

Разбор задания №2 (11 кл)

Робот-маляр

Постановка задачи

На завод купили самые новые модели роботов-маляров. Они умеют закрашивать только треугольники по заданным координатам.

Перед каждым запуском в них надо заливать X банок краски, для того чтобы закрасить заданный треугольник из расчета 1 банка краски на 1 квадратный метр.

Написать программу, которая вычисляет, сколько банок краски необходимо залить роботу-маляру, чтобы закрасить треугольник по заданным координатам.

Постановка задачи

- Входные данные

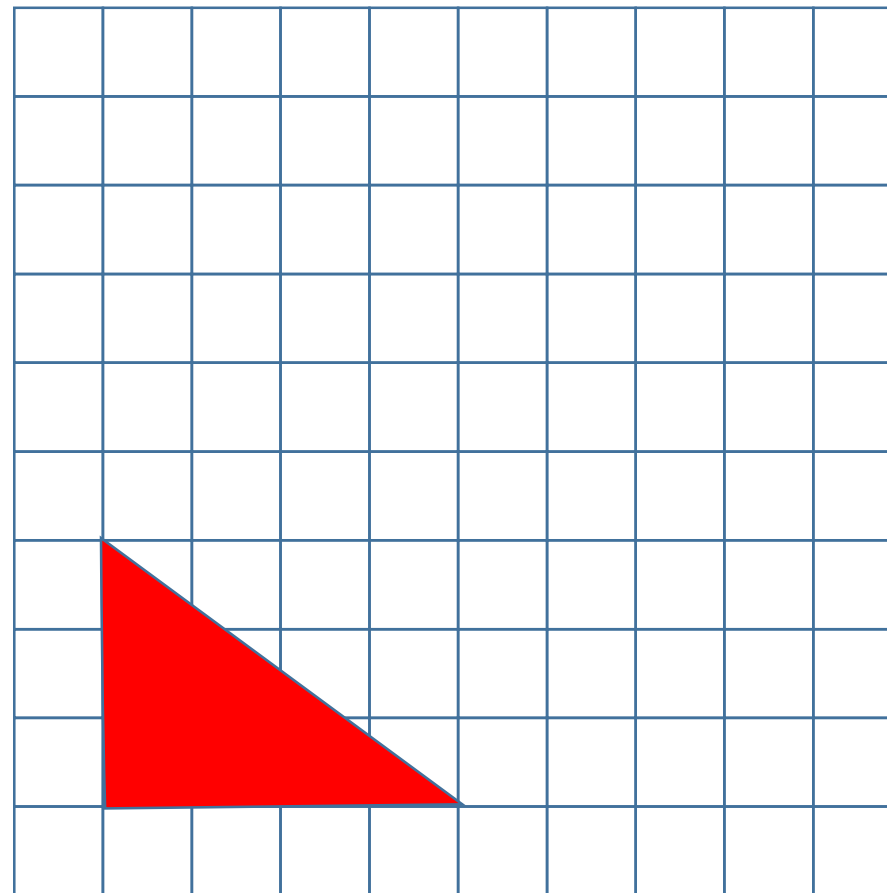
- Программа получает на вход координаты трех точек на плоскости. Координаты каждой точки (два числа в метрах) подаются каждое с новой строки. Все координаты являются целыми числами от -100 до 100 включительно.

- Выходные данные

- Программа выводит объем краски из учета, что на 1 квадратный метр робот тратит 1 банку краски (с округлением в большую сторону).
- Если по введенным точкам нельзя построить треугольник, программа выводит "NOT TRIANGLE"
 - Когда это происходит?
- Когда входные данные некорректны (координаты больше 100 или меньше -100, либо не являются целыми числами), программа должна выдавать сообщение об ошибке "ERROR".

