

Реализация алгоритмов в компьютере. Язык C++

Программа и программирование

Программа – это алгоритм, записанный на языке программирования и предназначенный для исполнения компьютером. В программе действия должны быть описаны до мельчайших подробностей в точном соответствии с синтаксисом языка.

Задачу преобразования исходной программы в машинный код выполняет специальная программа – **компилятор**.

Транслятор (компилятор и интерпретатор) – «программирующая» программа, предназначенная для перевода предписаний исходного кода в операции непосредственно выполняемые компьютером.

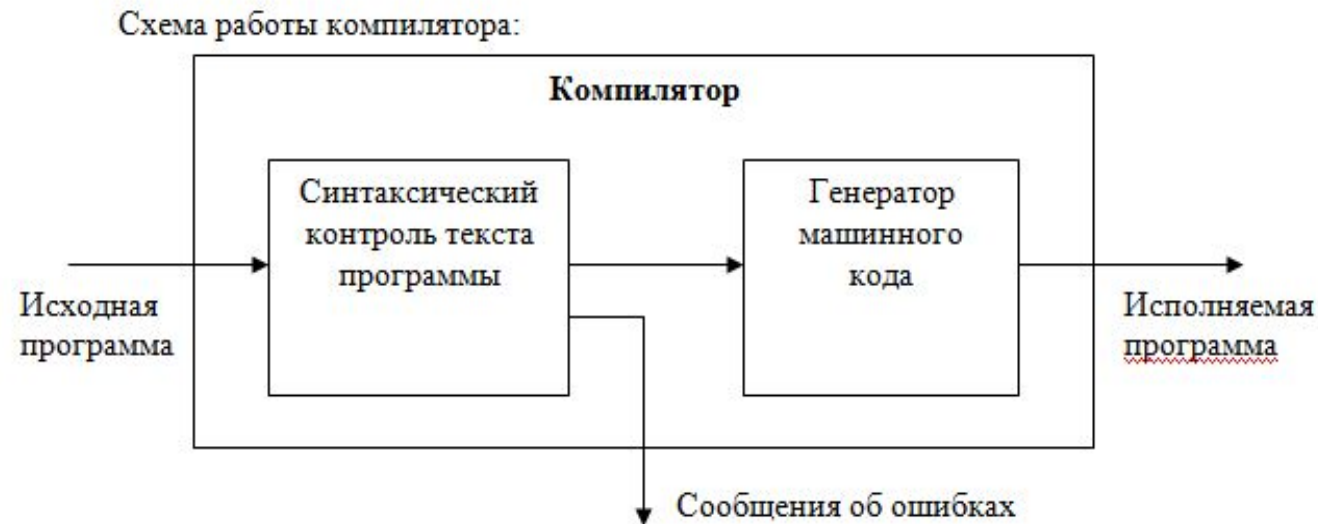
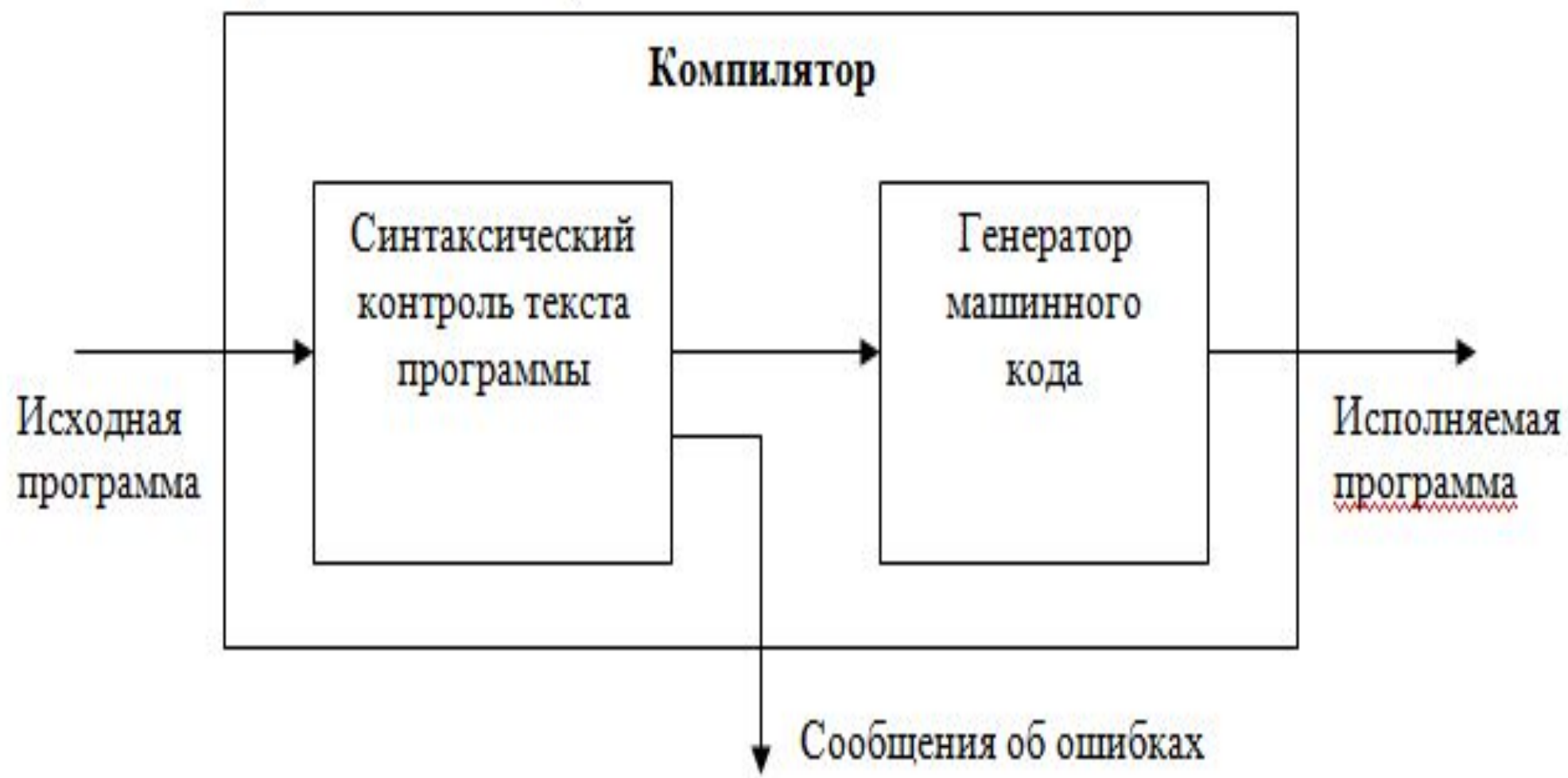


