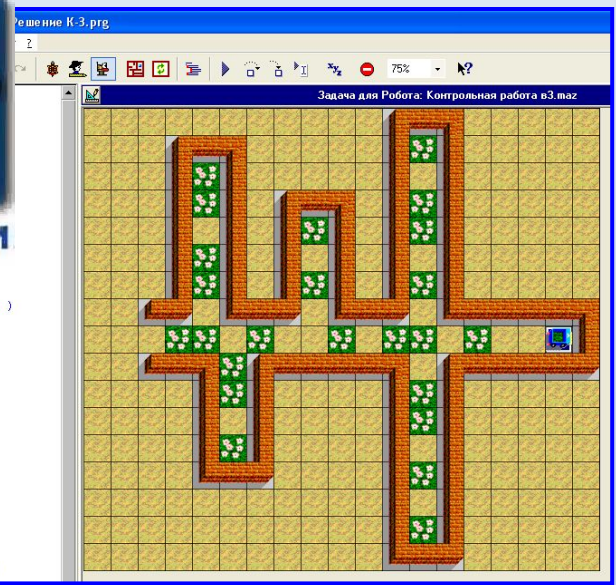




```
налево;  
}  
если ( справа_свободно )  
{  
    направо;  
    Коридор ;  
    направо;  
}  
если ( грядка )  
{  
    посади;  
}  
}  
}  
Коридор  
{  
    пока ( впереди_свободно )
```



Исполнитель РОБОТ

Основные алгоритмические конструкции

Содержание презентации:

- ❖ Система программирования «Исполнители 2.5»
- ❖ Линейные программы
- ❖ Цикл ПОВТОРИ n РАЗ
- ❖ Цикл ПОКА
- ❖ Ветвление
- ❖ Вложенные циклы
- ❖ Процедуры



В курсе используется Си подобная система
программирования «Исполнители 2.5»

Автор: Поляков К.Ю. <http://kpolyakov.spb.ru/>

Занятие 1. Система программирования «Исполнители 2.5». Линейная программа.

The screenshot shows the 'Система "Исполнители"' (System 'Executors') interface. It features a menu bar with 'Файл', 'Правка', 'Настройка', 'Шаблоны', and 'Лабиринт'. Below the menu is a toolbar with various icons for file operations and execution. The main workspace is divided into two panes: 'Программа' (Program) on the left, showing a simple linear code structure with curly braces, and 'Задача для Робота: z1.maz' (Task for the Robot: z1.maz) on the right, displaying a maze environment with a robot icon. The Windows taskbar at the bottom shows the time as 5:33 and several open applications including 'Математика', 'robowlp', 'Слушать радио 1...', 'Исполнитель Роб...', 'Уроки Робот Си', and 'Исполнители'.

Основное меню программы

Окно редактора

Окно для организации ввода и вывода

Задача для робота (обстановка)



Система "Исполнители"

Файл Правка Настройка Шаблоны Лабиринт ?

Типы переменных
Операторы
Функции

Программа

{
}
);

слева_стена
справа_стена
впереди_стена
сзади_стена
слева_клуба
справа_клуба
впереди_клуба
сзади_клуба
слева_свободно
справа_свободно
впереди_свободно
сзади_свободно
грядка
клуба
база

**Ре
ро**

**Восстановить
лабиринт**

**Шаблоны
команд**

**Остановить
выполнение
программы**

Основные команды:

направо; - повернуться на 90 градусов
вправо

налево; - повернуться на 90 градусов влево

кругом; - развернуться кругом (на 180
градусов)

вперед (n); - перейти на n клеток вперед

назад (n); - перейти на n клеток назад

посади; - посадить цветы на грядке, где
стоит Робот

Програ

Начало
программы

ОБОК

Задача для Робота

вперед (1);

налево;

вперед (1);

посади;

вперед (1);

посади;

направо;

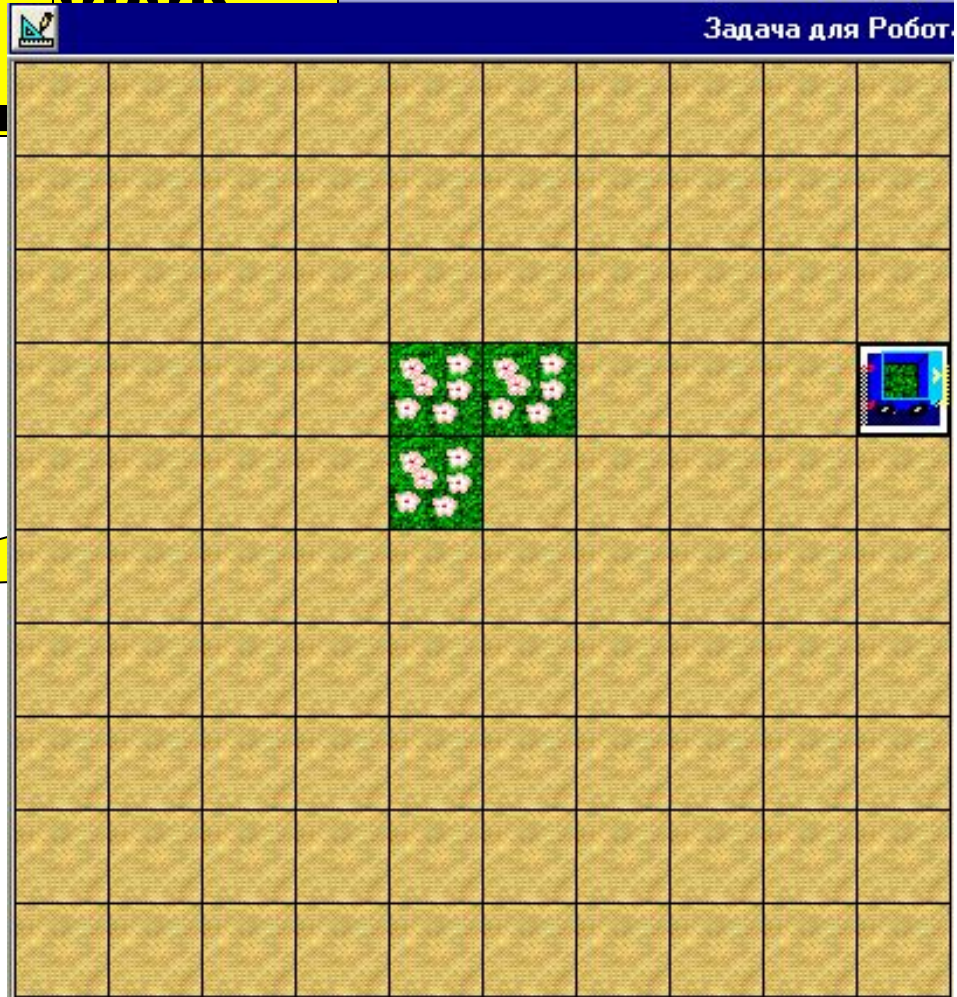
вперед (1);

посади:

впе

Конец

программы



Задача 1. Робот должен посадить цветы на все грядки и дойти до Базы.