Примеры

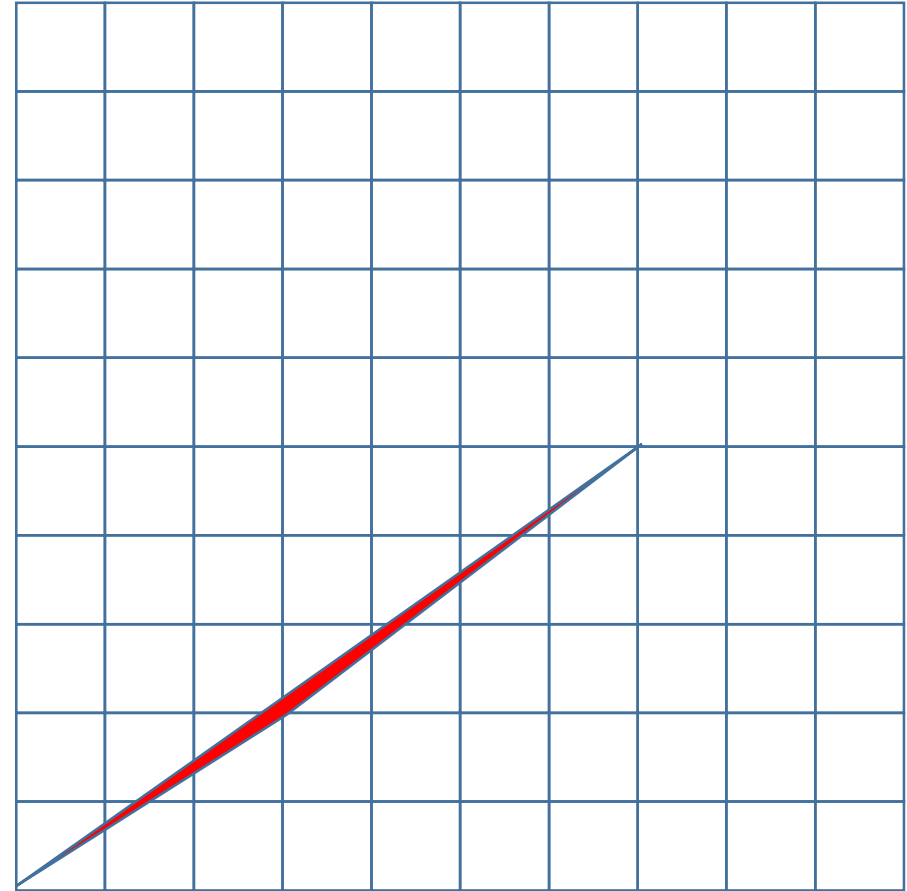
$(1;1), (5;1), (1;4) \rightarrow 6$



Примеры

$(1;1), (5;1), (1;4) \rightarrow 6$

$(0;0), (7;5), (3;2) \rightarrow 1$

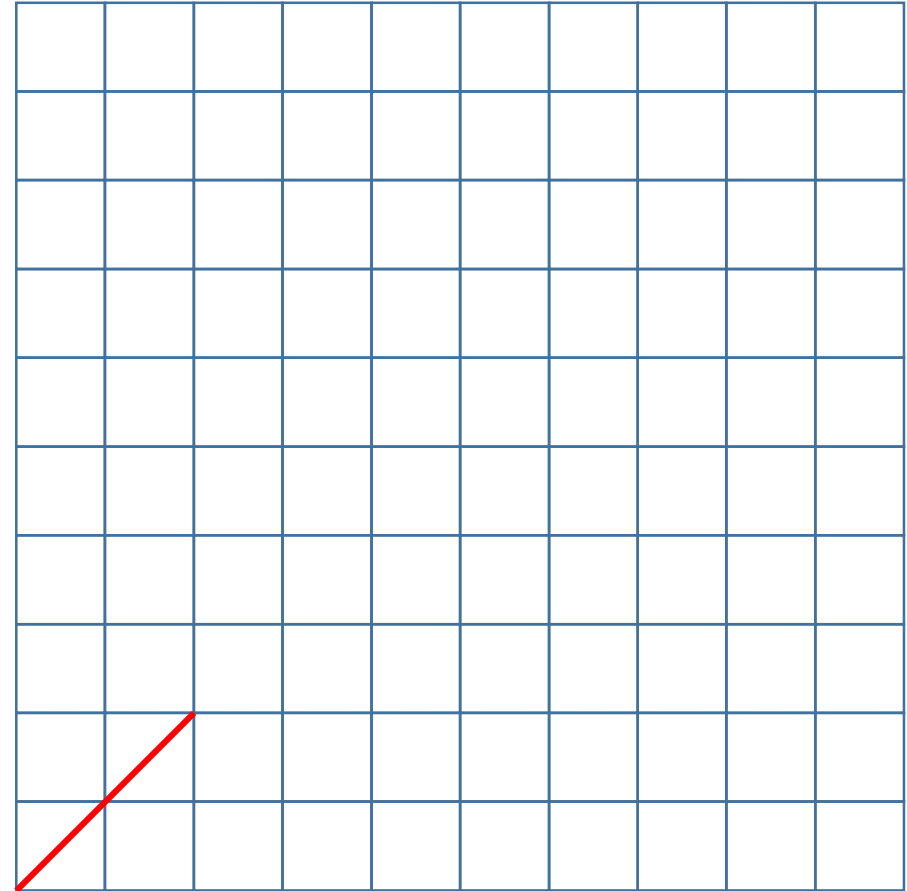


Примеры

$(1;1), (5;1), (1;4) \rightarrow 6$

$(0;0), (7;5), (3;2) \rightarrow 1$

$(0;0), (1;1), (2;2) \rightarrow$ NOT TRIANGLE



Примеры

$(1;1), (5;1), (1;4) \rightarrow 6$

$(0;0), (7;5), (3;2) \rightarrow 1$

$(0;0), (1;1), (2;2) \rightarrow \text{NOT TRIANGLE}$

$(1;-500), (1;-1), (2;1) \rightarrow \text{ERROR}$

Примеры

$(1;1), (5;1), (1;4) \rightarrow 6$

$(0;0), (7;5), (3;2) \rightarrow 1$

$(0;0), (1;1), (2;2) \rightarrow \text{NOT TRIANGLE}$

$(1;-500), (1;-1), (2;1) \rightarrow \text{ERROR}$

$(1;5), (F;6), (2;3) \rightarrow \text{ERROR}$

Задачи, которые нужно решить

- Чтение координат трех вершин
- Проверка исходных данных
- Вычисление площади треугольника
- Вывод результата

Способы нахождения площади треугольника (<https://ru.wikipedia.org/wiki/Треугольник>)

Площадь треугольника связана с его основными элементами следующими соотношениями.

$$1. S_{\triangle ABC} = \frac{1}{2}bh_b, \text{ так как } h_b = a \sin \gamma, \text{ то:}$$

$$2. S_{\triangle ABC} = \frac{1}{2}ab \sin \gamma$$

$$3. S_{\triangle ABC} = \frac{abc}{4R}$$

$$4. S_{\triangle ABC} = \sqrt{p(p-a)(p-b)(p-c)} = \frac{1}{4} \sqrt{(a+b+c)(b+c-a)(a+c-b)(a+b-c)} \text{ — формула Герона}$$

