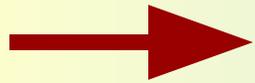
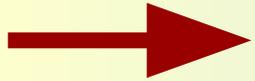


# **Лекция 1. Этапы разработки программы.**



Человек вводит в компьютер исходные данные (например, с клавиатуры).



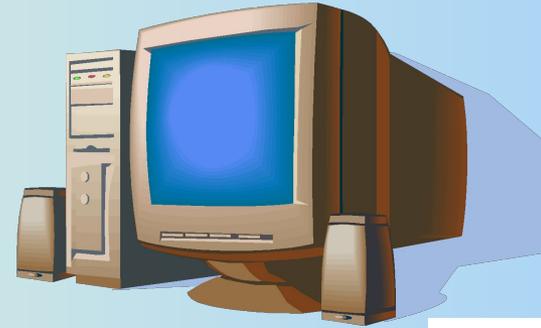


Процессор преобразует исходные данные в результат по определенному алгоритму.

```
c:\ c:\documents and settings\user\my documents\test.txt
2 2 2
2 5 6
8 0 6
Minimum=0
Maximum=8
```

→ Компьютер выводит результат на:

Экран



принтер



в файл



**Алгоритм, записанный на  
специальном языке,  
называется программой.**

```
// Halo.cpp : Defines the entry point for the console application.
//
#include "stdafx.h"
#include <conio.h>
#include <iostream>

using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    cout<<"Hallo!!!"<<endl;
    getch();
    return 0;
}
```

**Исходный текст**



**Результаты**

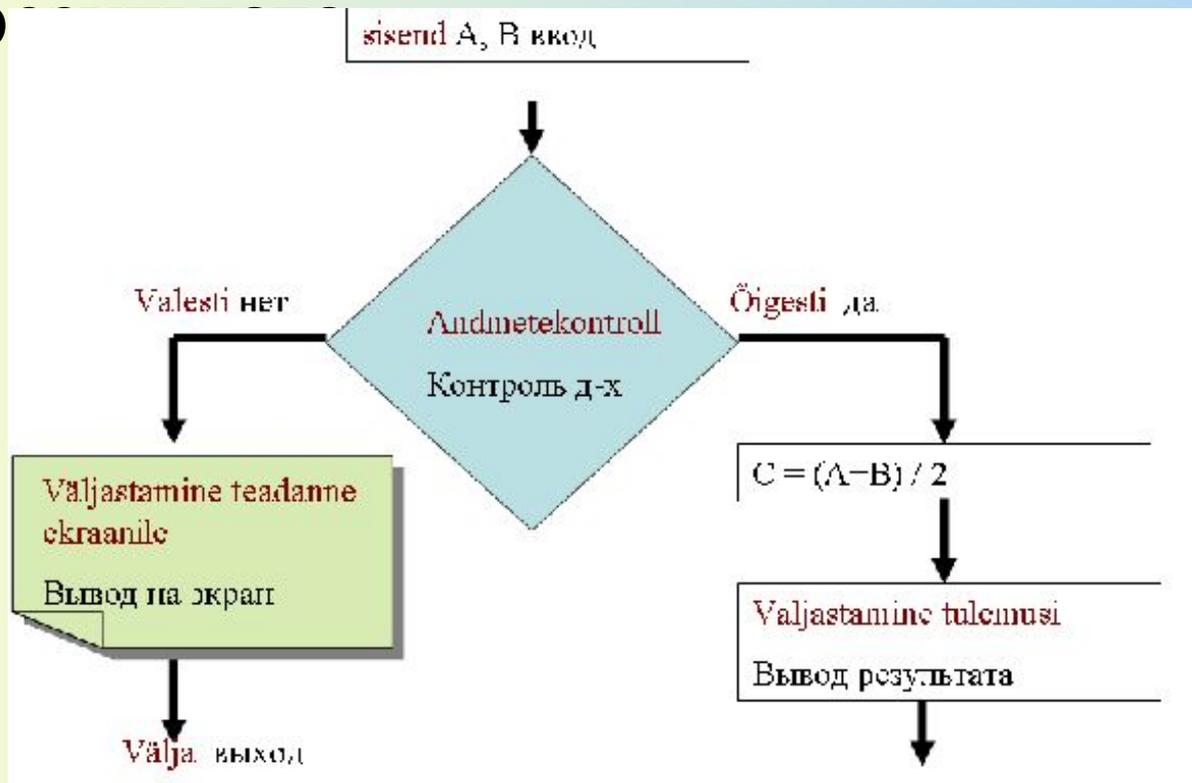
**Программирование** — это процесс создания программы, который может быть представлен с помощью последовательности шагов:

1. **Спецификация** – формулирование требований программе.

