Технологии разработки алгоритмов решения инженерных задач

лекция №1

Преподаватель: Дреев Александр Николаевич

1. Введение в предмет
   1.1. определение алгоритма
   1.2. свойства алгоритма
   1.3. Условия для возможности выполнения алгоритма
   1.4. Способы задания алгоритма. Блок-схема, функциональная схема, схема потоков информации.
   1.5. Требования к оформлению работ

Введение в предмет

определение алгоритма

алгоритм - конечное заранее определена последовательность действий, выполнение которых приводит к розвьязанння группы задач.

Действия должны быть понятными исполнителю и исполнитель должен быть способен их выполнить. Интеллект имеет способность создавать новые алгоритмы.

Введение в предмет

свойства алгоритма

конечности - для выполнения действий нужна конечная количество времени.

1. Подпрыгни.
2. Перейди п.1

Это не алгоритму.

правильность - алгоритм дает правильный ответ или определяет ее отсутствие.

сложность - указывает на количество операций необходимых для получения результата и зависимость их количества от количества входных данных.

Введение в предмет

свойства алгоритма

сложность задают O () символикой.

1. *i* = 0. *S* = 0.
2. Число с номером *и* добавляем к *S*
3. *и* увеличиваем на 1
4. Если есть еще числа, тогда п.2

Сложность O (n)

1. Рассадить n человек в n стульев так, как они еще не садились.
2. Если есть еще варианты, то п.1

Сложность O (n!)

Введение в предмет

свойства алгоритма

понятность - все команды должен знать исполнитель

однозначность - все команды трактуются единственным образом

Введение в предмет

Условия возможности выполнения

- Исполнитель понимает язык
- Алгоритм обеспечивает конечности количества шагов (отсутствуют зацикливания, это следствие правильности алгоритма)
- Исполнитель имеет доступ к входным данным
- Исполнителю достаточно памяти и времени
- Алгоритм использует другие уже известные алгоритмы

- Алгоритм можно выполнить, даже если он дает неправильные результаты

Введение в предмет

Способы записи алгоритма

Описательный, рецепт - алогоритм для человека, команды не однозначны и их толкование зависит от контекста.

текстовый - алгоритм задается последовательным текстом, возможно по шагам, как на предыдущих слайдах.

Блок-схеме - каждое действие отделена геометрической фигурой, о содержании которой договорено заранее; порядок перехода между действиями задано линиями и стрелками.

Введение в предмет

Способы записи алгоритма

алгоритмических языке - алогоритм текстовый, но использована искусственная речь, например для облегчения взаимопонимания с автоматом- исполнителем.