$$5. S_{\triangle ABC} = \frac{a^2 \sin \beta \sin \gamma}{2 \sin \alpha}$$

$$6. S_{\triangle ABC} = 2R^2 \sin \alpha \sin \beta \sin \gamma$$

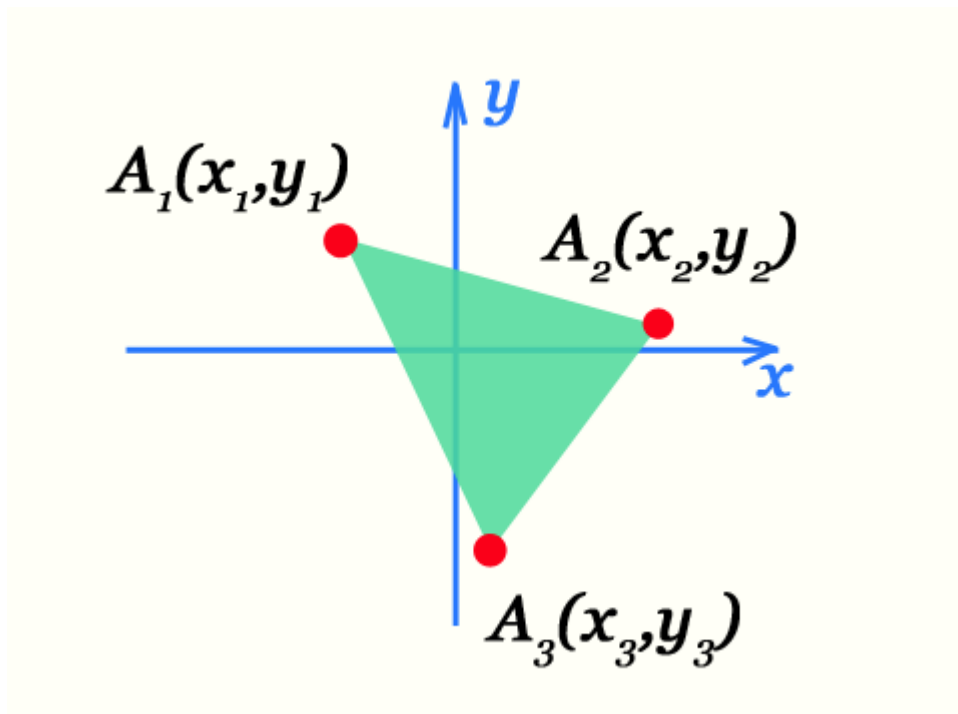
$$7. S_{\triangle ABC} = \frac{c^2}{2(\operatorname{ctg} \alpha + \operatorname{ctg} \beta)}$$

$$8. S_{\triangle ABC} = \frac{1}{2}(\vec{CA} \wedge \vec{CB}) = \frac{1}{2}(x_A - x_C)(y_B - y_C) - \frac{1}{2}(x_B - x_C)(y_A - y_C) \text{ — ориентированная площадь треугольника.}$$

$$9. S_{\triangle ABC} = \frac{1}{\sqrt{(\frac{1}{h_a} + \frac{1}{h_b} + \frac{1}{h_c})(\frac{1}{h_c} + \frac{1}{h_b} - \frac{1}{h_a})(\frac{1}{h_a} + \frac{1}{h_c} - \frac{1}{h_b})(\frac{1}{h_a} + \frac{1}{h_b} - \frac{1}{h_c})}} \text{ — см. Аналоги формулы Герона}$$

Выбранная формула вычисления площади треугольника

$$S = \frac{1}{2} |(x_1 - x_3)(y_2 - y_3) - (x_2 - x_3)(y_1 - y_3)|$$



При целочисленных координатах площадь нам понадобится либо целое число банок, либо останется ровно 0.5 банки краски

Простейшее решение

```
PROGRAM RobotPainter;  
VAR  
    X1, Y1, X2, Y2, X3, Y3: INTEGER;  
BEGIN  
    READLN(X1);  
    READLN(Y1);  
  
    READLN(X2);  
    READLN(Y2);  
  
    READLN(X3);  
    READLN(Y3);  
  
    WRITELN(ABS((X1-X3)*(Y2-Y3)-(Y1-Y3)*(X2-X3))*0.5:0:0)  
END.
```

Анализ решения

- Решена часть задачи
 - Ввод данных
 - Вывод площади
 - Проходит часть «хороших» тестов
- Что не сделано
 - Обработка ошибок
 - Проверка, является ли фигура треугольником

Проверка на «треугольность»

```
PROGRAM RobotPainter;
VAR
  S, DoubleS, X1, Y1, X2, Y2, X3, Y3: INTEGER;
BEGIN
  READLN(X1);
  READLN(Y1);
  READLN(X2);
  READLN(Y2);
  READLN(X3);
  READLN(Y3);
  DoubleS := ABS((X1-X3)*(Y2-Y3)-(Y1-Y3)*(X2-X3));
  S := (DoubleS + 1) DIV 2;
  IF S > 0
    THEN
      WRITELN(S)
    ELSE
      WRITELN('NOT TRIANGLE')
END.
```

Анализ решения

- Проходит все «хорошие» тесты
- Все еще нет проверки ошибок ввода

Проверка ввода

```
CONST
```

```
  MaxValue = 100;
```

```
  MinValue = -100;
```

```
FUNCTION ReadIntegerInRange(VAR Value: INTEGER): BOOLEAN;
```

```
VAR
```

```
  Buffer: STRING;
```

```
  ErrorCode: INTEGER;
```

```
BEGIN
```

```
  READLN(Buffer);
```

```
  VAL(Buffer, Value, ErrorCode);
```

```
  IF ErrorCode <> 0
```

```
    THEN
```

```
      ReadIntegerInRange := FALSE
```

```
    ELSE
```

```
      ReadIntegerInRange := (Value >= MinValue) AND (Value <= MaxValue)
```

```
END;
```


VAR

...