Информация



Задача решена верно. Робот выполнил задание и вернулся на базу.

OK

**Задача
решена
успешно**

Ошибка



Задача решена неверно. Обнаружены следующие ошибки:
- Робот не пришел на базу

OK

**Задача не
решена**



Упорядочить программу

Программа

```
{
  | вперед ( 1 );   налево;
  вперед ( 1 );   посади;
    вперед ( 1 );
    посади;           направо;
  вперед ( 1 );   посади;
  вперед ( 4 );
}
```

Программа

```
{
  вперед ( 1 );
  налево;
  вперед ( 1 );
  посади;
  вперед ( 1 );
  посади;
  направо;
  вперед ( 1 );
  посади;
  вперед ( 4 );
}
```

Система "Исполнители"

Файл Правка Настройка Шаблоны Лабиринт ?



Программа

```
{  
  вперед ( 1 );  
  налево;  
  вперед ( 1 );  
  посади;  
  вперед ( 1 );  
  посади;  
  направо;  
  вперед ( 1 );  
  посади;  
  вперед ( 4 );  
}
```

Задача для Робота

**Редактировать
лабиринт**



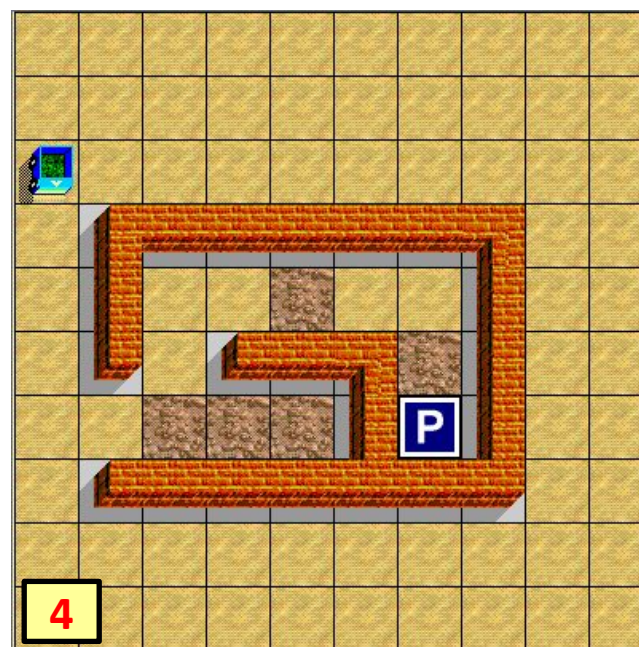
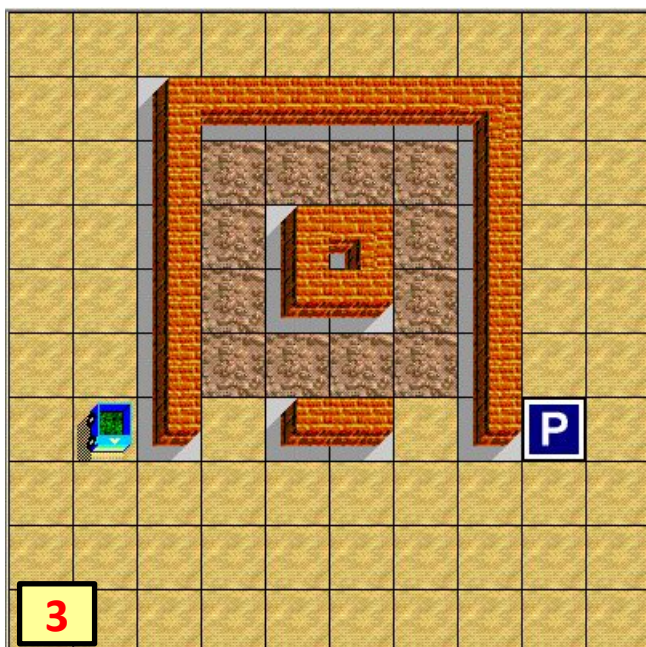
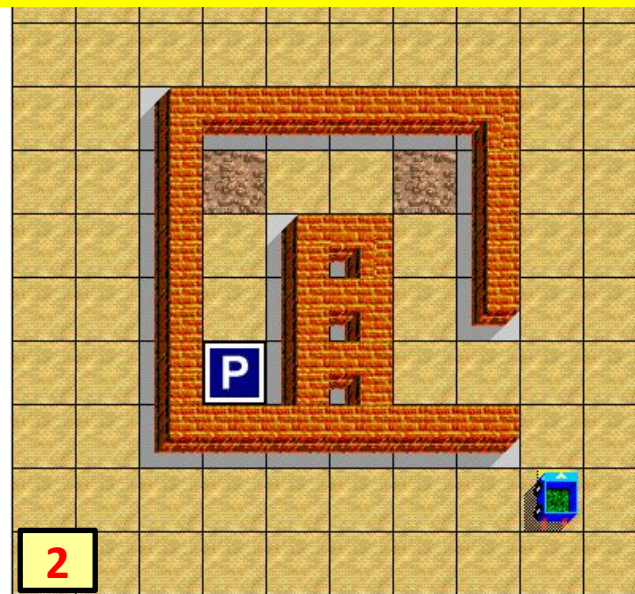
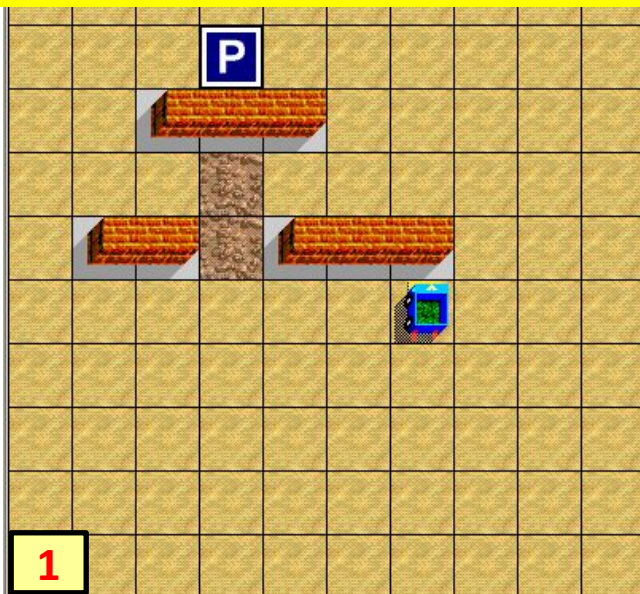