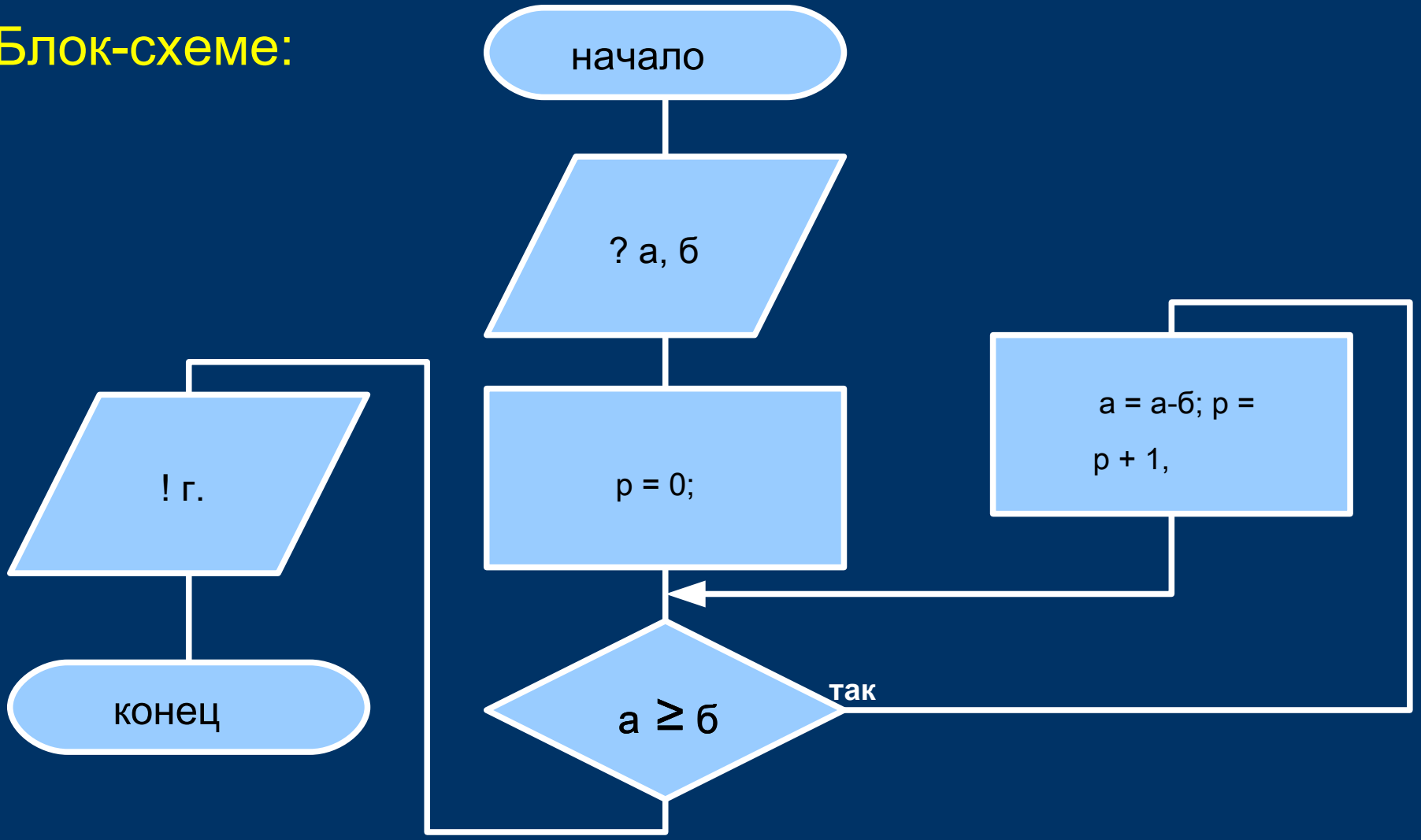
функциональной схеме - алгоритм задается связями между функциональными блоками; хорошо объясняет механизм обработки информации.

Схема потоков информации - указывает на источники и приемники информации, отражает параллельные и независимые потоки обработки информации; акцент на последовательности обработки.

Введение в предмет

Способы записи алгоритма

Блок-схеме:



Введение в предмет

Способы записи алгоритма

Алгоритмических языке:

Вход: а - деленное; Вход:

б - Делитель;

Выход: р - результат деления; р = 0;