Схема работы компилятора:



Задачи компилятора

1. Проверяет текст исходной программы на отсутствие синтаксических ошибок.
2. Создает (генерирует) исполняемую программу – машинный код.

Генерация машинного кода компилятором свидетельствует только об отсутствии в тексте программы синтаксических ошибок.

Стиль программирования

- использование комментариев;
- использование несущих смысловую нагрузку имен переменных, процедур и функций;
- использование отступов;
- использование пустых строк.

Этапы решения задач на компьютере

- Постановка задачи (выявление объектов)
- Построение математической модели (с помощью математических средств описываются характеристики объекта)
- Разработка алгоритма (поиск метода решения сформулированной математической задачи)
- Выполнение программы (ввод, загрузка программы в память компьютера, перевода, трансляции отладки и выполнения)

Объекты задач и программ

Любая задача связана с реальными **объектами** из своей предметной области. При анализе и решении задачи реальные объекты подменяются **объектами задачи**.

Виды объектов:

1. Константа
2. Переменная
3. Массив
4. Выражения и операторы
5. Комментарии
6. Идентификаторы

Технология программирования

- Постановка задачи
- Проектирование программы
- Написание и отладка программы

Модель предметной области

Модель для задачи нахождения суммы и разности двух целых чисел

$$S=a+b$$

$$R=b-a$$

ER-Модель предметной области абонемент библиотеки



Задача 2. Наборы пирожных

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 0.4 секунд |
| Ограничение по памяти: | 512 мегабайт |

На складе кондитерской фабрики хранятся пирожные двух видов — круассаны и эклеры. Круассанов A штук, а эклеров — B штук. Есть неограниченный запас подарочных коробок, в каждую коробку можно положить только три пирожных. При этом требуется, чтобы в коробке были пирожные обоих видов, то есть в одну коробку можно положить два круассана и один эклер или один круассан и два эклера.

Определите, можно ли упаковать все имеющиеся пирожные в коробки и выведите подходящий способ размещения пирожных по коробкам.