IsValid: BOOLEAN;

BEGIN

IsValid := ReadIntegerInRange(X1);

IsValid := IsValid AND ReadIntegerInRange(Y1);

IsValid := IsValid AND ReadIntegerInRange(X2);

IsValid := IsValid AND ReadIntegerInRange(Y2);

IsValid := IsValid AND ReadIntegerInRange(X3);

IsValid := IsValid AND ReadIntegerInRange(Y3);

IF NOT IsValid

THEN

WRITELN('ERROR')

ELSE

BEGIN

DoubleS := ABS((X1-X3)*(Y2-Y3)-(Y1-Y3)*(X2-X3));

S := (DoubleS + 1) DIV 2;

IF S > 0

THEN

WRITELN(S)

ELSE

WRITELN('NOT TRIANGLE')

END

END.

Спустя несколько минут после сдачи программы на проверку



Программа прошла
100% хороших и
90% плохих тестов

Улучшенная проверка ввода

```
FUNCTION ReadIntegerInRange(VAR Value: INTEGER): BOOLEAN;  
VAR  
    Buffer: STRING;  
    ErrorCode: INTEGER;  
BEGIN  
    IF EOF  
    THEN  
        ReadIntegerInRange := FALSE  
    ELSE  
        BEGIN  
            READLN(Buffer);  
            VAL(Buffer, Value, ErrorCode);  
            IF ErrorCode <> 0  
            THEN  
                ReadIntegerInRange := FALSE  
            ELSE  
                ReadIntegerInRange := (Value >= MinValue) AND (Value <= MaxValue)  
            END  
        END  
END;  
END;
```

Анализ решения

- Проходит часть все «хорошие» тесты
- Проходит все «плохие»
- Дублируется код чтения координат вершин
- Код слишком низкоуровневый

```
IsValid := ReadIntegerInRange(X1);  
IsValid := IsValid AND ReadIntegerInRange(Y1);  
IsValid := IsValid AND ReadIntegerInRange(X2);  
IsValid := IsValid AND ReadIntegerInRange(Y2);  
IsValid := IsValid AND ReadIntegerInRange(X3);  
IsValid := IsValid AND ReadIntegerInRange(Y3);
```



Считываем координаты
первой вершины

Считываем координаты
второй вершины

Считываем координаты
третьей вершины

Чтение координат вершины

{Читает координаты точки из INPUT}

{Возвращает TRUE, если координаты входят в допустимый диапазон}

{Иначе возвращает FALSE}

```
FUNCTION ReadPoint(VAR X, Y: INTEGER): BOOLEAN;
```

```
BEGIN
```

```
    ReadPoint := ReadIntegerInRange(X) AND ReadIntegerInRange(Y);
```

```
END;
```

```
BEGIN
  IsValid := ReadPoint(X1, Y1) AND ReadPoint(X2, Y2) AND ReadPoint(X3, Y3);
  IF NOT IsValid
  THEN
    WRITELN('ERROR')
  ELSE
    BEGIN
      DoubleS := ABS((X1-X3)*(Y2-Y3)-(Y1-Y3)*(X2-X3));
      S := (DoubleS + 1) DIV 2;
      IF S > 0
      THEN
        WRITELN(S)
      ELSE
        WRITELN('NOT TRIANGLE')
      END
    END
  END.
```

Анализ решения

- Код стал более выразительным
- По-прежнему у нас 6 независимых переменных
- Программа оперирует не с вершинами треугольника, а с отдельными координатами

Вводим тип данных «Точка»

TYPE

{Тип данных Точка}

Point = RECORD

 X, Y: INTEGER;

