

Елена Альфредовна Ерохина

Программирование

Контрольные точки первого модуля

Программирование

- Лекции
- Семинары
- 3 лаб. работы
- 1 контрольная работа

Лабораторные работы по программированию для группы АП-11

Проводятся на каф. ИТАС (ауд. 433, 435).

Начало занятия в 10.30

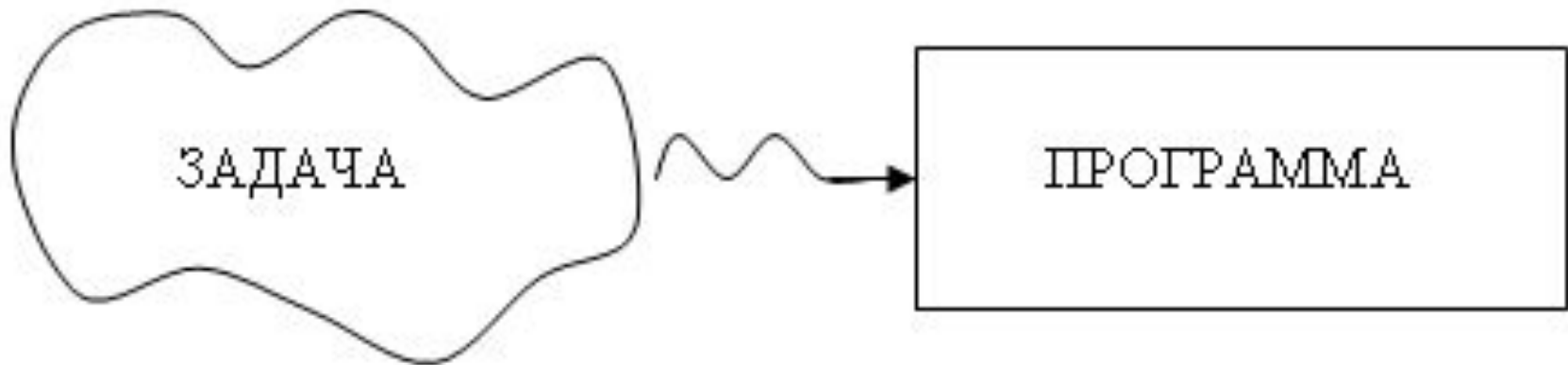
1. 21 сентября
2. 5 октября
3. 19 октября

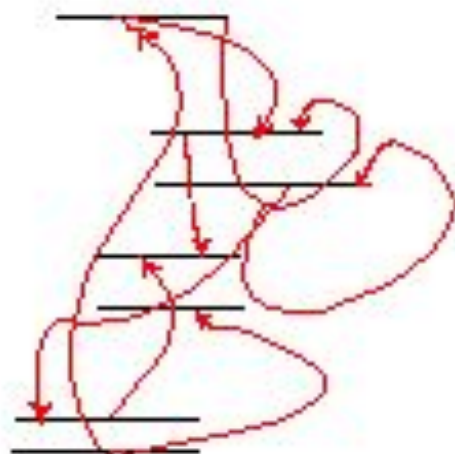
Контрольная работа 19 октября.

Примечание. В случае изменения графика староста известит группу заблаговременно

Три подхода к составлению алгоритмов и программ

1. «Традиционный»





BS - программа

Недостатки метода

- программы трудно читать;
- трудно искать и исправлять ошибки в программах;
- почти невозможно вносить изменения в программу;
- этому «искусству» трудно научиться и невозможно обучить.

