

**ВЕТВЛЕНИЕ И  
ПОСЛЕДОВАТЕЛЬНАЯ  
ДЕТАЛИЗАЦИЯ  
АЛГОРИТМА**

# ЦИКЛ

Решим задачу: исходное положение ГРИС – у левого края поля, направление – на восток. Требуется нарисовать горизонтальную линию через весь экран.

Задачу можно решить, написав 15 раз команду **шаг** (если поперёк поля рисунка 15 шагов).

# ЦИКЛ

## Короткий вариант:

пока впереди не край повторять

нц

шаг

кц

Здесь использована команда, которая называется **циклом**.

# ЦИКЛ

Формат команды цикла:

пока <условие> повторять

нц

<тело цикла>

кц

Служебное слово **нц** обозначает начало цикла,  
**кц** – конец цикла.

# ВЕТВЛЕНИЕ

Познакомимся еще с одной командой ГРИС.  
Она называется командой **ветвления**.  
Формат команды ветвления такой:

```
если <условие>  
    то <серия 1>  
    иначе <серия 2>  
кв
```

Служебное слово **кв** обозначает *конец*  
*ветвления*.

# ПОЛНОЕ ВЕТВЛЕНИЕ

По-прежнему ГРИС может проверять только два условия: «впереди край?» или «впереди не край?». <Серия> — это одна или несколько следующих друг за другом команд. Если <условие> справедливо, то

выполняется

<серия 1>,

в противном

случае —

<серия 2>.



Рис. 5.12. Блок-схема полного ветвления

Пример показан на рис.

Такое ветвление называется **полным**.

# НЕПОЛНАЯ ФОРМА ВЕТВЛЕНИЯ

В некоторых случаях используется неполная форма команды ветвления. Например:

**если впереди край  
то поворот  
КВ**



Рис. 5.13. Блок-схема неполного ветвления

Неполная команда ветвления имеет следующий формат:

**если <условие>  
то <серия>  
КВ**

Здесь <серия> выполняется, если <условие> справедливо.

# ПРИМЕР ЗАДАЧИ С ДВУХШАГОВОЙ ДЕТАЛИЗАЦИЕЙ

**Задача:** Построить орнамент, состоящий из квадратов, расположенных по краю поля. Исходное положение ГРИС — в верхнем левом углу, направление на юг

Процедуру, рисующую цепочку квадратов от края до края поля, назовем РЯД.  
Процедуру, рисующую один квадрат, назовем КВАДРАТ.

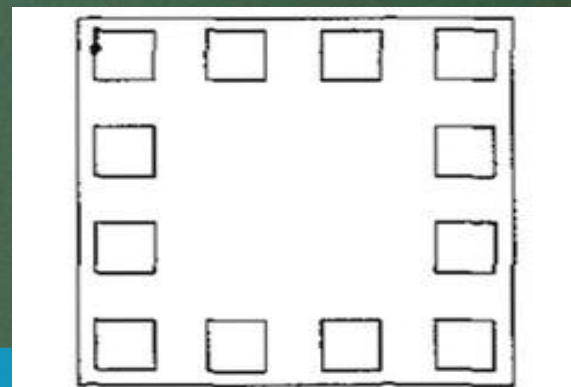


Рис. 5.14. Результат выполнения программы «Орнамент»



# СНАЧАЛА НАПИШЕМ ОСНОВНУЮ ПРОГРАММУ:

программа Орнамент  
нач

сделай РЯД

поворот

сделай РЯД

поворот

сделай РЯД

поворот

сделай РЯД

кОН



# ТЕПЕРЬ НАПИШЕМ ПРОЦЕДУРЫ РЯД И КВАДРАТ:

процедура РЯД	процедура КВАДРАТ
нач	нач
прыжок	шаг
прыжок	поворот
пока впереди не край, повторять	шаг
кц	поворот
сделай КВАДРАТ	шаг
если впереди не край	поворот
то прыжок	шаг
кв	поворот
кц	прыжок
кон	кон

В процедуре РЯД в теле цикла содержится неполное ветвление. Структуру такого алгоритма можно назвать так: цикл с вложенным ветвлением.

