Соловьев С.Ю.

soloviev@glossary.ru

Алгоритмы и Алгоритмические языки

www.park.glossary.ru/pascal/

Лекция No. 11

2022

Что дальше?

- Нисходящее программирование
- Отладка и тестирование
- Сложность алгоритмов
- Структуры данных
  - Списки
  - Очереди
  - Стеки
  - Таблицы
    - неупорядоченные
    - упорядоченные
    - хеширование
      - открытое
      - закрытое
  - Деревья поиска
    - полностью сбалансированные
    - Фибоначчи
    - AVL

Нисходящее программирование

– методика составления программ, предполагающая:

-1- выделение из исходной задачи *более простых* подзадач;

-2- оформление (заголовков) процедур и функций, решающих подзадачи;

-3- разработку алгоритма решения исходной задачи с использованием процедур и функций, описанных на этапе -2-; { уже/еще }

-4- применение методики нисходящего программирования к каждой из подзадач.

Нисходящее программирование. Пример

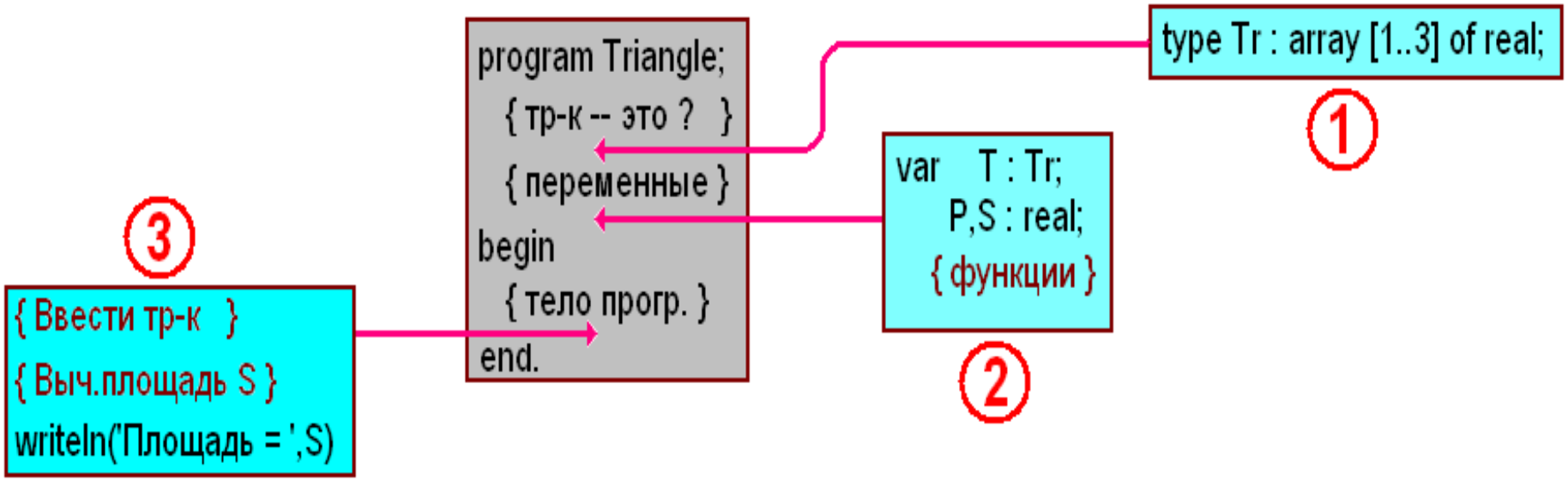
Задача: Вычислить площадь заданного треугольника.

Старт.

```
program Triangle;
  { тр-к -- это ? }
  { переменные }
begin
  { тело прогр. }
end.
```