2. Структурный



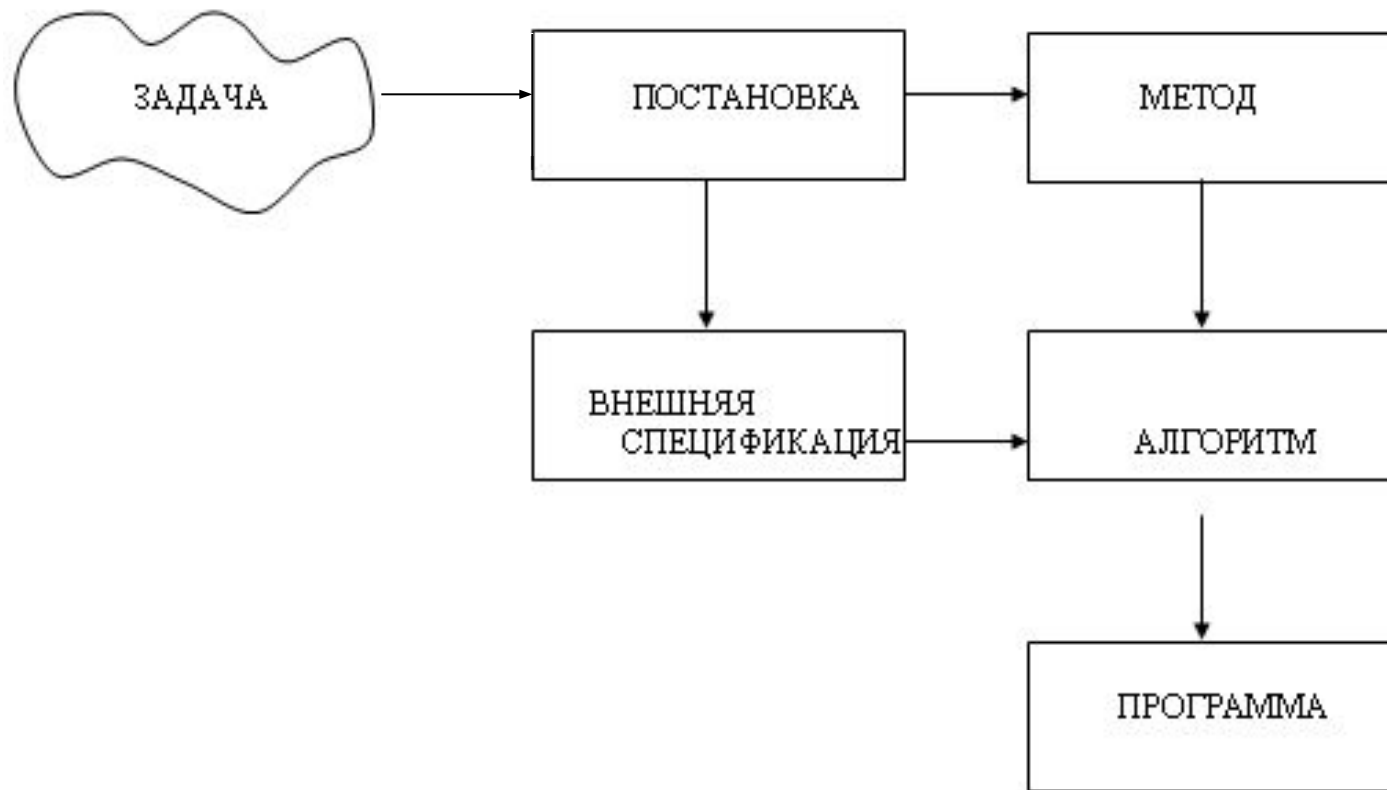
Преимущества

- Есть алгоритм, по которому можно разобраться в работе программы;
- если алгоритм структурный, то и программа структурирована.

Недостатки

- Переход от задачи к алгоритму не формализован, т.е. неясно, как он происходит;
- если алгоритм не структурный, то и структура программы может быть нечеткой.

3. Систематический



Преимущества метода

- Надежность программ;
- эффективность (программы легко оптимизировать);
- экономичность (сокращается время отладки);
- эргономичность (алгоритмы удобны для чтения и понимания);
- модифицируемость (программы легко изменять);
- полная документированность программ;
- этот метод эффективен при обучении программированию.

Определение структурного программирования

Структурное программирование – это программирование с использованием только четырех базовых конструкций (структур).

Основные алгоритмические структуры (конструкции)

- Простая последовательность действий;
- условная конструкция (выбор)
 - if ... then...else
 - case ...of
- ЦИКЛЫ:
 - от ... до (со счетчиком)
 - for ...to / downto...do
 - с предусловием
 - while...do
 - с постусловием
 - repeat...until
- подпрограммы:
 - procedure, function

Основные определения

- **Постановка задачи** – математически точное описание поставленной задачи.
- **Метод решения задачи** – обобщенный способ решения задач данного класса.
- **Внешняя спецификация** – описание входных и выходных данных задачи, а также всех вариантов хода диалога пользователя и программы.
- **Алгоритм** – последовательность команд, которые должен выполнить данный исполнитель для решения данной задачи.
- **Программа** – алгоритм, записанный на заданном языке программирования.

Общий вид постановки задачи

Задача: <условие задачи>

Дано: <исходные данные>

Результат: <результаты решения задачи>

При: <ограничения на допустимость исходных данных>

Связь: < как связаны «дано» и «результат»>

Постановка задачи

Задача: решить линейное уравнение $ax + b = 0$

Вариант 1.

Дано: a, b .

Результат: x .

При: $a \neq 0$.

Связь: $ax + b = 0$.

Вариант 2.

Дано: a, b .

Результат: x ,

или сообщение «нет корней»,

или сообщение «корень –

любое число».

При: -

Связь: $ax + b = 0$.

Метод решения задачи

- При $a \neq 0$ $x = -b / a$.
- При $a=0$ и $b=0$ сообщение «корень – любое число».
- При $a=0$ и $b \neq 0$ сообщение «нет корней».

Внешняя спецификация

Решение линейного уравнения
Введите коэффициенты a , b
<a>

При $a \neq 0$

— Корень уравнения $x = \langle\langle x \rangle\rangle$

При $a = 0$ и $b = 0$

— Корень – любое число

При $a = 0$ и $b \neq 0$

— Нет корней

Обозначения

- $\langle \rangle$ - ввод данных пользователем;
- $\langle \langle \rangle \rangle$ - вывод результата;
- $\{$ - показывает связь условия и соответствующих действий;
- $\{ \}^*$ - повторение действий.

Алгоритм (на псевдокоде)

Алг «решение линейного уравнения»

Нач

вывод(«Решение линейного уравнения »)

вывод(«введите коэффициенты a,b»)

ввод(a,b)

если $a \neq 0$ то

$x := -b/a$

 вывод(«Корень уравнения $x =$ », x)

иначе

если $b = 0$ то

 вывод(«корень – любое число»)

иначе

 вывод(«нет корней»)

все

все

Кон

a

вещ b

вещ x

вещ

Алгоритм (блок-схема)