**Создать новый
лабиринт**

**Вс
эле
лабиринта**

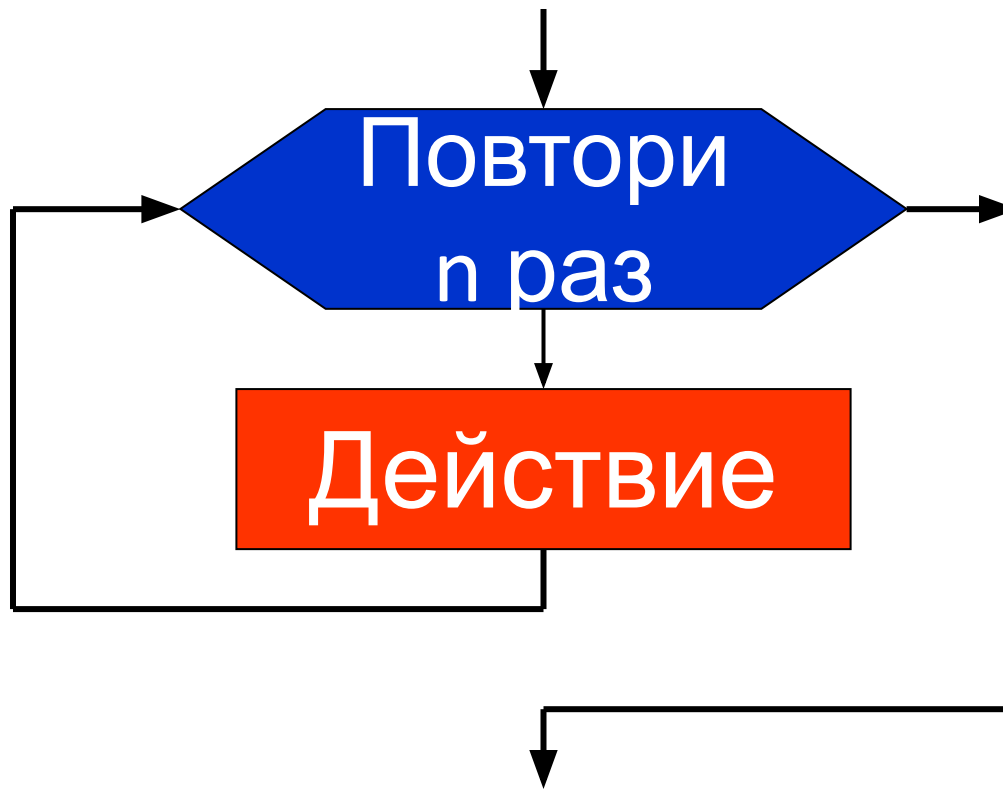
**Выйти из режима
редактирования с
сохранением
лабиринта**

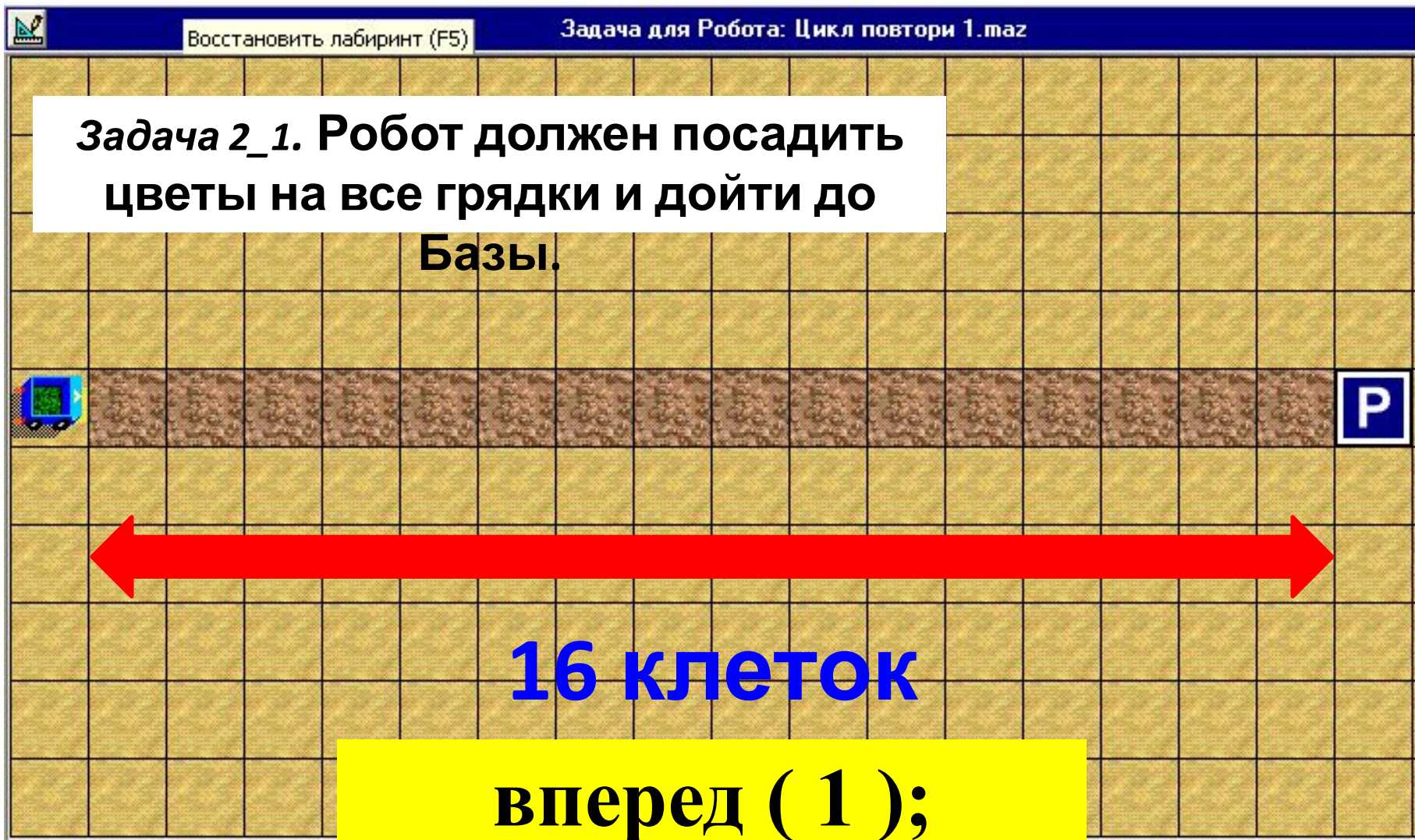
Задачи для самостоятельного решения



Занятие 2. Цикл ПОВТОРИ n РАЗ

```
repeat (количество повторений)  
{  
    повторяемые действия;  
}
```