## 2. Разработка алгоритма -

определение последовательности действий для получения правильного

р



### 3. Кодирование - запись алгоритма на языке программирования.

```
#include "stdafx.h"
#include <conio.h>
#include <iostream>

using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    cout<<"Hallo!!!"<<endl;
    getch();
    return 0;
}
```

4. Отладка - исправление ошибок.

5. Тестирование - проверка работы программы на различных входных данных.



*Спецификация* - определение требований к программе.

- Описывается исходная информация.
- Формулируются требования к результату.
- Описывается поведение программы в особых случаях, например, при вводе неверных данных.

# Пример спецификации.

**Задание** : составить программу, рассчитывающую среднее значение двух целых чисел.

**Исходная информация** : два целых числа.

**Требования к результату** : целое число, показанное на экране.

**При вводе неверных данных** на экран выводить сообщение.



На этапе *разработки алгоритма* необходимо определить последовательность действий, которые надо выполнить для получения результата.

Результатом этапа разработки алгоритма является его подробное словесное описание или блок - схема. **блок - схема.**



## Пример алгоритма.

1. Ввести с клавиатуры 2 целых числа

**ЕСЛИ ДАННЫЕ ВВЕДЕНЫ ВЕРНО, ТО**

а) Вычислить среднее значение по формуле:  
сложить два целых числа и разделить на

два.

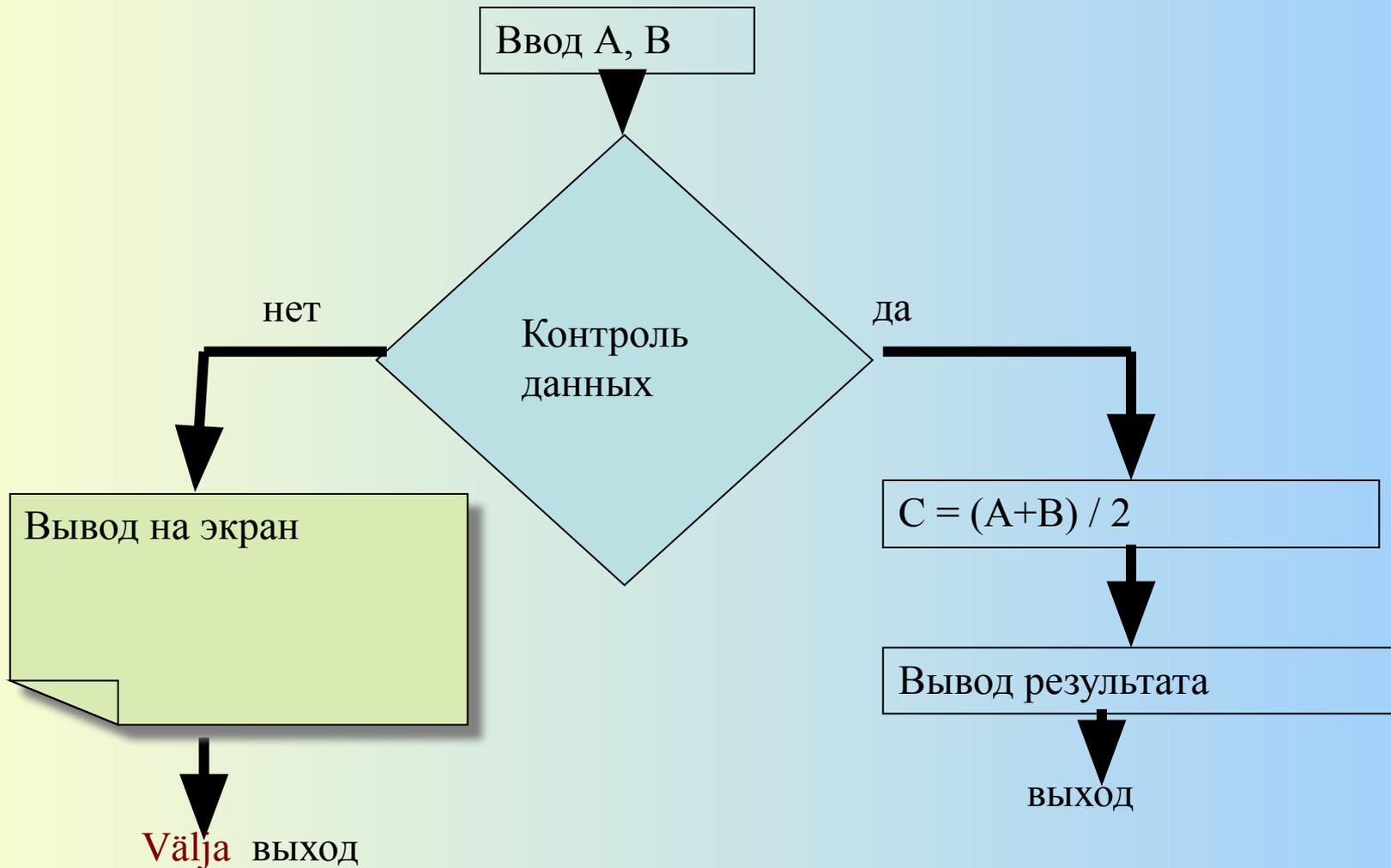
б) Вывести результат на экран.