END;

Было



x1



y2



x3



x2

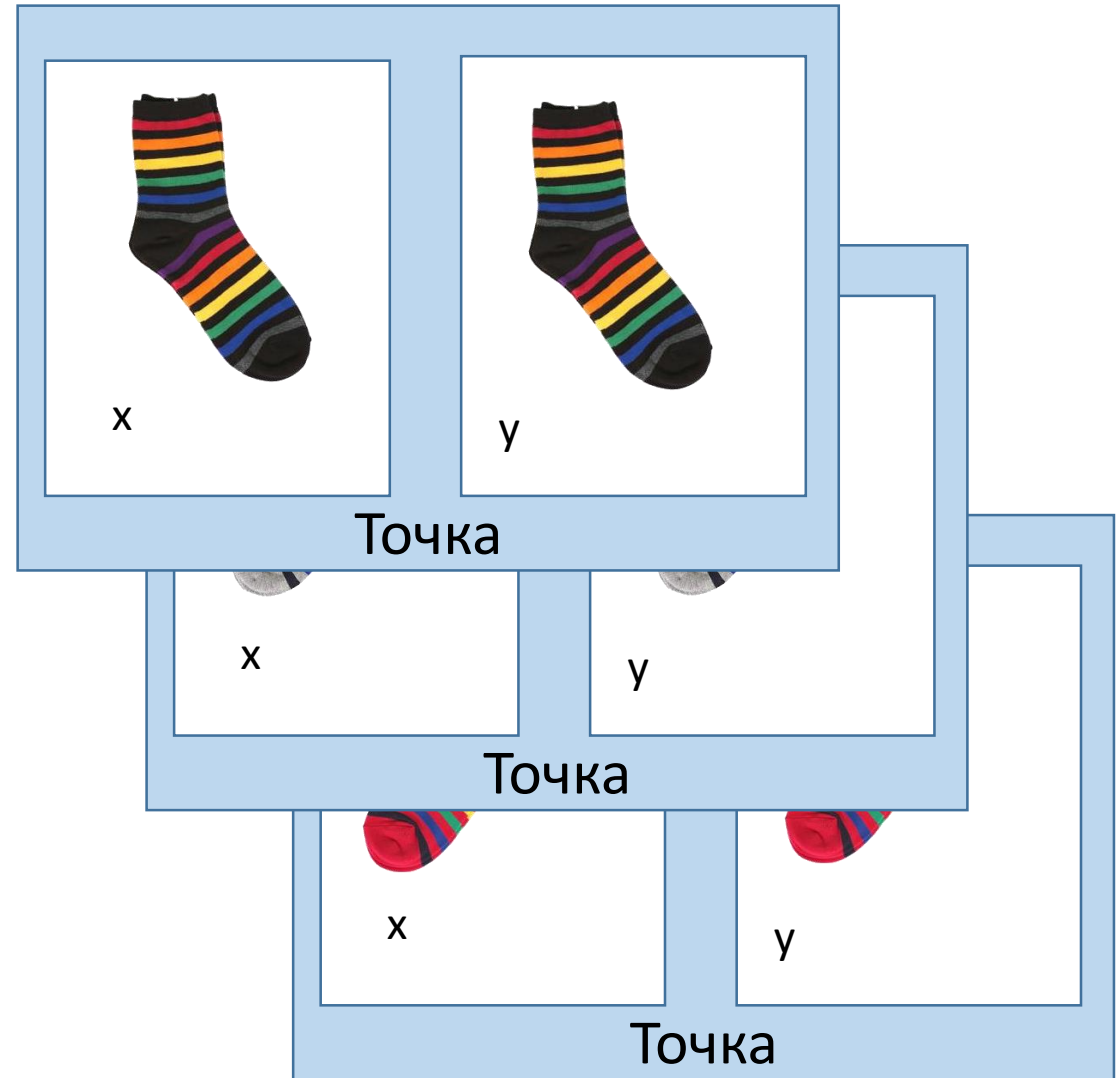


y3



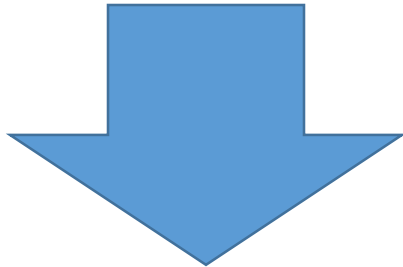
y1

Стало



Дорабатываем ReadPoint

```
FUNCTION ReadPoint(VAR X, Y: INTEGER): BOOLEAN;  
BEGIN  
  ReadPoint := ReadIntegerInRange(X) AND ReadIntegerInRange(Y);  
END;
```



```
FUNCTION ReadPoint(VAR P: Point): BOOLEAN;  
BEGIN  
  ReadPoint := ReadIntegerInRange(P.X) AND ReadIntegerInRange(P.Y);  
END;
```

VAR

...

P1, P2, P3: Point;

BEGIN

IsValid := **ReadPoint(P1) AND ReadPoint(P2) AND ReadPoint(P3);**

IF NOT IsValid

THEN

WRITELN('ERROR')

ELSE

BEGIN

DoubleS := **ABS((P1.X-P3.X)*(P2.Y-P3.Y)-(P1.Y-P3.Y)*(P2.X-P3.X));**

S := (DoubleS + 1) DIV 2;

IF S > 0

THEN

WRITELN(S)

ELSE

WRITELN('NOT TRIANGLE')

END

END.

Анализ решения

- Программа теперь оперирует точками, а не целыми числами
- Код вычисления площади треугольника труден для понимания

```
DoubleS := ABS((P1.X-P3.X)*(P2.Y-P3.Y)-(P1.Y-P3.Y)*(P2.X-P3.X));  
S := (DoubleS + 1) DIV 2;
```

Выделяем функцию вычисления количества банок краски

```
{Вычисляет количество банок краски, нужных для закраски треугольника}
FUNCTION CalculateAmountOfPaint(P1, P2, P3: Point): INTEGER;
VAR
    DoubleS: INTEGER;
BEGIN
    DoubleS := ABS((P1.X-P3.X)*(P2.Y-P3.Y)-(P1.Y-P3.Y)*(P2.X-P3.X));
    CalculateAmountOfPaint := (DoubleS + 1) DIV 2
END;
```

VAR

...

AmountOfPaint: INTEGER;

BEGIN

IsValid := ReadPoint(P1) AND ReadPoint(P2) AND ReadPoint(P3);

IF NOT IsValid

THEN

 WRITELN('ERROR')

ELSE