Программа

{

повтори (16)

{

вперед (1);

посади;

}

вперед (1);

}

**Количество
повторений**

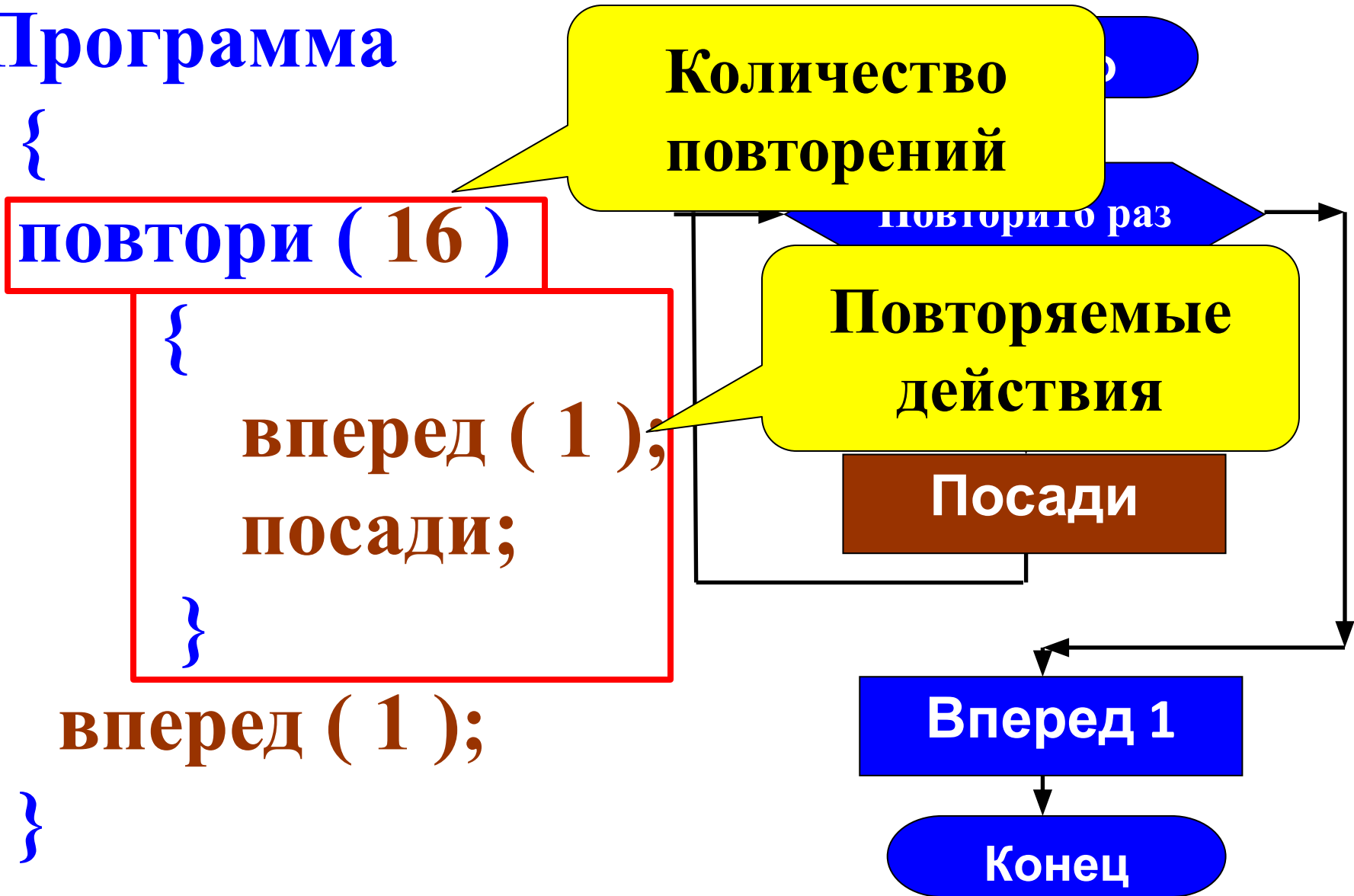
повторить раз

**Повторяемые
действия**

Посади

Вперед 1

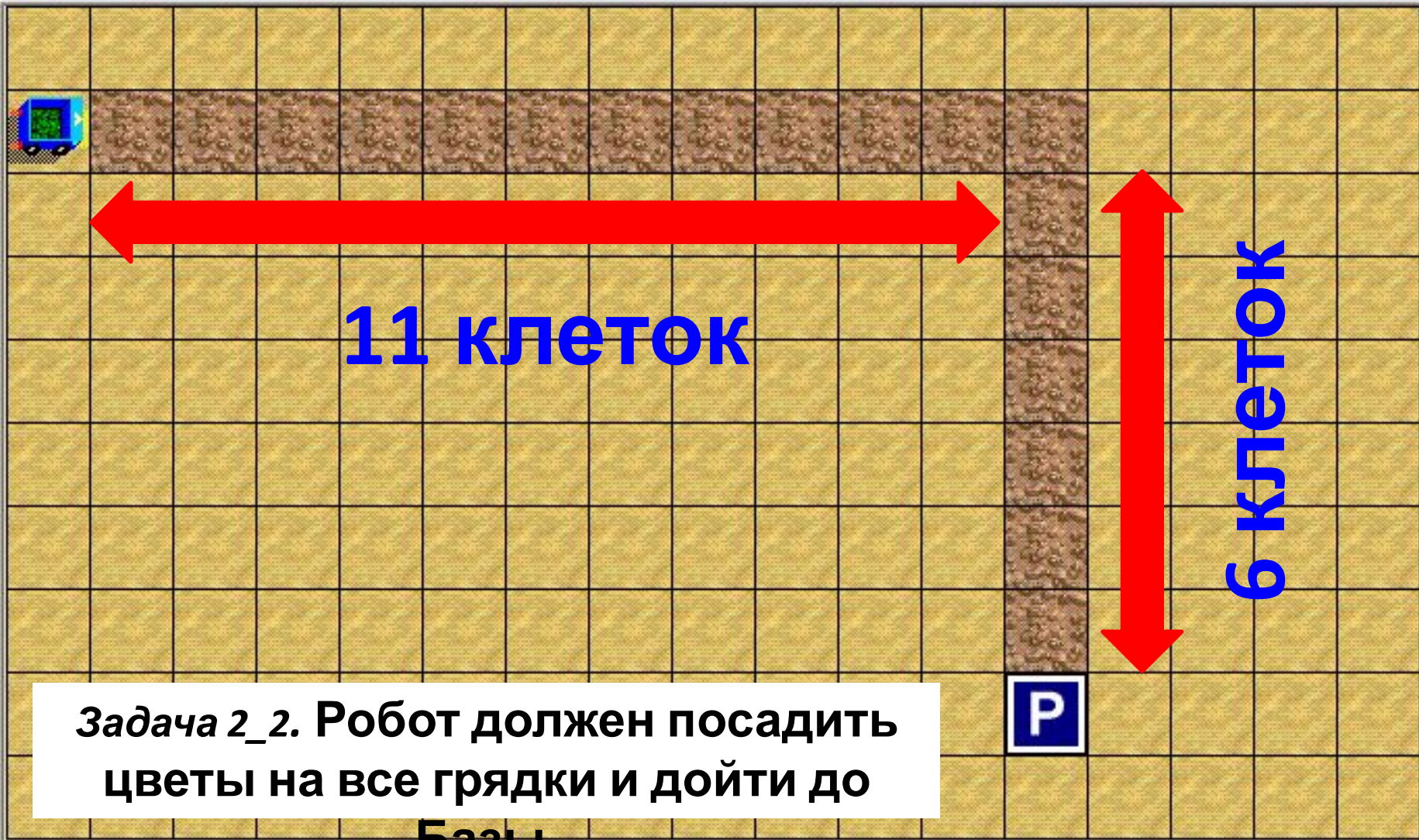
Конец





Восстановить лабиринт (F5)