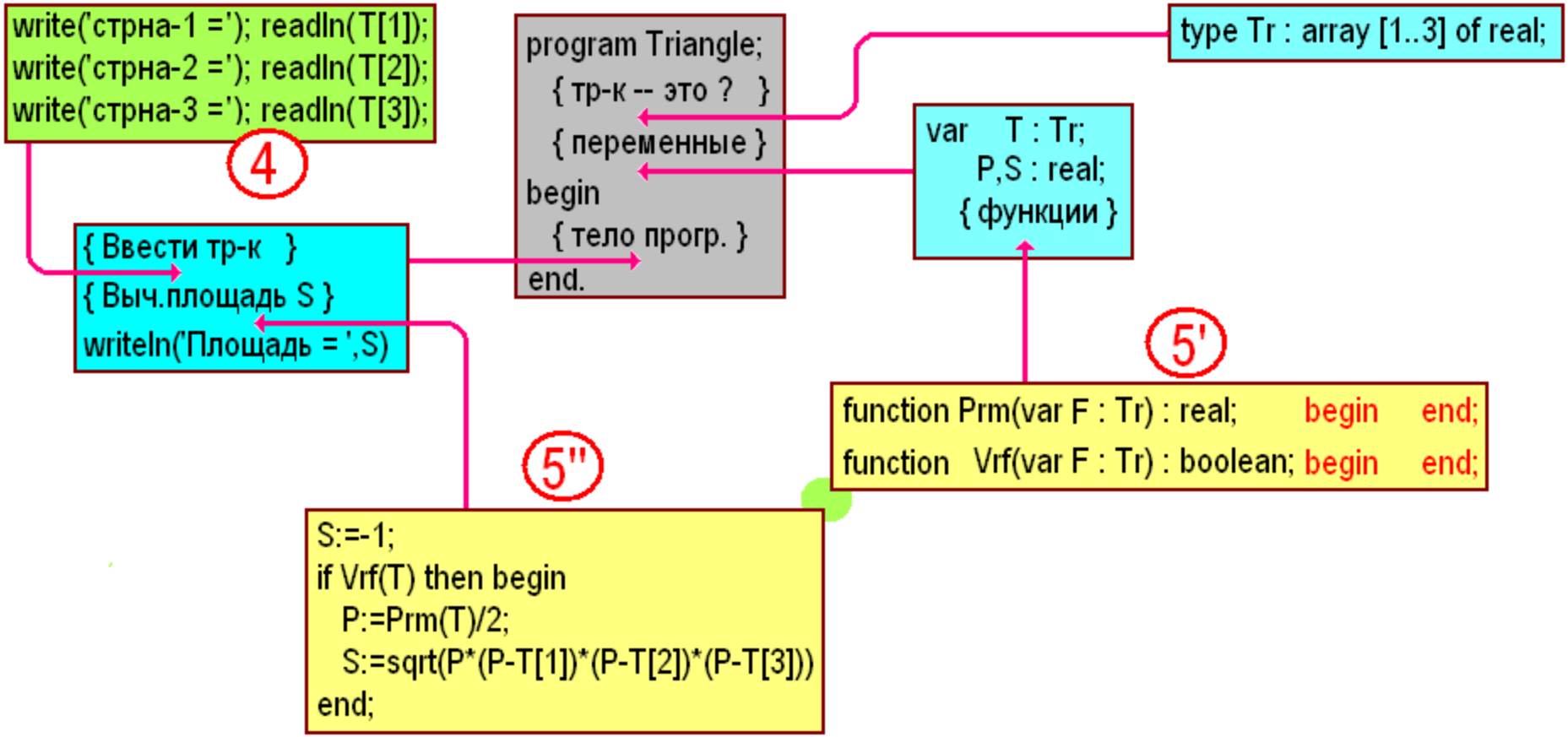
Нисходящее программирование. Пример

Часть 1.



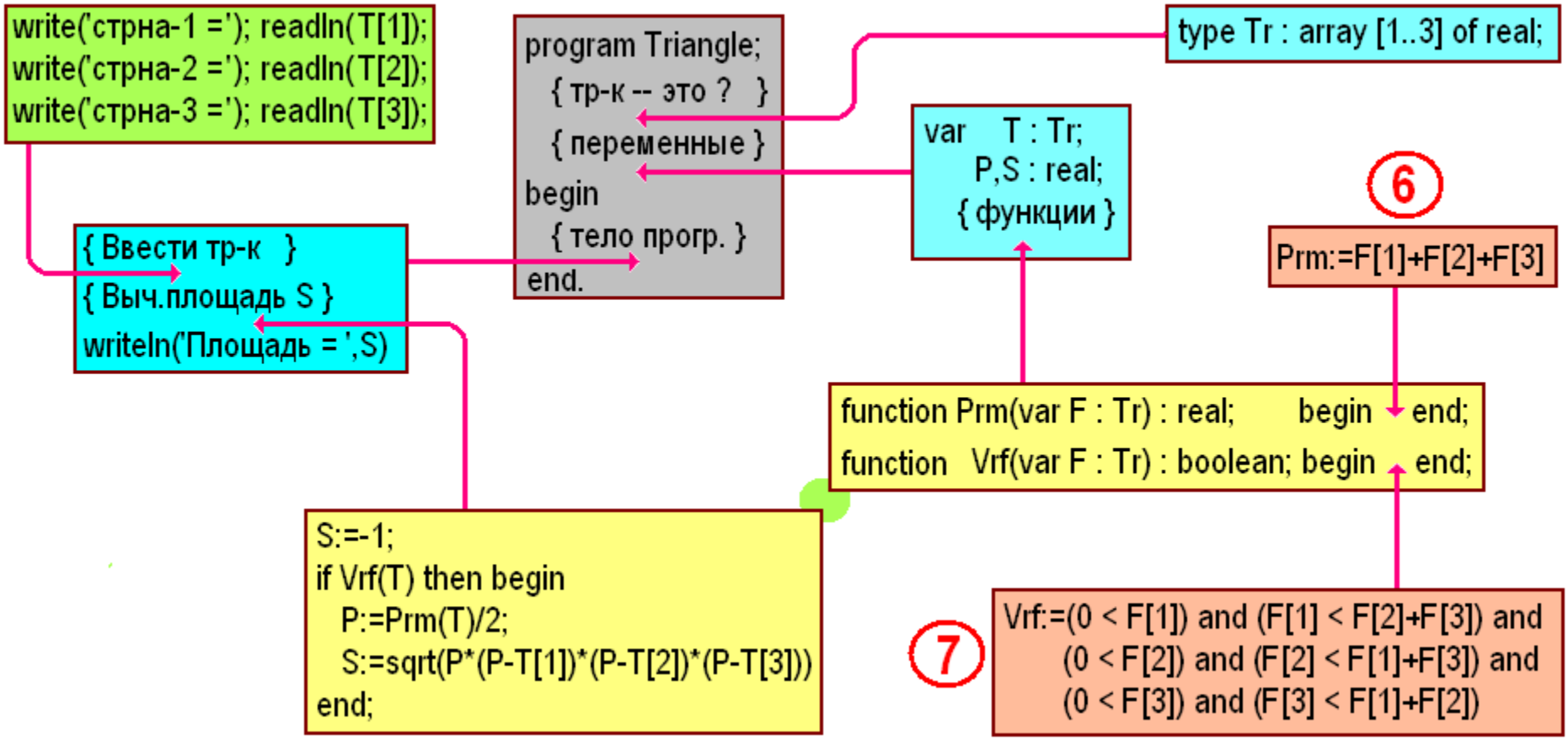
Нисходящее программирование. Пример

Часть 2.