Пока а≥б делай (а =

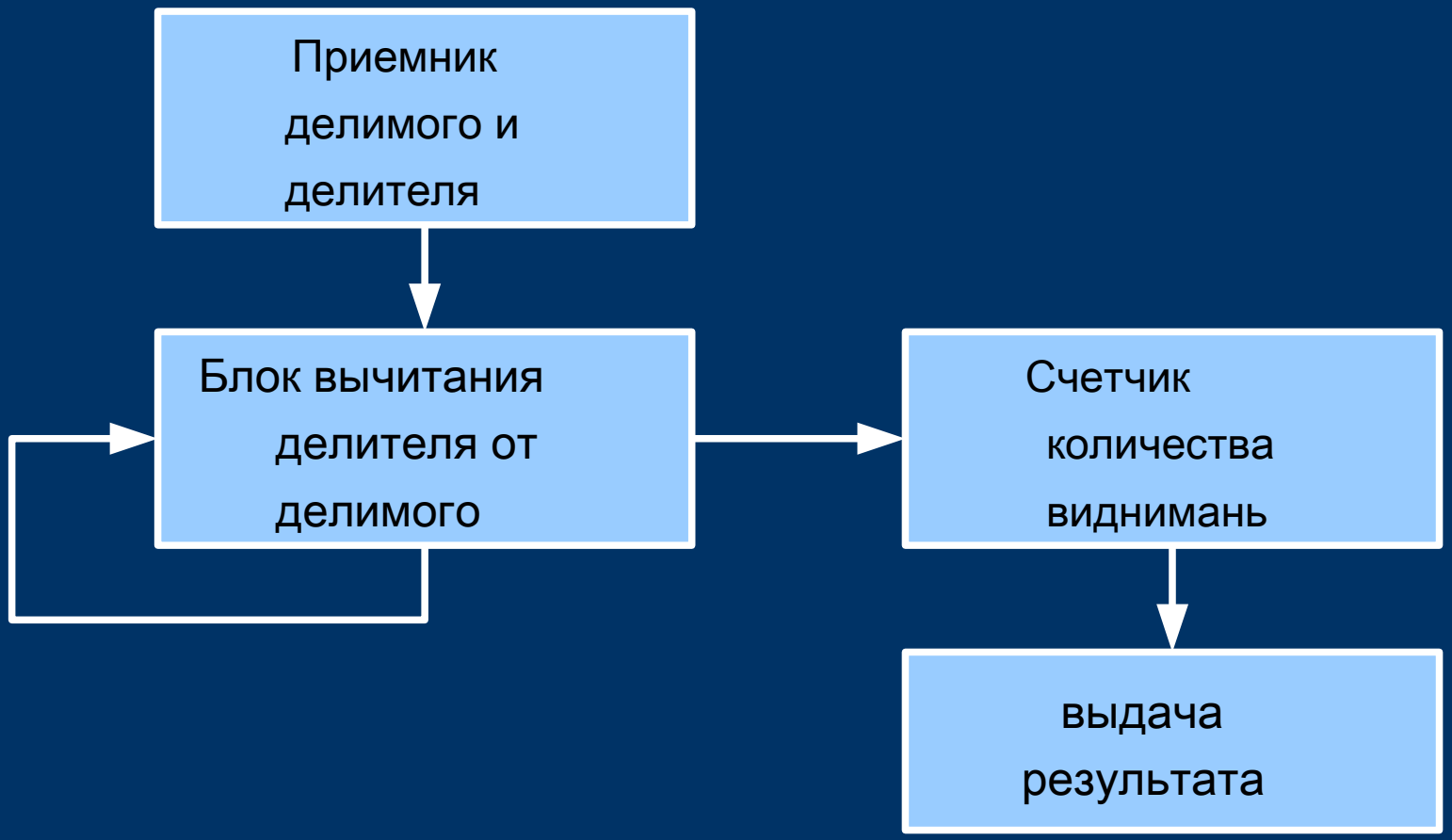
а-б, р = р + 1,

)

Введение в предмет

Способы записи алгоритма

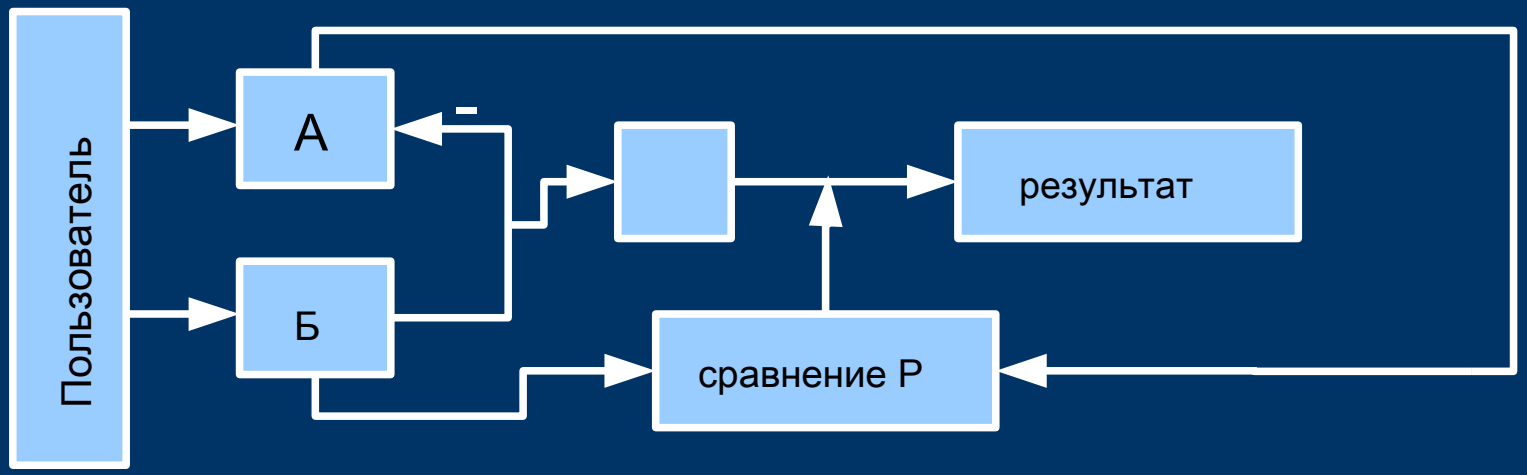
Функциональная схема:



Введение в предмет

Способы записи алгоритма

Схема потоков информации:



Введение в предмет

качество алгоритма

Качество алгоритма:

1) Минимальная сложность O ()
2) Универсальность, у доспустимих задач
3) Защита от неправильного использования
4) Максимальное быстродействие
5) Минимальность количества команд
6) Минимальность потоков информации
7) полиморфность, допустимость изменения структуры
8) Простота

Введение в предмет

Требования к оформлению работ

Работы оформляются в отчетах по стандартным титульном листе, либо в отдельном подписанном тетради.

Работа содержит

1) Текст задания
2) Текстом: основная идея ров'язання
3) Структура алгоритма
4) Потоки информации в алгоритме
5) Блок-схема алгоритма, общая

Введение в предмет

Требования к оформлению работ

6) Блок-схемы вспомогательных алгоритмов (при необходимости или по требованию преподавателя)

7) Запись на языке программирования

8) Оценка сложности алгоритма теоретически и экспериментально

9) Выводы (или выполнено задание, как иначе можно было ровьязаты задачу, является алгоритм качественным, в каких пределах входных данных можно использовать алгоритм)

Введение в предмет

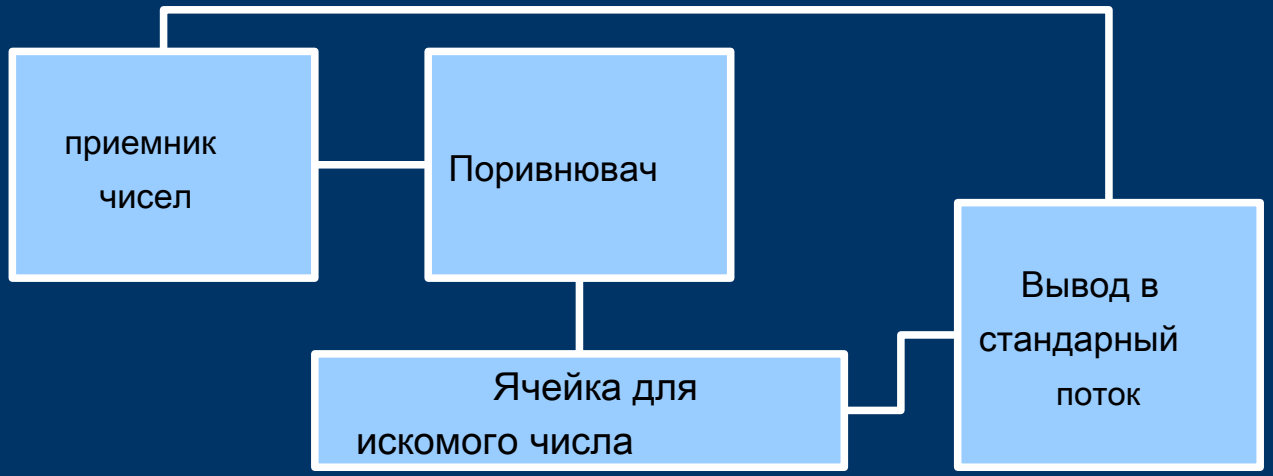
Требования к оформлению работ

Пример.

Задание: В последовательности целых чисел, из стандартного потока данных, разделенных знаком перевода сроки найти наибольшее число и вывести его в стандартный поток данных.