Задача для Робота: Цикл повтори 2.маз



Задача 2_2. Робот должен посадить цветы на все грядки и дойти до Базы.

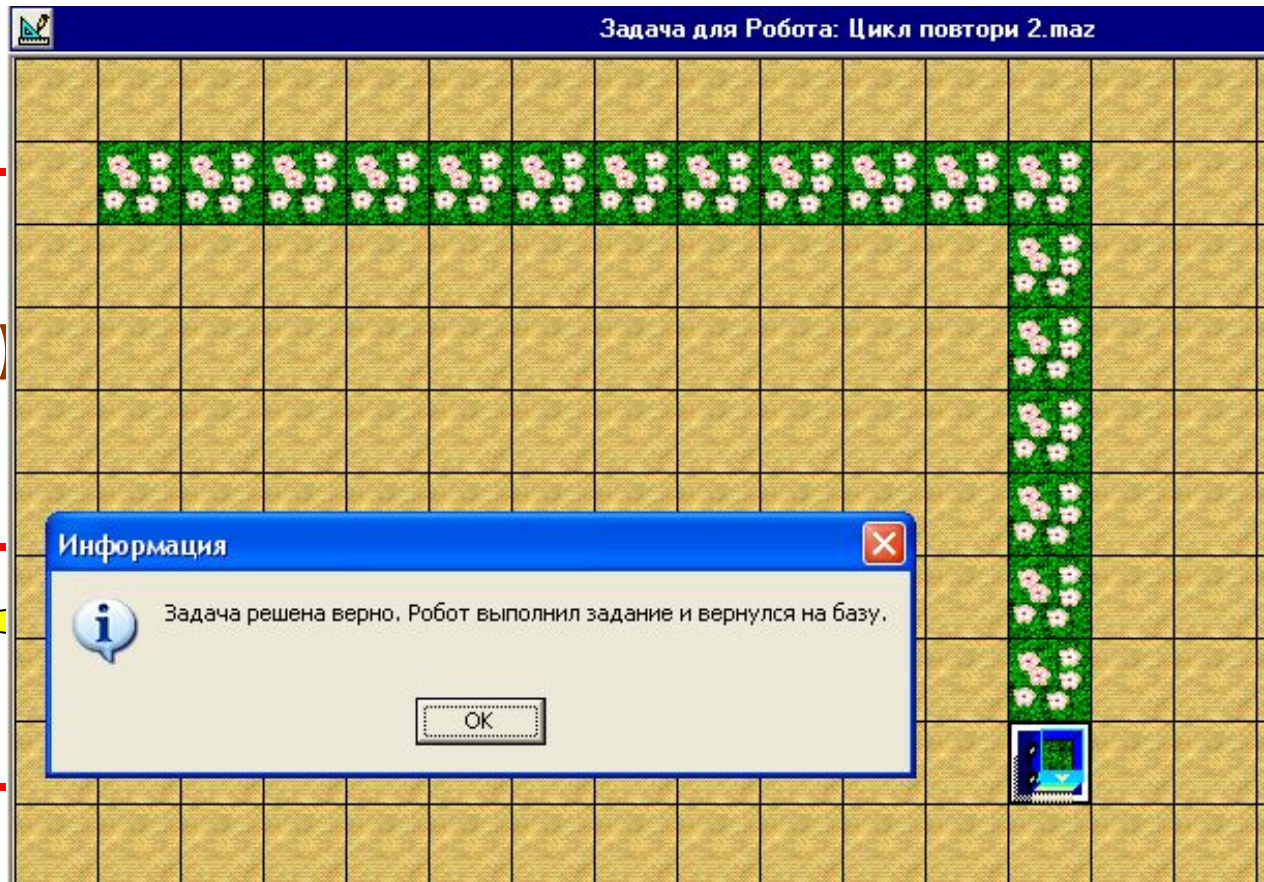
Программа

```
{
  повтори ( 11 )
  {
    вперед ( 1 )
    посади;
  }
}
```

```
вперед ( 1 );
направо;
посади;
```