 BEGIN

AmountOfPaint := CalculateAmountOfPaint(P1, P2, P3);

 IF **AmountOfPaint** > 0

 THEN

 WRITELN(**AmountOfPaint**)

 ELSE

 WRITELN('NOT TRIANGLE')

 END

END.

Анализ решения

- Код программы стал понятнее
- Программа оперирует сущностями предметной области
- Как можно улучшить программу?
 - Ввести тип «Треугольник»

```
VAR
```

```
...
```

```
T: Triangle;
```

```
BEGIN
```

```
IF ReadTriangle(T)
```

```
THEN
```

```
    BEGIN
```

```
        AmountOfPaint := CalculateAmountOfPaint(T);
```

```
        IF AmountOfPaint > 0
```

```
        THEN
```

```
            WRITELN(AmountOfPaint)
```

```
        ELSE
```

```
            WRITELN('NOT TRIANGLE')
```

```
        END
```

```
    ELSE
```

```
        WRITELN('ERROR')
```

```
END.
```


Выводы

- Не пытайтесь написать всю программу сразу
 - Начните с простейшей работающей версии
- Оперируйте сущностями предметной области
- Прячьте сложный код внутри функций с понятными именами
- На каждом шаге проводите анализ своего решения
- Сперва обдумайте решение, а потом пишите код

Разбор задач по составлению
тестового набора

Программа для подсчёта количества и суммы чётных и нечётных чисел

- Дана программа, которая считывает до 10 целых чисел и выводит сумму чётных и сумму нечётных, если они есть
 - Если никаких чисел нет, вывести NO NUMBERS
- Ограничения:
 - Числа должны быть в диапазоне от -999 до +999 включительно
 - Если не так – вывести ERROR
 - Чисел не должно быть больше 10
 - Если не так – вывести ERROR