Нисходящее программирование. Пример

Часть 3. Последняя



Нисходящее программирование. Пример

Итоговый текст программы

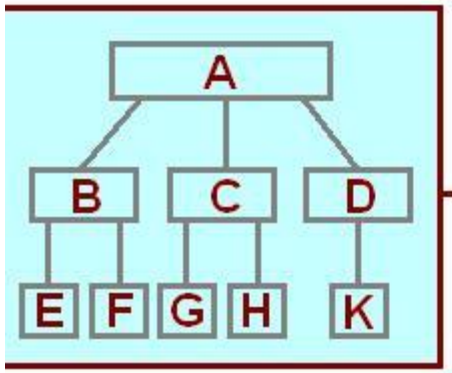
```
program Triangle;
  { тр-к это?  }
type Tr = array [1..3] of real;
  { переменные }
var     T : Tr;
       P,S : real;
  { функции }
  function Prm(var F : Tr) : real;
  begin   Prm:=F[1]+F[2]+F[3]
  end;
  function Vrf(var F : Tr) : boolean;
  begin Vrf:=(0 < F[1]) and (F[1] < F[2]+F[3])  and
                (0 < F[2]) and (F[2] < F[1]+F[3])  and
                (0 < F[3]) and (F[3] < F[1]+F[2])
  end;
```

```
begin
  { тело прогр. }
  { Ввести тр-к   }
    write('сторона-1 ='); readln(T[1]);
    write('сторона-2 ='); readln(T[2]);
    write('сторона-3 ='); readln(T[3]);
  { Выч.площадь S }
    S:=-1;
    if Vrf(T) then begin
      P:=Prm(T)/2;
      S:=sqrt(P*(P-T[1])*(P-T[2])*(P-T[3]))
    end;
    writeln('Площадь = ',S)
end.
```

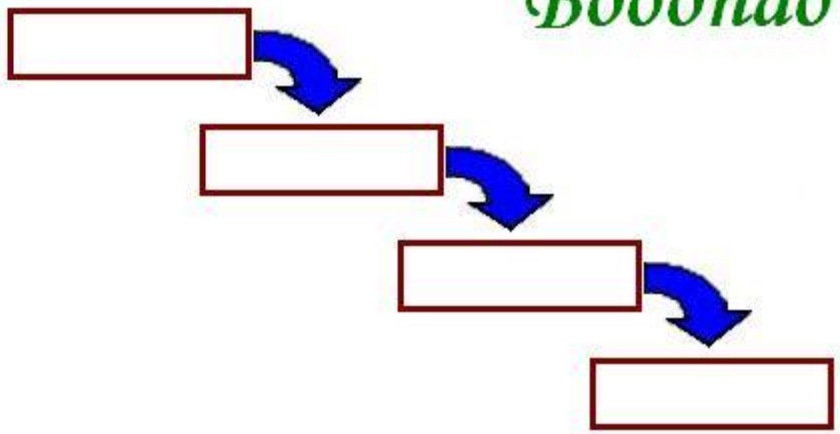
Нисходящее vs. Восходящее
программирование