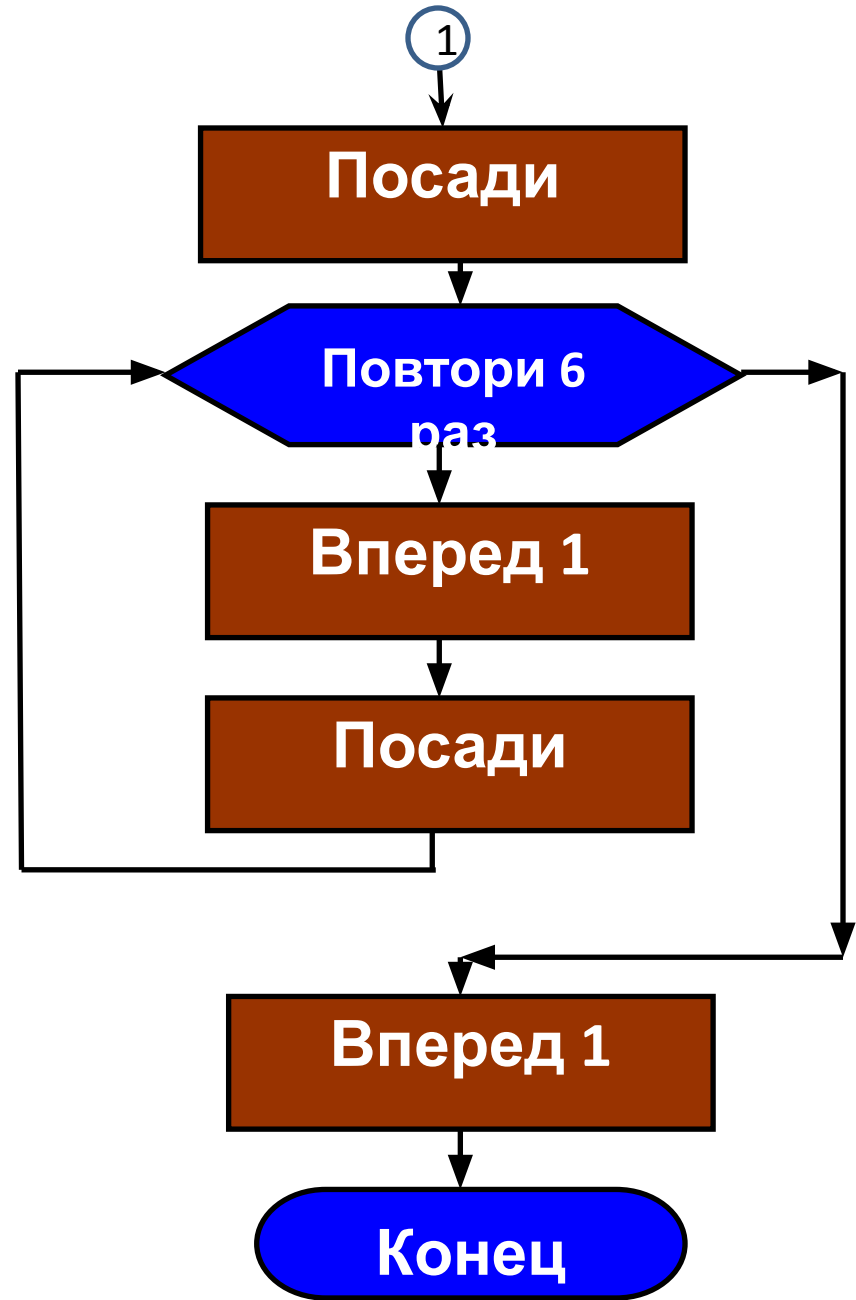
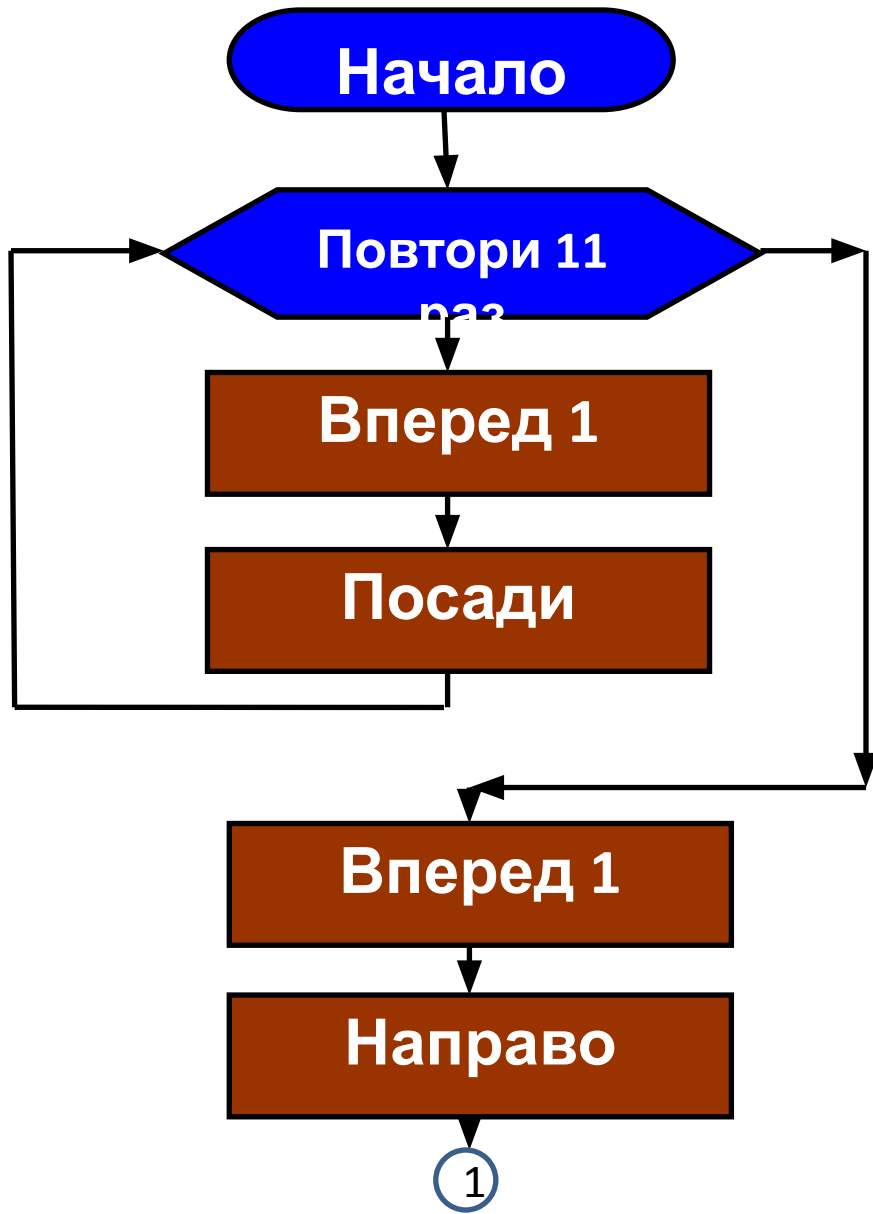
```
повтори ( 6 )
{
  вперед ( 1 );
  посади;
}
```

```
вперед ( 1 );
}
```