**ЕСЛИ ДАННЫЕ ВВЕДЕНЫ НЕВЕРНО, ТО**

вывести на экран сообщение



# Блок – схема.



## **Кодирование.**

После того как определены требования к программе и составлен алгоритм решения, он записывается на выбранном языке программирования.

В результате получается **исходная программа.**



## Программа.

```
#include <iostream> //подключение заголовочного файла
```

```
.....
```

```
int main() //главная функция
```

```
{ int A,B, C=0; //объявление 3-х целых чисел со знаком
```

```
cout<<«Введите 2 целых числа: »; //текст на экране
```

```
cin>>A>>B; //ввод с клавиатуры 2-х целых чисел
```

```
//контроль введённых данных
```

```
if ((A>=0 && A<=999) && (B>=0 && B<=999))
```

```
{ // если данные введены правильно
```

```
C=(A+B)/2; //вычисление среднего значения 2-х чисел
```

```
cout<<"C= "<<C<<endl; // вывод результата на экран
```

```
}
```

```
//если данные введены не правильные
```

```
else cout<<«в данных ошибки !!! »<<endl;
```

```
return 0; //конец программы
```

```
}
```



**Отладка** — это процесс поиска и устранения ошибок.

Ошибки в программе разделяют на 2 группы:

синтаксические (ошибки в тексте)

алгоритмические алгоритмические (ошибки,

связанные с неверно разработанным алгоритмом, например, из-за нарушения последовательности выполнения команд)

**Этап отладки можно считать законченным, если программа правильно работает при любом правильном наборе входных данных.**



Сделаем **синтаксические ошибки** в тексте программы

```
#include <iostream.h> //подключение заголовочного файла
using namespace std;
int main() //главная функция peafunksioon
{
    → in A,B, C=0; //в слове int отсутствует буква t
    cout<<"Введите 2 целых числа: ""; //текст на экране
    cin>>A>>B; //ввод с клавиатуры 2-х целых чисел
    //контроль ввода данных
    → if ((A>=0 && A<=999) (B>=0 && B<=999)) //отсутствуют &&
    { //если данные введены правильно, то вычислить «С»
        C=(A+B)/2;
        cout<<"C= "<<C<<endl; //вывести результат на экран
    }
    //если введены неверные данные вывести сообщение
    else
        cout<<"в данных ошибки!!! "<<endl;
    return 0; //программа возвращает нуль
}
```

Компилятор отреагирует на эти ошибки следующим образом.

Compiling...

1programma.cpp

.....error C2065: 'in' : undeclared identifier

.....error C2146: syntax error : missing ';' before identifier 'A'

.....error C2065: 'A' : undeclared identifier

.....error C2065: 'B' : undeclared identifier

.....error C2065: 'C' : undeclared identifier

Error executing cl.exe.