Нисходящее

1. A *исп.* B', C', D'
2. B *исп.* E', F'
3. E
4. F
5. C *исп.* G', H'
6. G
7. H
8. D *исп.* K'
9. K



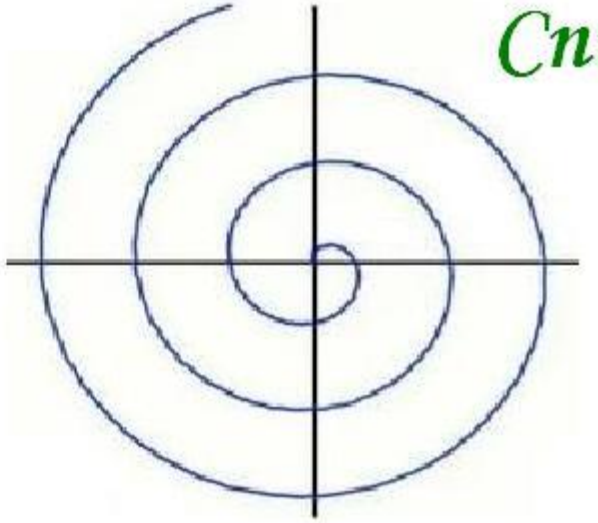
Водопад



Восходящее

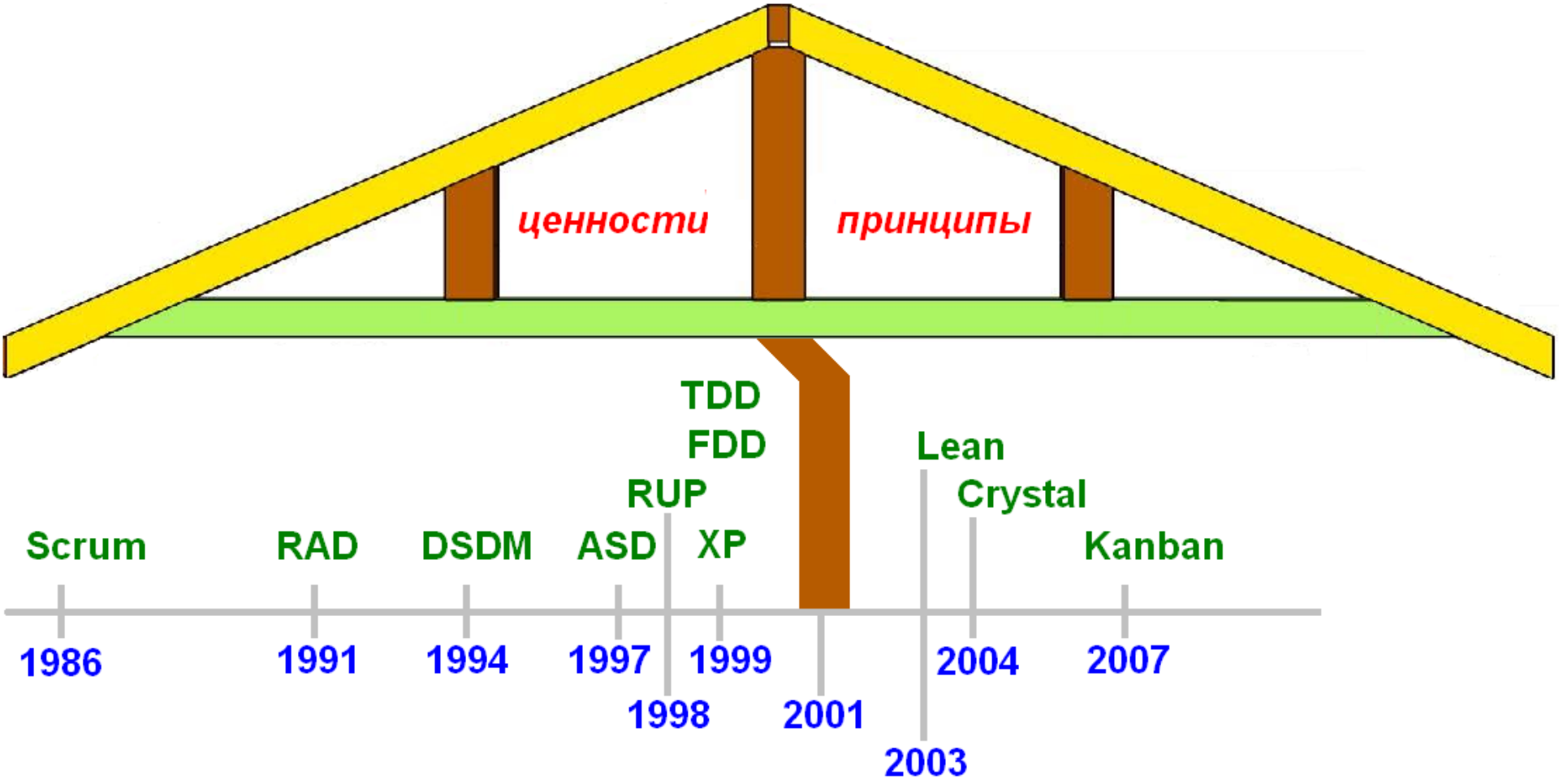
1. E B A
2. F B A
3. G C A
4. H C A
5. K A