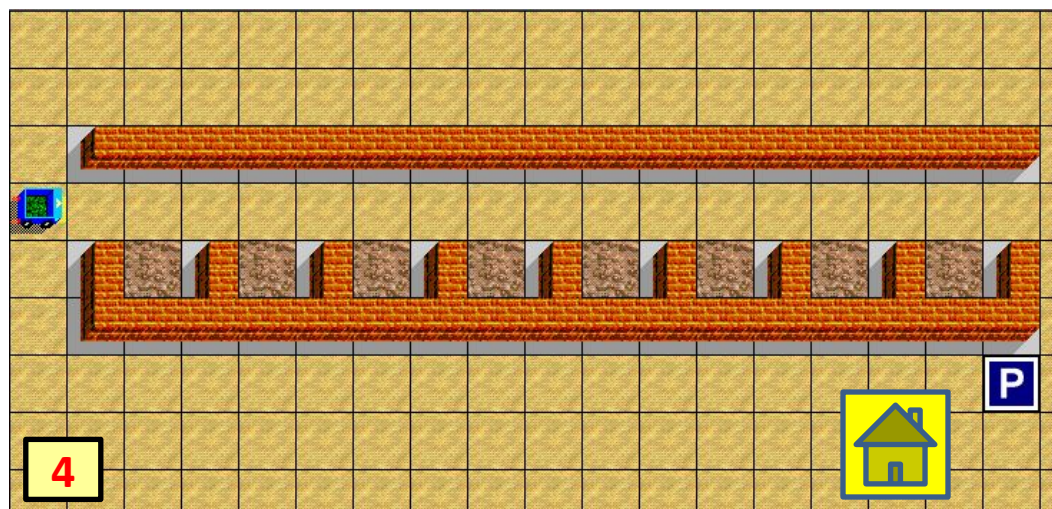
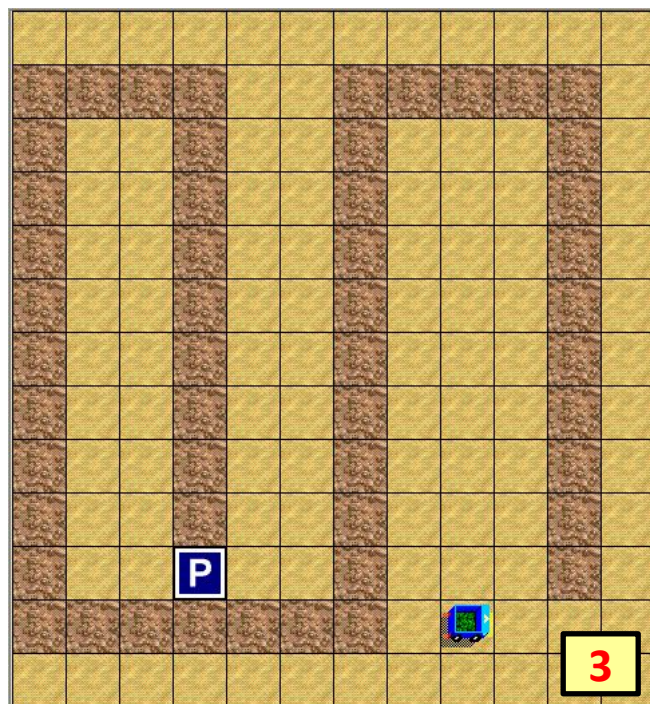
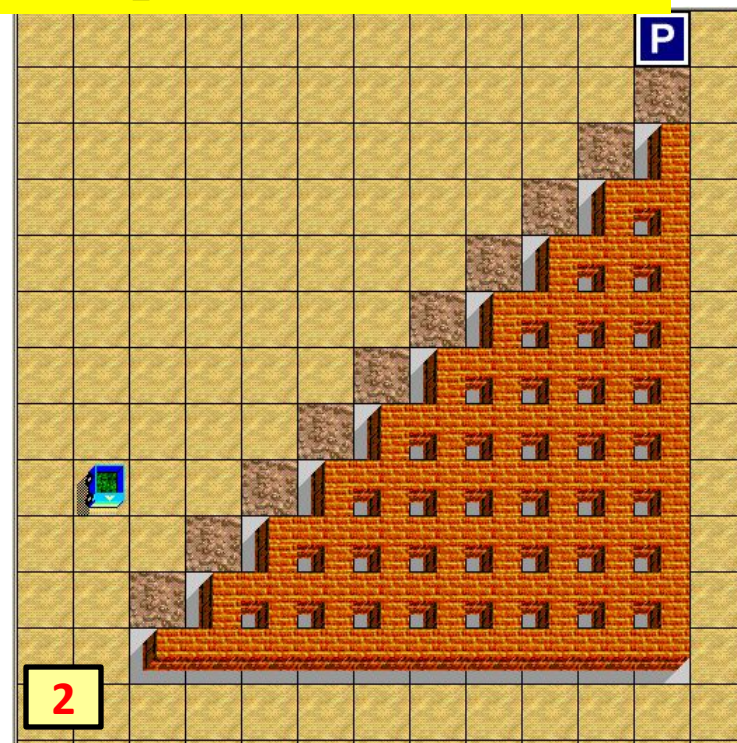
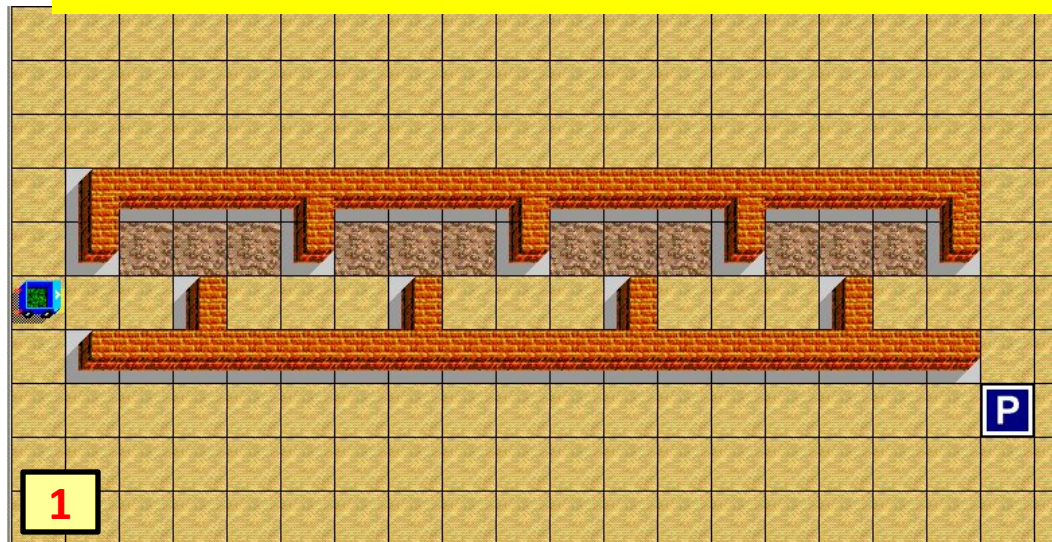