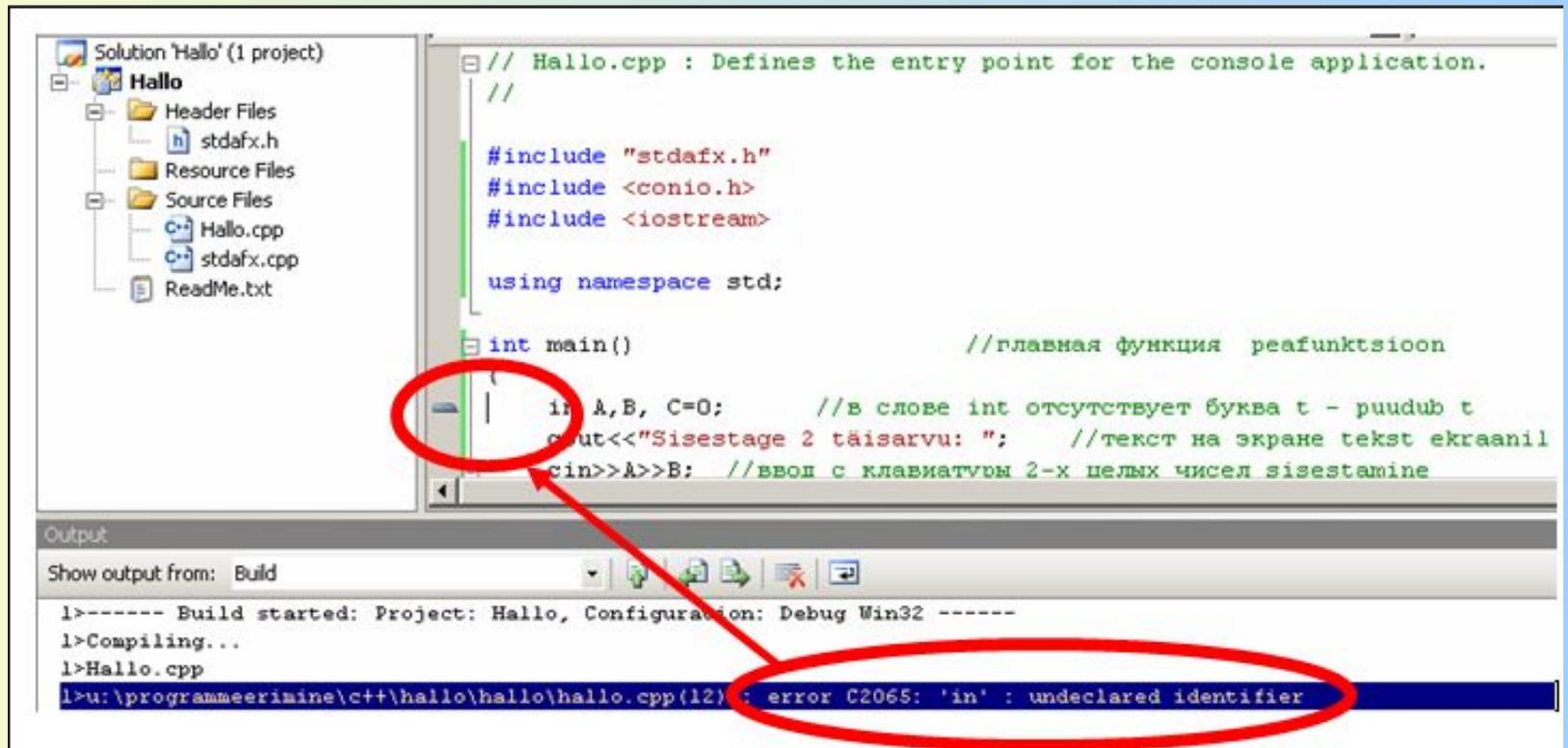
1programma.exe - 6 error(s), 0 warning(s)

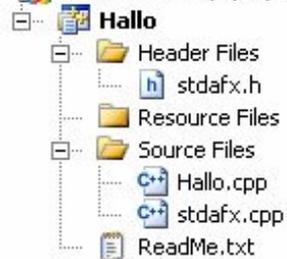
Исправим **in** на **int** и снова компилируем

Компилятор выдаст последующие сообщения об ошибках.

Двойным щелчком мышью по сообщению определяем возможное место ошибки.