Спираль



Agile–семейство методологий

гибкой разработки программных продуктов

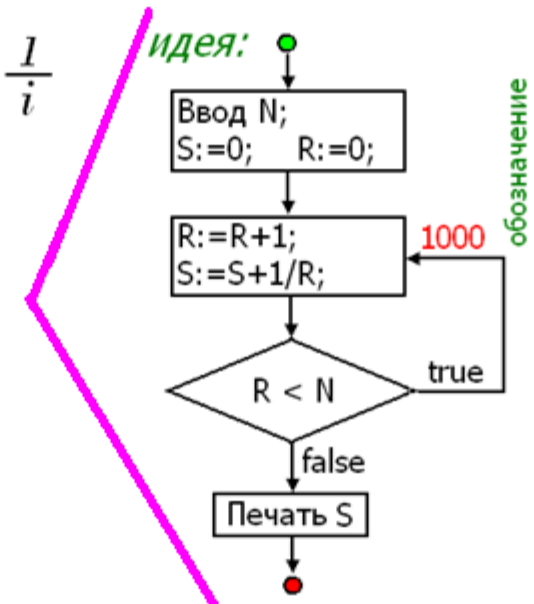


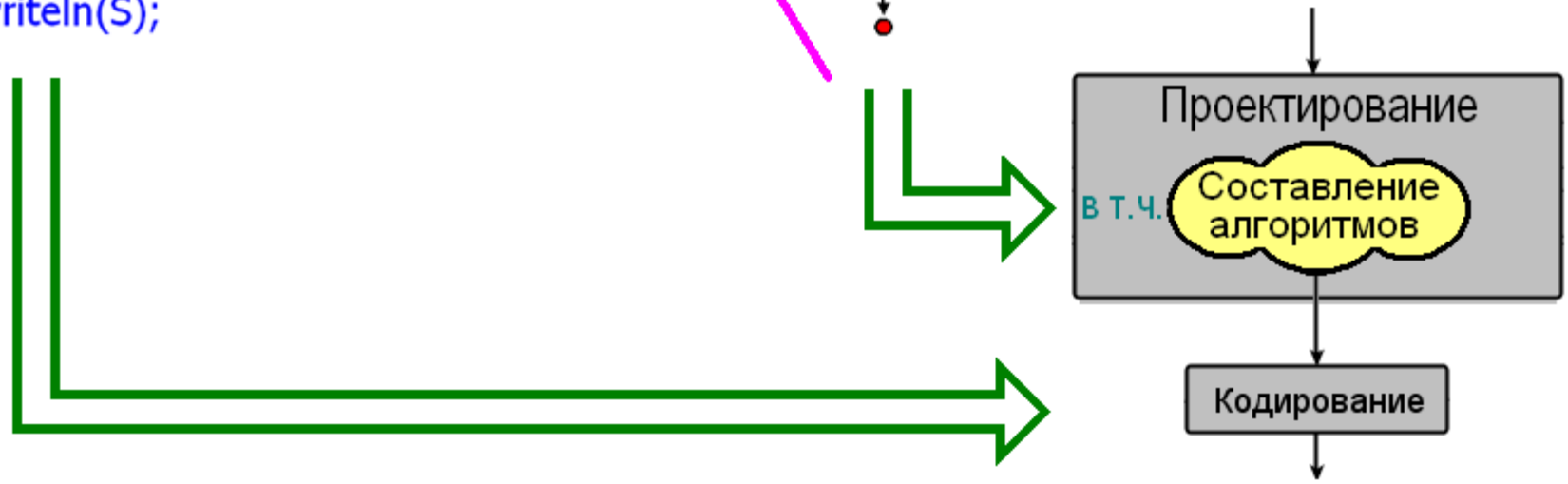
Отладка и тестирование

упрощенное введение

```
program Sep02;
label 1000;
var S,R : real;
      N : integer;
begin   write('=>'); readln(N);
        R:=0;
        S:=0;
1000:   R:=R+1;
        S:=S+1/R;
        if round(R) < N then goto 1000;
        writeln(S);
end.
```