Требования задания

- Предоставить набор файлов с тестовыми данными для выявления правильных и неправильных программ
 - Правильно написанная программа должна пройти все тесты успешно
 - Неправильные программы должны провалить хотя бы один из тестов
- Правильная программа реализует все требования задания
- Неправильная программа либо делает не всё, что нужно, либо делает то, что не нужно

N	input<N>.txt	output<N>.txt	Комментарий
1	<пусто>	NO NUMBERS	Нет чисел – сообщаем об этом
2	42	Sum of 1 even numbers is 42	Только чётные
3	57	Sum of 1 odd numbers is 57	Только нечётные
4	1 2 3 4 5	Sum of 2 even numbers is 6 Sum of 3 odd numbers is 9	Чётные и нечётные Многострочный ввод
5	1000 1	ERROR	Ближайшее число за верхней границей
6	-1000 1	ERROR	Ближайшее число за нижней границей
7	-999	Sum of 1 odd numbers is -999	Наименьшее допустимое число
8	999	Sum of 1 odd numbers is 999	Наибольшее допустимое число
9	1 2 3 4 5 6 7 8 9 10 11	ERROR	Введено более 10 чисел
10	Hello	ERROR	Не целое число
11	1.5	ERROR	Не целое число

Программа для замены текста

- Дана программа для замены текста
 - Сперва вводится искомая строка, потом строка-заменитель
 - Программа должна вывести результат замены всех вхождений искомой строки на строку-заменитель во всех последующих строках
- Требования
 - Поиск является чувствительным к регистру символов
 - Если искомая строка пустая, текст должен копироваться без замены
 - Если строка заменитель пустая – вхождения искомой строки удаляются
 - Если не введена искомая строка или строка-заменитель программа не должна ничего выводить

Требования задания

- Предоставить набор файлов с тестовыми данными для выявления правильных и неправильных программ
 - Правильно написанная программа должна пройти все тесты успешно
 - Неправильные программы должны провалить хотя бы один из тестов
- Правильная программа реализует все требования задания
- Неправильная программа либо делает не всё, что нужно, либо делает то, что не нужно

N	input<N>.txt	output<N>.txt	Комментарий
1	<пусто>	<пусто>	Не введена искомая строка
2	Hello	<пусто>	Не введена строка-заменитель
3	dog cat a quick brown fox jumps over the lazy dog	a quick brown fox jumps over the lazy cat	
4	Hello Some text	Some text	Искомая строка пустая
5	mama ma mamama	mama	Заменённый текст повторно не должен обрабатываться
6	Foot Basket Football players like football.	Basketball players like football.	Поиск ведётся с учётом регистра
7	hello hello world, you are amazing! hello kitty!	world, you are amazing! kitty!	Заменяются все вхождения Пустая строка-заменитель удаляет из исходной строки искомый текст
8	hello hello hello world, you are amazing!	hello world, you are amazing!	Искомый текст и текст-заменитель совпадают

Разбор задания №2 (9 кл)

Робот-художник

VAR

AX, AY, BX, BY, CX, CY, AB2, AC2, BC2: INTEGER;

IsValid, IsValidTriangle, IsRightTriangle: BOOLEAN;

BEGIN

IsValid := ReadIntegerInRange(AX);

IsValid := IsValid AND ReadIntegerInRange(AY);

...

IF IsValid

THEN

BEGIN

AB2 := SQR(AX - BX) + SQR(AY - BY);

BC2 := SQR(BX - CX) + SQR(BY - CY);

AC2 := SQR(AX - CX) + SQR(AY - CY);

IsRightTriangle := (AB2 = BC2 + AC2) OR (BC2 = AB2 + AC2) OR (AC2 = AB2 + BC2);

IsValidTriangle := (AB2 > 0) AND (BC2 > 0) AND (AC2 > 0);

IF IsRightTriangle AND IsValidTriangle

THEN

WRITELN('YES')

ELSE

WRITELN('NO')

END

ELSE

WRITELN('ERROR')

END.