Компилятор указывает синей стрелкой, где нужно искать неправильное написание команды.





```
// Hallo.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include <conio.h>
#include <iostream>

using namespace std;

int main() //главная функция peafunktsioon
{
    in A,B, C=0; //в слове int отсутствует буква t - puudub t
    cout<<"Sisestage 2 täisarvu: "; //текст на экране tekst ekraanil
    cin>>A>>B; //ввод с клавиатуры 2-х целых чисел sisestamine
    //контроль ввода данных andmete kontroll
    if ((A>=0 && A<=999) (B>=0 && B<=999)) //puuduvad &&
    { //если данные введены правильно, то вычислить «C»
        //kui andmed on sisestatud õigesti, siis arvuta C
        C=(A+B)/2;
        cout<<"C= "<<C<<endl; //вывести результат на экран Väljasta ekraanile
    }
    //если введены неверные данные вывести сообщение
    //kui andmed on sisestatud valesti, väljasta teade
    else
        cout<<"andmetel on vead!!! "<<endl;
    return 0; //программа возвращает нуль programm tagastab nulli
}
```

## Output

Show output from: Build

```
1>----- Build started: Project: Hallo, Configuration: Debug Win32 -----
1>Compiling...
1>Hallo.cpp
1>u:\programmeerimine\c++\hallo\hallo\hallo.cpp(12) : error C2065: 'in' : undeclared identifier
1>u:\programmeerimine\c++\hallo\hallo\hallo.cpp(12) : error C2146: syntax error : missing ';' before identifier 'A'
1>u:\programmeerimine\c++\hallo\hallo\hallo.cpp(12) : error C2065: 'A' : undeclared identifier
1>u:\programmeerimine\c++\hallo\hallo\hallo.cpp(12) : error C2065: 'B' : undeclared identifier
1>u:\programmeerimine\c++\hallo\hallo\hallo.cpp(12) : error C2065: 'C' : undeclared identifier
1>Build log was saved at "file:///u:\PROGRAMMEERIMINE\C++\Hallo\Hallo\Debug\BuildLog.htm"
1>Hallo - 5 error(s), 0 warning(s)
===== Build: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped =====
```

*Тестирование* особенно важно, если предполагается, что программой будут пользоваться другие.

*Тестирование* позволяет проверить, как ведет себя программа при как можно большем количестве входных наборов данных, в том числе и заведомо неверных.

```
c:\ "U:\KEEL_C_JA_OBJEKTORIENTEERI... _ _ X
Sisestage 2 taisarvu: 2 3
C= 2
Press any key to continue_
```

```
c:\ "U:\KEEL_C_JA_OBJEKTORIENT... _ _ X
Sisestage 2 taisarvu: 6 r
C= 3
Press any key to continue_
```

```
c:\ "U:\KEEL_C_JA_OBJEKTORIENTEE... _ _ X
Sisestage 2 taisarvu: A 9
andmed ei ole oige!!!
Press any key to continue_
```

Если ввести вещественные значения для A и B, то результат останется нулевым.

```
int A,B, C=0;
```

```
c:\ "U:\KEEL_C_JA_OBJEKTORIENTEERITUD_P... _ _ X
Sisestage 2 taisarvu: 1.1 3.3
C= 0
Press any key to continue
```

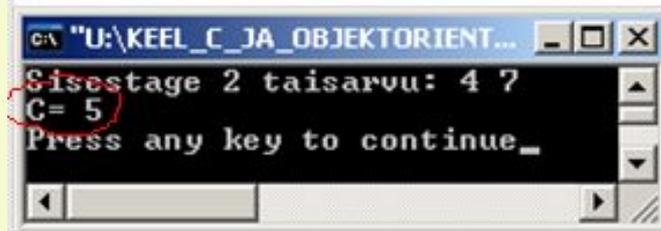


Алгоритмические ошибки связаны, в первую очередь, с неверно разработанным алгоритмом задачи, например, изменением порядка расчёта.

```
#include <iostream.h>

int main()
{
    int A,B, C;
    cout<<"Sisestage 2 taisarvu: ";
    cin>>A>>B;
    C=0;
    C=(A+B)/2;
    cout<<"C= " <<C<<endl;

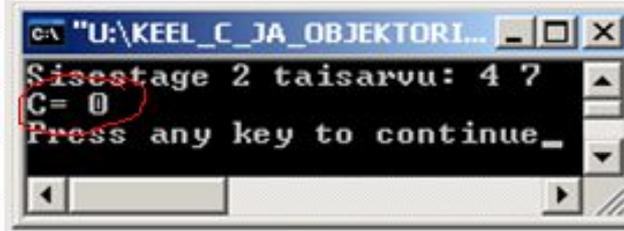
return 0;
}
```



```
#include <iostream.h>

int main()
{
    int A,B, C;
    cout<<"Sisestage 2 taisarvu: ";
    cin>>A>>B;
    C=(A+B)/2;
    C=0;
    cout<<"C= " <<C<<endl;

return 0;
}
```



Спасибо за внимание!