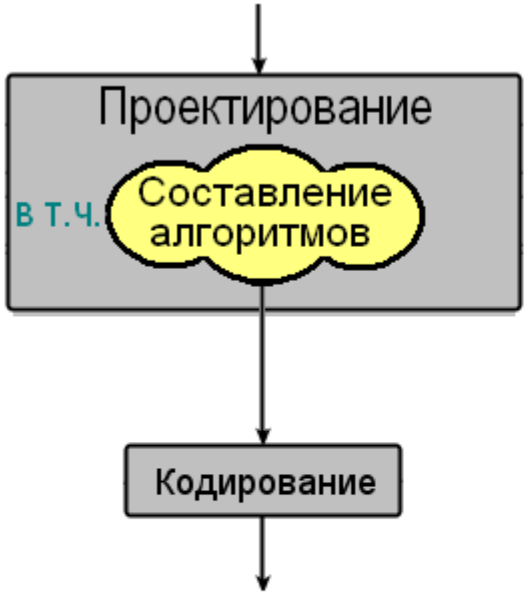
$$S = \sum_{i=1}^{n} \frac{1}{i}$$





Отладка и тестирование

ошибки и дефекты

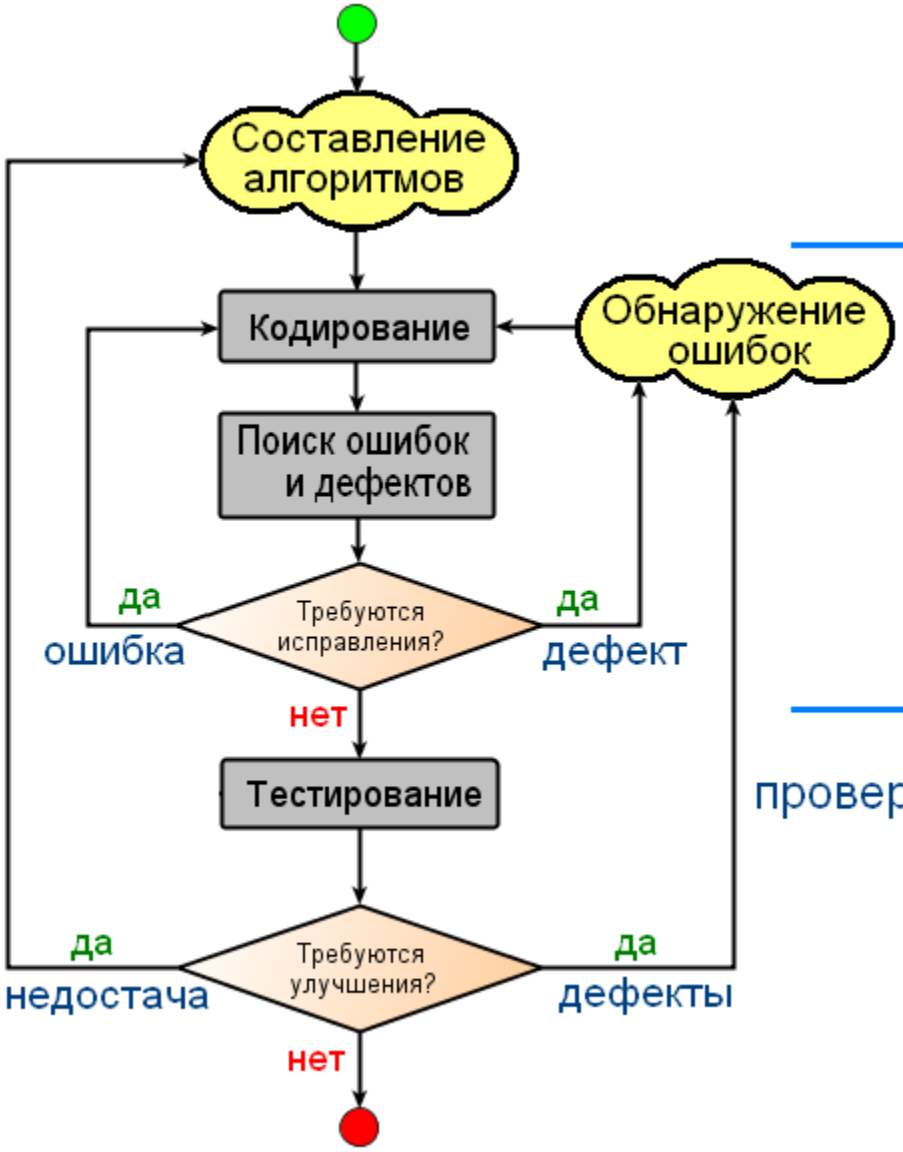


Ошибка — изъян в программе
Который приводит к отклонению
от требований задачи.

Дефект — различие между фактическим и ожидаемым поведением программы

баг/bug

Отладка vs. Тестирование



последовательного

Отладка – процесс

выявления ошибок или дефектов и
внесение исправлений
в текст программы или
в технологию разработки

проверка ПО на соответствие
требованиям задачи

Тесты

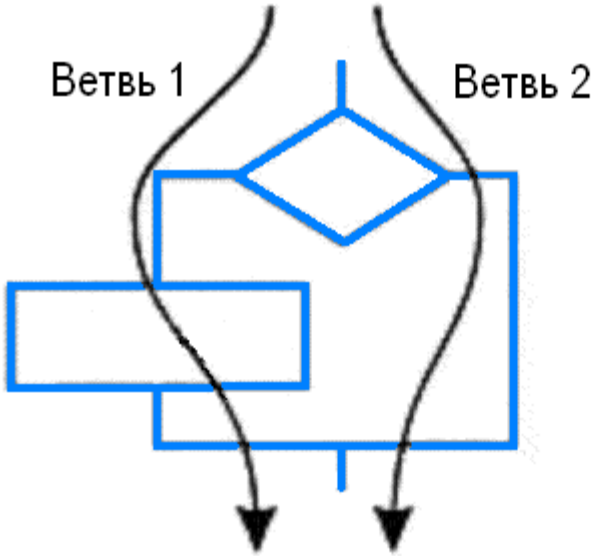
Тест – набор исходных данных, предназначенных для получения заданных ожидаемых результатов.