# НА РИС. 5.15 ПРИВЕДЕНА БЛОК-СХЕМА ПРОЦЕДУРЫ РЯД:

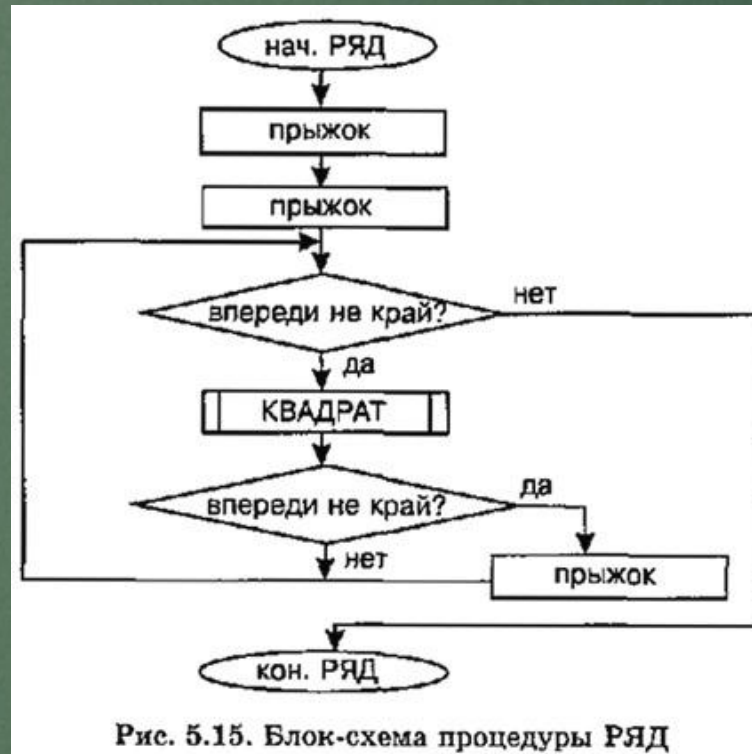
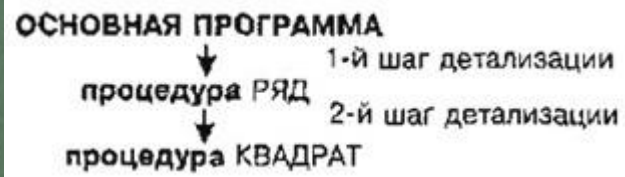


Рис. 5.15. Блок-схема процедуры РЯД

**СОСТАВЛЕНИЕ ЭТОЙ ПРОГРАММЫ  
ПОТРЕБОВАЛО ДВУХ ШАГОВ  
ДЕТАЛИЗАЦИИ АЛГОРИТМА, КОТОРЫЕ  
ВЫПОЛНЯЛИСЬ В ТАКОЙ  
ПОСЛЕДОВАТЕЛЬНОСТИ:**



Теперь вам известны все команды управления графическим исполнителем. Их можно разделить на три группы:

- *простые команды;*
- *команда обращения к процедуре;*
- *структурные команды (команды цикла и ветвления).*

## СКИ графического исполнителя

### Простые команды

шаг

поворот

прыжок

### Обращение к процедуре

сделай <имя процедуры>

### Структурные команды

пока <условие>, повторять

нц

<тело цикла>

кц

если <условие>

то <серия 1>

иначе <серия 2>

кв

## **ЗАДАНИЯ:**

**Путем пошаговой детализации составьте программы управления графическим исполнителем для решения следующих задач:**

- расчертить все поле горизонтальными пунктирными линиями;**
- нарисовать квадраты во всех четырех углах поля;**
- расчертить все поле в клетку со стороной, равной шагу.**

## **ВОПРОСЫ:**

- 1. Что такое пошаговая детализация?**
- 2. Из каких команд могут состоять вспомогательные алгоритмы последнего уровня детализации?**
- 3. Какой формат имеет команда ветвления? Какие действия исполнителя она определяет?**
- 4. Чем отличается полное ветвление от неполного?**