Идея решения: Первое число считаем самым. Пока в потоке ввода данных является числа, читаем одно, и если оно больше уже найдено, считаем его новым наибольшим числом.

Структурная схема:

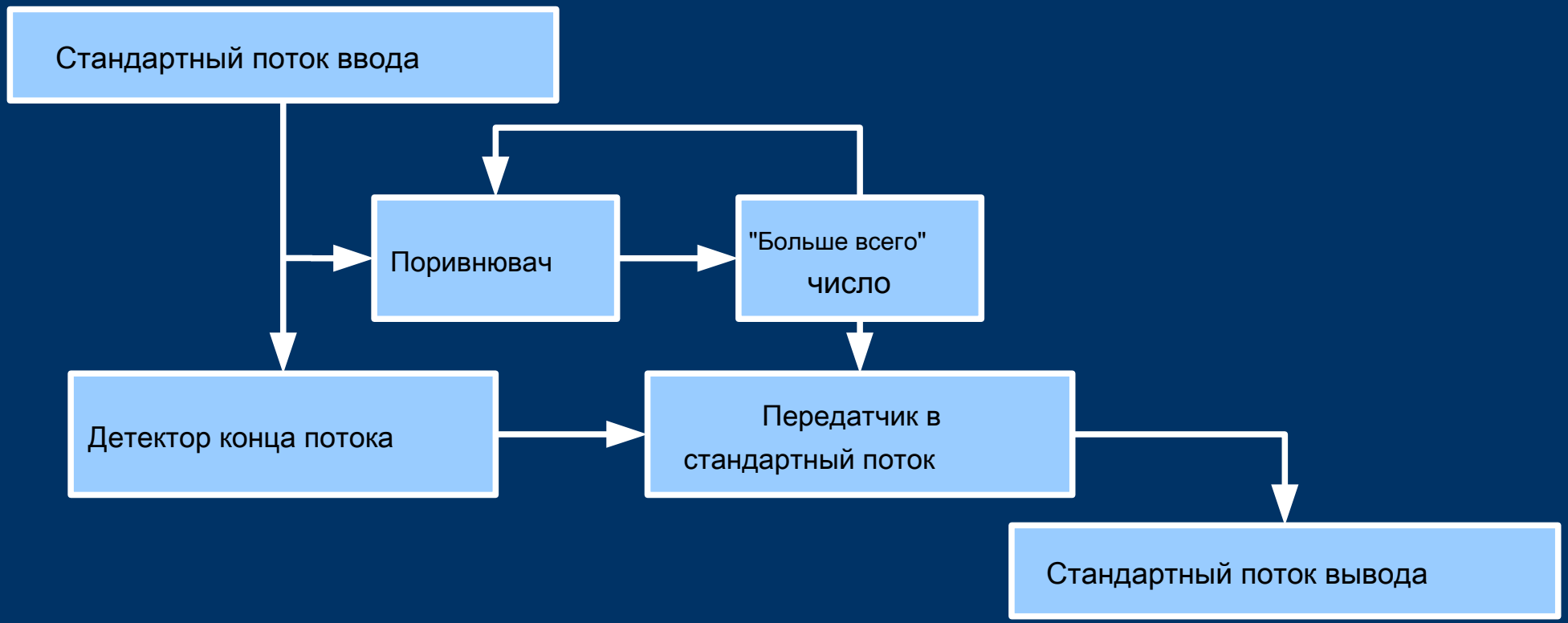


Введение в предмет

Требования к оформлению работ

Пример.

Схема потоков информации:

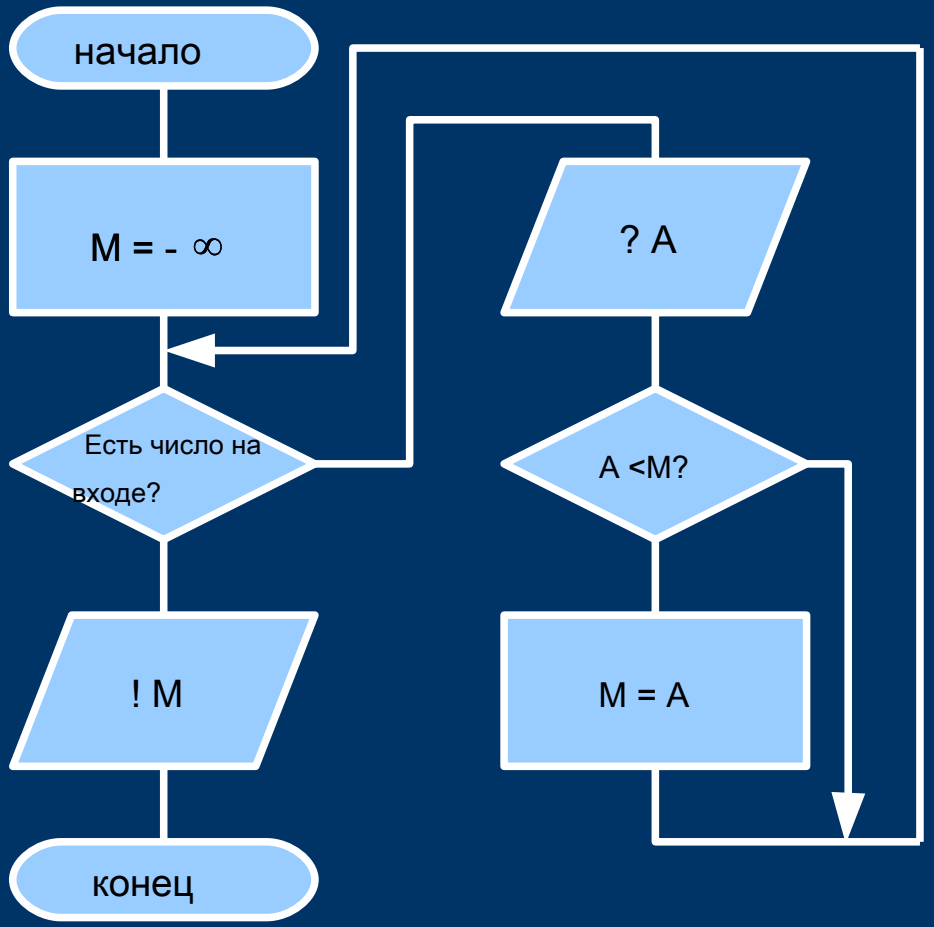


Введение в предмет

Требования к оформлению работ

Пример.

Блок-схема алгоритма:



Введение в предмет

Требования к оформлению работ

```
#include <iostream> int main ( int N, char * Param []) {    int M = 0x80000000 ; //
```

32-х разрядное наименьшее целое

```
int A = 0 ;
while (! Std :: cin.eof ()) // Пока не конец потока ввода, повторить
{    std :: cin >> A; // Читаем число

    if (A> M) M = A; // Если больше, то теперь оно в M
} std :: cout << M << std :: endl; // Вывод крупнейшего

return 0 ; }
```

Введение в предмет

Требования к оформлению работ

Файл data.txt с пробными числами:

8545

56

45

32

Запуск программы на выполнение:

C: \ CPP \ WIN32> program.exe <data.txt> result.txt Содержимое файла result.txt 56

Программа работает верно.

Введение в предмет

Требования к оформлению работ

Оценка сложности:

При выполнении программы, алгоритм сравнивает каждый входной элемент с одним элементом, поэтому время выполнения программы пропорционален количеству элементов. Сложность алгоритма O (n). data.txt - 1341 целых чисел

Запуск: tmeter.exe find_max.exe <data.txt Результат:
84 ms

data.txt - 2682 целых числа
Запуск: tmeter.exe find_max.exe <data.txt Результат:
172 ms

При увеличении входных данных вдвое, время работы увеличилось вдвое.
Теоретически оценка совпадает с практической.

Введение в предмет

Требования к оформлению работ

вывод:

Разработан алгоритм, программу отыскания максимального числа из набора чисел. Программа прошла тестовые задания и работает правильно. Алгоритм имеет линейную сложность. Программа может быть использована как фильтр в формировании командных файлов.

При поиске наибольшего числа нужно проверять все числа, поэтому алгоритм с меньшей сложностью создать нельзя.

Пути совершенствования программы не найдено.

Ответы на контрольные вопросы