– описание реакции программы в ответ на обработку проверочных исходных данных

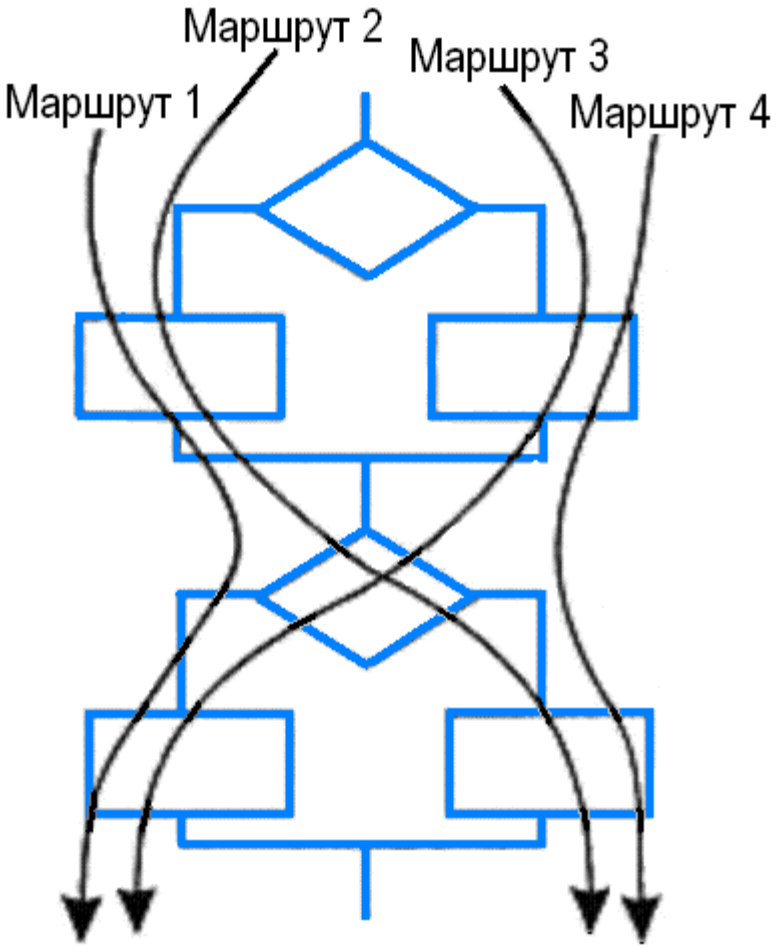
Полнота набора тестов

Набор тестов > Test Suite | Тест сьют

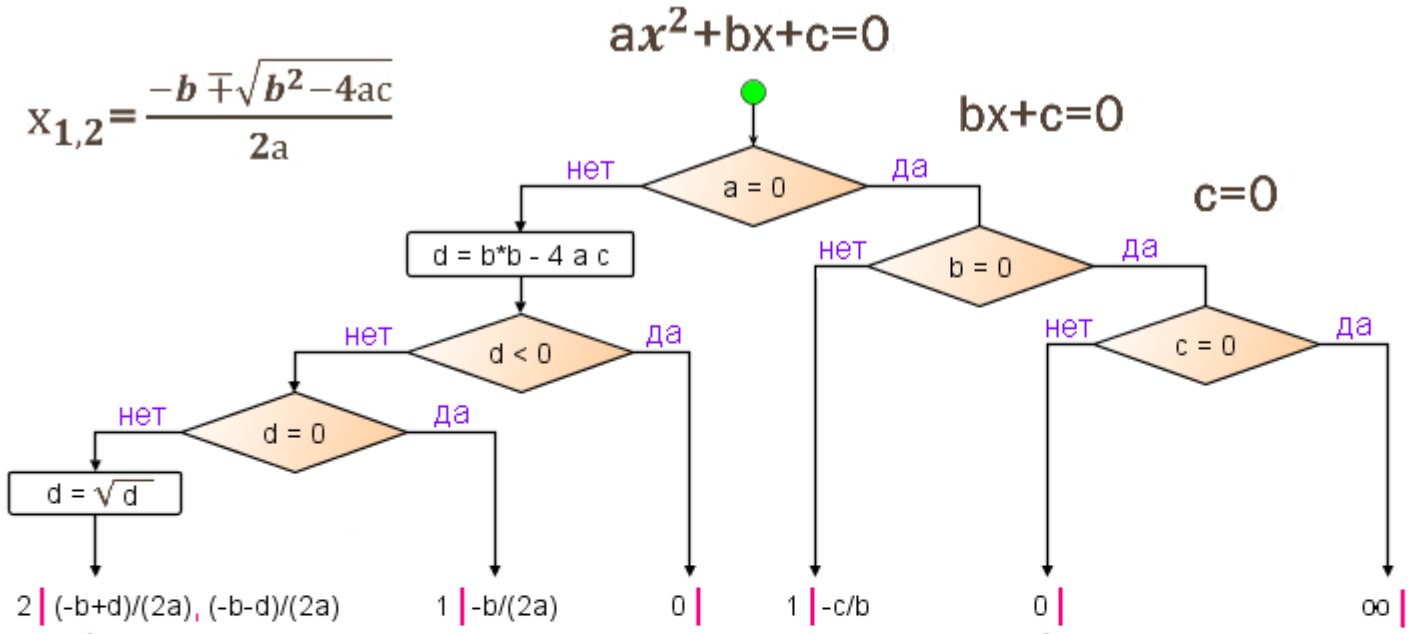
Полнота набора тестов 1/2



Покрытие - операторов
- маршрутов
- данных
- функциональных возможностей
- требований



Полнота набора тестов 2/2



$x_{1,2}=\frac{2c}{-b\mp\sqrt{b^2-4ac}}$

4ac = 0 !!!

a = 0 c = 0 1
a = 0 c ≠ 0 2,3
a ≠ 0 c = 0 ?

	a	b	c		
1.	0	0	0	∞	
2.	0	0	3	0	
3.	0	4	2	1	-0.5
4.	1	1	1	0	
5.	4	4	1	1	-0.5
6.	2	1	-1	2	-1, 0.5
7.	4	2	0	2	0, -0.5

Виды тестирования

По объекту тестирования

Функциональное //functional **Т**
Т производительности//performance
Нагрузочное //load **Т**
Стресс//stress- **Т**
Т стабильности//stability/endurance/soak
Юзабилити//usability **Т**
Т интерфейса пользователя
Т безопасности //security
Т локализации//localization
Т совместимости//compatibility

По времени проведения тестирования

Альфа//alpha- **Т**
Дымовое//smoke **Т**
Т новой функциональности//new feature
Регрессионное//regression **Т**
Приемочное//acceptance **Т**
Бета//beta- **Т**

