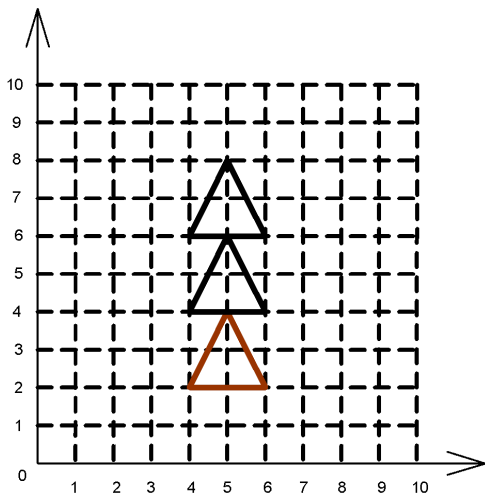


Использование вспомогательных алгоритмов

Теоретический материал

Продемонстрировать рисунок, на котором изображены повторяющиеся объекты (не менее 3-5) и фрагмент программы, изображающий один объект.

<i>Рисунок</i>	<i>Фрагмент программы, изображающий нижний треугольник</i>
	<pre> ToPoint (4, 2); PenDown; OnVector (1, 2); OnVector (1, -2); OnVector (-2, 0); PenUp; </pre>

Организовать обсуждение:

- Как быстро получить требуемое изображение, имея фрагмент программы? (*скопировать его*)

Получим программу:

Program Elka;

uses Drawman;

begin

Field(9,9);

ToPoint (1,1);

PenDown;

OnVector(3,2);

OnVector(3,-2);

OnVector(-6,0);

PenUp;

} 1-й треугольник

ToPoint (1,3);

PenDown;

OnVector(3,2);

OnVector(3,-2);

OnVector(-6,0);

PenUp;

} 2-й треугольник

ToPoint (1,5); PenDown; OnVector(3,2); OnVector(3,-2); OnVector(-6,0); PenUp;	}	3-й треугольник
--	---	-----------------

End.

- Укажите недостатки данного способа (*программа становится большой и в ней присутствуют повторяющиеся части*)

Этого недостатка можно избежать.

В программировании существует возможность сокращать запись программ, в которых многократно повторяется один и тот же набор команд. Такой набор команд оформляют в виде **подпрограммы** (вспомогательного алгоритма), другое название – **процедура**. Процедуре дается имя, которое указывается в основной программе для вызова данной процедуры.

Таким образом, приведенную выше программу можно записать в следующем виде:

Program Elka; uses Drawman;		
procedure Triangle; begin PenDown; OnVector (3,2); OnVector (3,-2); OnVector (-6,0); PenUp; end;	}	Процедура с именем Triangle Последовательность команд, которая используется для вычерчивания каждого треугольника, оформим в виде процедуры с именем Triangle .
begin Field (9,9); ToPoint (1,1); Triangle ; ToPoint (1,3); Triangle ; ToPoint (1,5); Triangle ;	}	Основная программа В основной программе вместо каждой повторяющейся последовательности команд запишем имя процедуры – Triangle .

End.

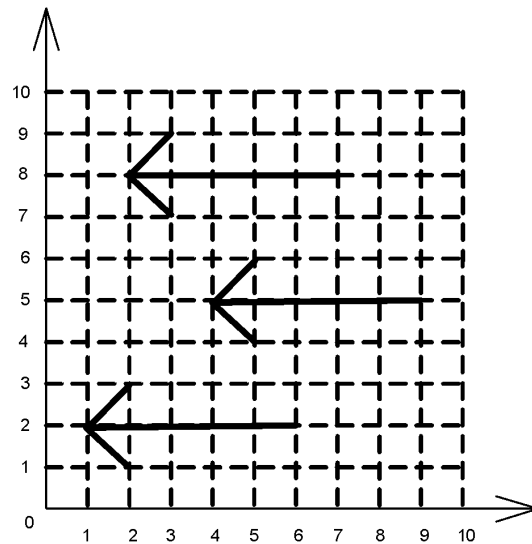
Практический материал

Задание 1. Составьте программу построения следующего изображения с использованием процедуры **Strela**.

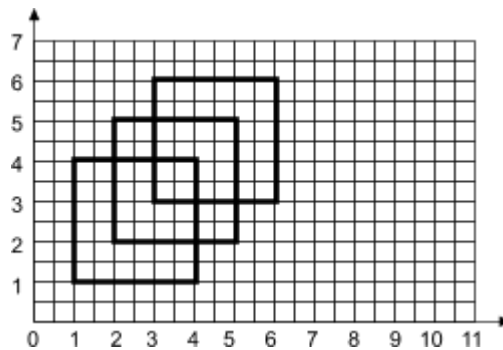
```

procedure Strela;
begin
  PenDown;
  OnVector (-1, -1);
  OnVector (1, -1);
  OnVector (-1, 1);
  OnVector (5, 0);
  PenUp;
end;

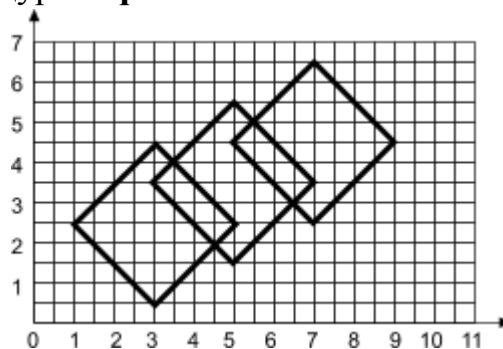
```



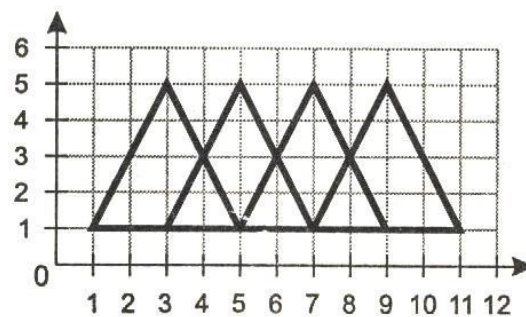
Задание 2. Составьте программу построения следующего изображения с использованием процедуры **Square1**.



Задание 3. Составьте программу построения следующего изображения с использованием процедуры **Square2**.



Задание 4. Составьте программу построения следующего изображения с использованием процедуры **Triangle**.



Задание 5. Составьте программу построения следующего изображения с использованием процедуры **Flag**.

