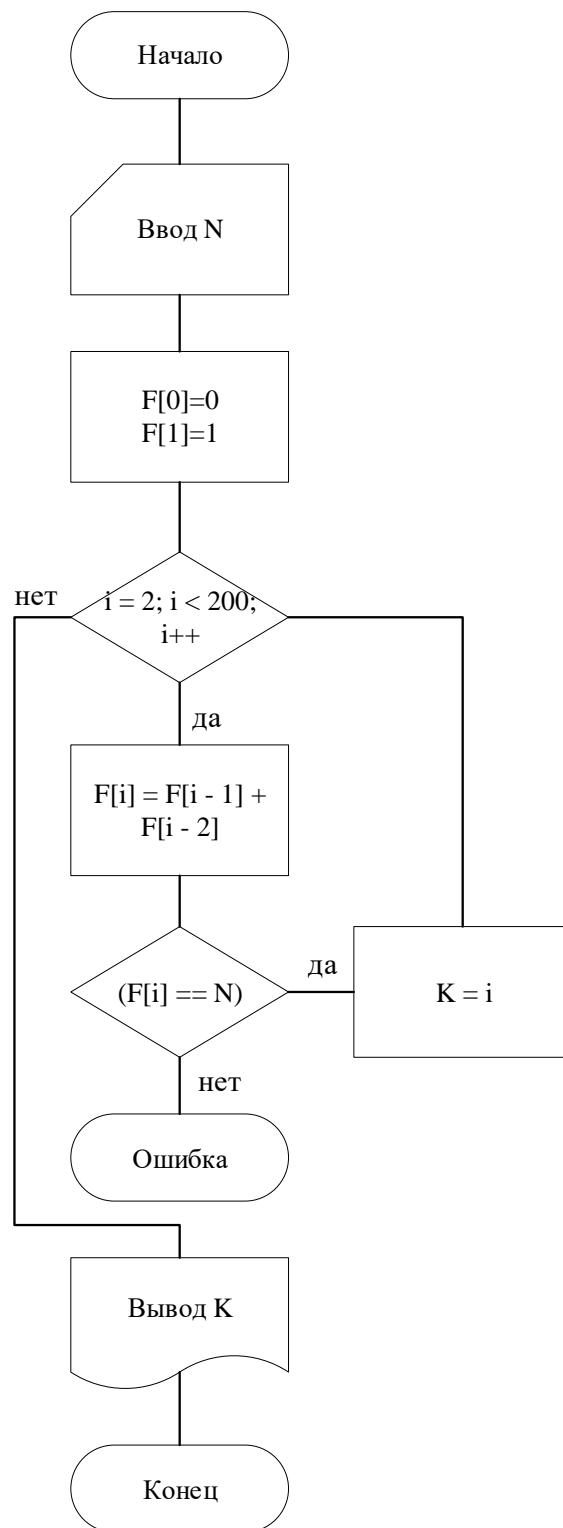


Рисунок 6 — Блок-схема к заданию 6

## Код программы

### Листинг 1 — Задание 1 (Вывести все целые от А до В)

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6. namespace Код_Лабораторной_11__1_6_
7. {
8.     class Program
9.     {
10.         static void Main(string[] args)
11.         {
12.             int A, B;
13.             Console.WriteLine("Обязательное условие A<B");
14.             Console.Write("Введите число A: ");
15.             A = int.Parse(Console.ReadLine());
16.             Console.Write("Введите число B: ");
17.             B = int.Parse(Console.ReadLine());
18.             for (int i = A; i <= B; i++)
19.             {
20.                 for (int j = 0; j < i; j++)
21.                 {
22.                     Console.Write(i + " ");
23.                 }
24.             }
25.             Console.ReadKey();
26.         }
27.     }
28. }
```

### Листинг 2 — Задание 2 (Найти длину незанятой части А)

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6. namespace Код_Лабораторной_11__1_6_
7. {
8.     class Program
9.     {
10.         static void Main(string[] args)
11.         {
12.             int A, B;
13.             Console.WriteLine("Обязательное условие A>B");
14.             Console.Write("Введите длину отрезка A: ");
15.             A = int.Parse(Console.ReadLine());
16.             Console.Write("Введите длину отрезка B: ");
17.             B = int.Parse(Console.ReadLine());
18.             while (A >= B)
19.             {
20.                 A = A - B;
21.             }
22.             Console.Write("Незанятая часть отрезка: " + A);
23.             Console.ReadKey();
24.         }
25.     }
26. }
```