По степени подготовленности к тестированию

Т по документации//formal
Интуитивное//ad hoc **Т**

По знанию исходного кода

Т чёрного ящика//blackbox
Т белого ящика//whitebox
Т серого ящика//greybox

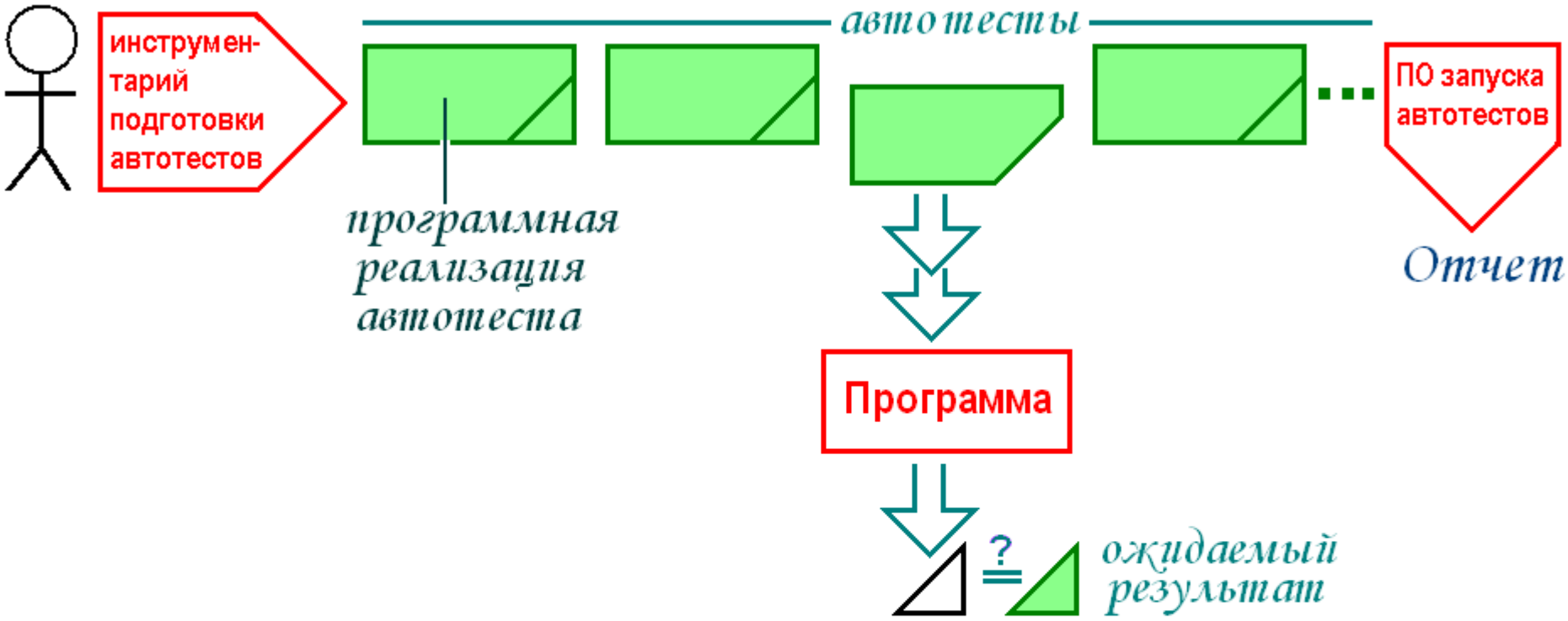
По субъекту тестирования

Альфа//alpha -тестировщик
Бета//beta -тестировщик

По признаку позитивности сценариев

Позитивное//positive **Т**
Негативное//negative **Т**

Инструментарий тестирования



⇨ DEVOPS

Обзор лекции No.11

Нисходящее программирование

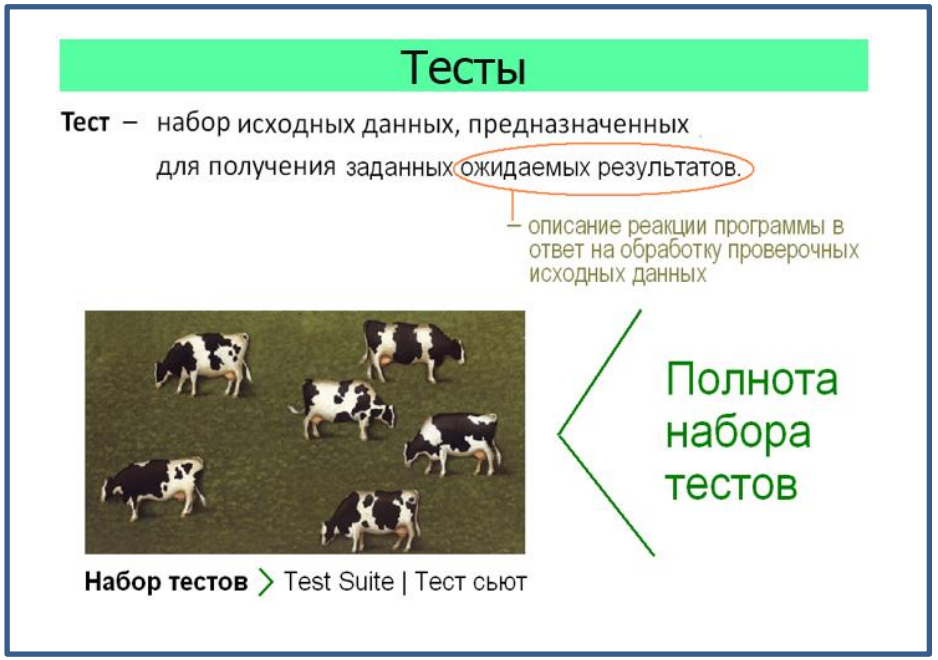
Отладка программ

Тестирование программ

Полнота набора тестов

--- Конец лекции No. 11 ---

Задача*



Дано. Программа COUT, которая в заданной синтаксически корректной ПАСКАЛЬ-программе удаляет все комментарии.

Требуется. Построить полную систему тестов для программы COUT.