Формат входных данных

Программа получает на вход два целых числа A и B , записанных в отдельных строках. $1 \leq A \leq 10^9$, $1 \leq B \leq 10^9$.

Формат выходных данных

Если можно разложить все пирожные по коробкам в соответствии с условием задачи, программа должна вывести два целых числа. Первое число равно количеству коробок, в которых лежит два круассана и один эклер. Второе число равно количеству коробок, в которых лежит один круассан и два эклера.

Если разложить все пирожные по коробкам нужным способом нельзя, программа должна вывести одно число -1 .

Система оценки

Решение, правильно работающее только для случаев, когда числа A и B не превосходят 100, будет оцениваться в 60 баллов.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 4 5 | 1 2 |
| 5 3 | -1 |

Замечание

В первом примере нужно взять одну коробку с двумя круассанами и одним эклером и две коробки с одним круассаном и двумя эклерами. Всего получится 4 круассана и 5 эклеров.

Пример реализации простейшей программы на компьютере

Условие: Составьте программу реализации арифметических операций «+» и «-» с двумя натуральными числами.

Определение проблемной области: используемая область знаний – арифметика; натуральные числа – это числа целые, положительные и не 0; осуществить операцию «+» или сложить два числа или найти сумму двух чисел – это увеличение значения первого аргумента на значение второго аргумента; осуществить операцию «-» или вычесть из одного числа другое или найти разность двух чисел – это уменьшить значение первого аргумента на значение второго аргумента.

Дано: два натуральных числа (целые, $\langle \rangle 0?; >0$) **Требуется:** найти сумму, как первое число прибавить второе число

Дано: два натуральных числа (целые, $\langle \rangle 0?; >0$), второе число $>$ первого числа. **Требуется:** найти разность, из значение второго числа уменьшить на значение первого

Макет исходных данных: программа получает на вход два натуральных числа a и b , записанных в отдельных строках, $1 < A < 1000000$, $1 < B < 1000000$, $B > A$

Макет результатов работы: программы выводит на экран на разных строках значение суммы S , значение разности R , в программе предусмотрена возможность ввода ненатурального числа.

Объекты: A, B, S, R – целые числа.

Дано: **A, B** Найти:

S, R

$$**S = A + B**$$

$$**R = B - A**$$

1. «Это программа вычисления суммы и разности двух натуральных чисел»
2. «Введите первое натуральное число (число > 0, число отличное от 0)», A
3. «Введите второе натуральное число (число > 0, число отличное от 0, число большее первого натурального числа)», B
4. $S = A + B$
5. $R = B - A$
6. «Результатом арифметической операции сложения является сумма, она равна», S
7. «Результатом арифметической операции вычитания является разность, она равна», R

Программа на языке C/C++ состоит из:

- Директив препроцессора
- Объявлений
- Описаний функций (главной и пользовательских функций, вызываемых в главной)

Функция в C++

```
1  тип_результата имя_функции (параметры)
2
3  {
4  описание данных;
5  оператор1;
6  оператор2;
7  оператор3;
8  ...
9  операторN;
10 }
```

Func.cpp hosted with ❤ by GitHub

```
1 директивы препроцессора //1
2
3 описание глобальных переменных //2
4
5 тип_результата main (параметры) //3
6
7 {
8 тело главной функции
9 }
10
11 тип_результата F1 (параметры 1) {
12 тело функции F1;
13 }
14
15 тип_результата F2 (параметры 2) {
16 тело функции F2;
17 }
18
19 тип_результата FN (параметры N) {
20 тело функции FN;
21 }
```


Директива #include

```
1 #include <имя_включаемого_файла>  
2 #include "имя_включаемого_файла"
```

Include.cpp hosted with ❤ by GitHub

```
1 #include <stdio.h> /*функции стандартного ввода-вывода*/  
2 #include <conio.h> /*функции консольного ввода-вывода*/  
3 #include <math.h> /*стандартные математические функции*/
```

Include.cpp hosted with ❤ by GitHub

ФУНКЦИЯ main

```
1 int main (int argc, char *argv[])  
2 {  
3  
4 }
```

Пример программы и функции main

```
1 #include <iostream>           // подключаем заголовочный файл iostream
2
3 int main()                     // определяем функцию main
4 {                               // начало функции
5     std::cout << "Hello World!"; // выводим строку на консоль
6     return 0;                  // выходим из функции
7 }                               // конец функции
```

Процесс создания программы