### Листинг 3 — Задание 3 (Вывести наименьшее из целых чисел K)

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6. namespace Код_Лабораторной_11__1_6_
7. {
8.     class Program
9.     {
10.         static void Main(string[] args)
11.         {
12.             int v, K, N;
13.             Console.Write("Введите число N: ");
14.             N = int.Parse(Console.ReadLine());
15.             K = 0;
16.             v = 0;
17.             while (v < N)
18.             {
19.                 K++;
20.                 v = v + K;
21.             }
22.             Console.Write("Сумма: " + v);
23.             Console.ReadKey();
24.         }
25.     }
26. }
```

### Листинг 4 — Задание 4 (Вывести количество месяцев и сумму)

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6. namespace Код_Лабораторной_11__1_6_
7. {
8.     class Program
9.     {
10.         static void Main(string[] args)
11.         {
12.             double P, K, S;
13.             Console.Write("Введите проценты: ");
14.             P = int.Parse(Console.ReadLine());
15.             S = 1000;
16.             K = 0;
17.             while (S <= 1100)
18.             {
19.                 S = S * ((P / 100) + 1);
20.                 K++;
21.             }
22.             Console.WriteLine("Сумма: " + S);
23.             Console.WriteLine("Через сколько месяцев: " + K);
24.             Console.ReadKey();
25.         }
26.     }
27. }
```


## Листинг 5 — Задание 5 (Найти НОД через алгоритм Евклида)

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6. namespace Код_Лабораторной_11__1_6_
7. {
8.     class Program
9.     {
10.         static void Main(string[] args)
11.         {
12.             int A, B;
13.             Console.Write("Введите число A: ");
14.             A = int.Parse(Console.ReadLine());
15.             Console.Write("Введите число B: ");
16.             B = int.Parse(Console.ReadLine());
17.             while (A != 0 && B != 0)
18.             {
19.                 if (A > B) A = A % B;
20.                 else B = B % A;
21.             }
22.             Console.Write("Наибольший общий делитель: " + (A+B));
23.             Console.ReadKey();
24.         }
25.     }
26. }
```

## Листинг 6 — Задание 6 (Найти порядковый номер числа Фибоначчи)


```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6. namespace Код_Лабораторной_11__1_6_
7. {
8.     class Program
9.     {
10.         static void Main(string[] args)
11.         {
12.             int N, K;
13.             int[] F = new int[200];
14.             Console.Write("Введите число Фибонначчи: ");
15.             N = int.Parse(Console.ReadLine());
16.             K = 0;
17.             F[0] = 0;
18.             F[1] = 1;
19.             for (int i = 2; i < 200; i++)
20.             {
21.                 F[i] = F[i - 1] + F[i - 2];
22.                 if (F[i] == N) K = i;
23.             }
24.             Console.Write("Номер числа Фибонначчи: " + K);
25.             Console.ReadKey();
26.         }
27.     }
28. }
```

## Результат программы




```
C:\Лабы С#\Лаб.раб. 11\Код Лабораторной 11 (1-6)
Обязательное условие A<B
Введите число A: 2
Введите число B: 5
2 2 3 3 3 4 4 4 4 5 5 5 5 5
```

Рисунок 7 — Результат выполнения программы 1




```
C:\Лабы С#\Лаб.раб. 11\Код Лабораторной 11 (1-6)
Обязательное условие A>B
Введите длину отрезка A: 9
Введите длину отрезка B: 4
Незанятая часть отрезка: 1
```

Рисунок 8 — Результат выполнения программы 2




```
C:\Лабы С#\Лаб.раб. 11\Код Лабораторной 11 (1-6)
Введите число N: 4
Сумма: 6
```

Рисунок 9 — Результат выполнения программы 3




```
C:\Лабы С#\Лаб.раб. 11\Код Лабораторной 11 (1-6)
Введите проценты: 10
Сумма: 1210
Через сколько месяцев: 2
```

Рисунок 10 — Результат выполнения программы 4



```
C:\Лабы С#\Лаб.раб. 11\Код Лабораторной 11 (1-6)
Введите число A: 48
Введите число B: 32
Наибольший общий делитель: 16
```

Рисунок 11 — Результат выполнения программы 5



```
C:\Лабы С#\Лаб.раб. 11\Код Лабораторной 11 (1-6)
Введите число фибонначчи: 13
Номер числа фибонначчи: 7
```

Рисунок 12 — Результат выполнения программы 6