Движемся по вертикали

Заходим на базу

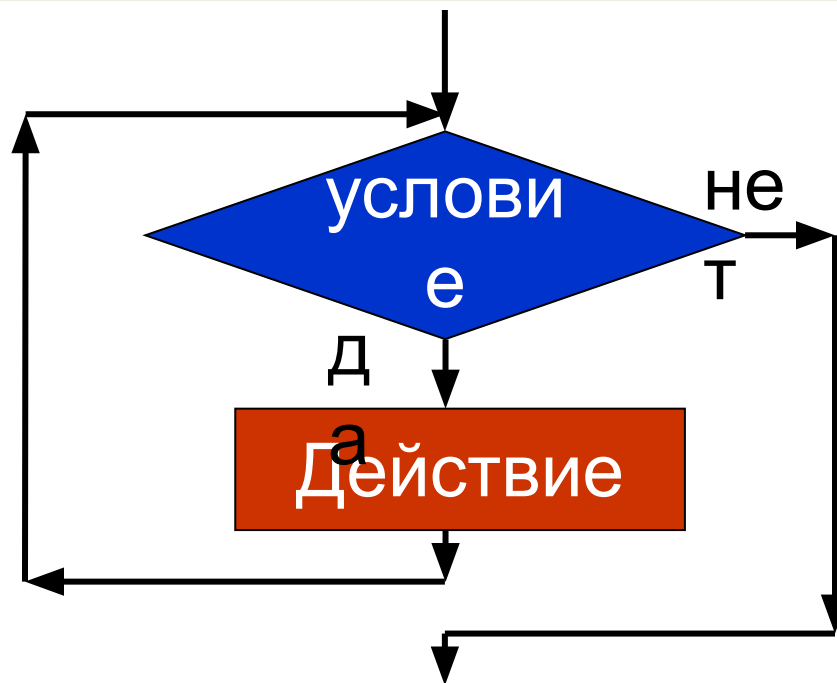


Задачи для самостоятельного решения



Занятие 3. Цикл ПОКА

```
while ( условие )  
{  
    повторяемые действия ;  
}
```



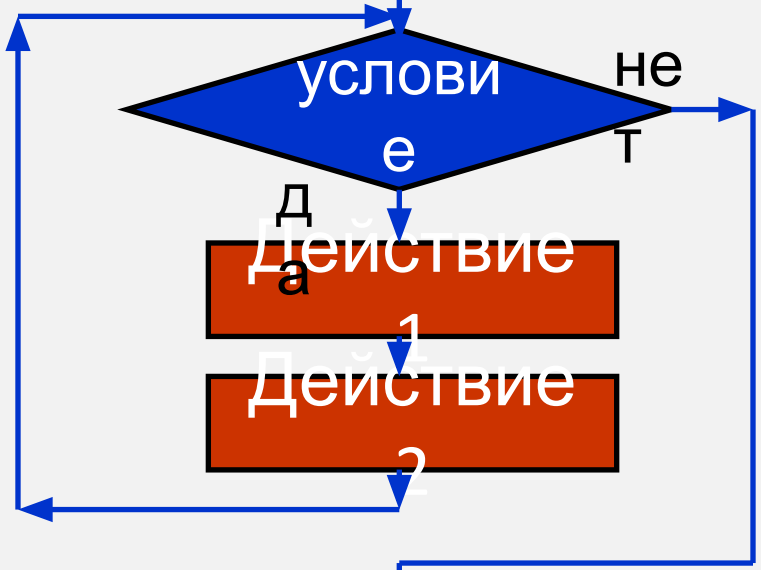
слева_стена справа_стена впереди_стена сзади_стена	условия определяют, есть ли стена в ближайшей клетке в указанном направлении
слева_клумба справа_клумба впереди_клумба сзади_клумба	условия определяют, есть ли клумба в соседней клетке в указанном направлении
слева_свободно справа_свободно впереди_свободно сзади_свободно	условия определяют, свободна ли ближайшая клетка в указанном направлении
грядка клумба база	условия определяют, является ли клетка, в которой стоит Робот, грядкой клумбой или Базой

Составные условия образуются из простых условий добавлением логических операций

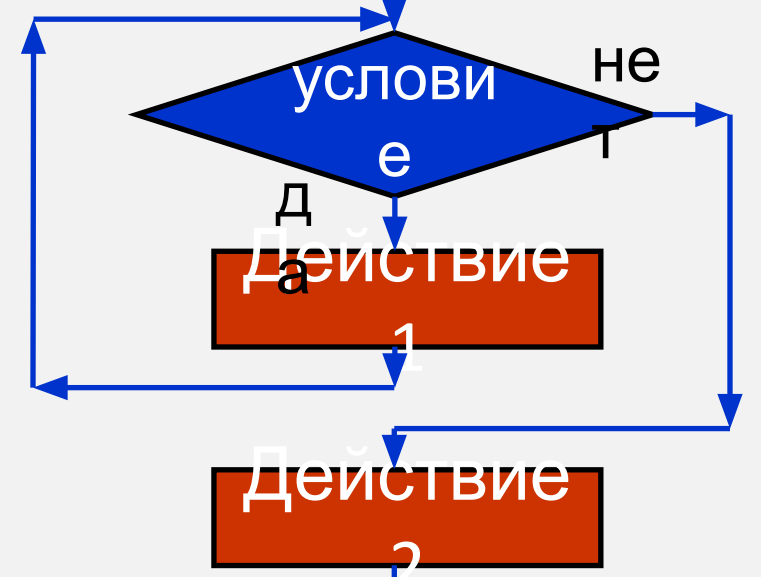
И, ИЛИ, НЕ.

Если **A**, **B** – простые условия, то:

1. Составное условие (**A и B**) будет выполняться только тогда, когда выполняются каждое из простых условий.
2. Составное условие (**A или B**) будет выполняться тогда, когда выполняется хотя бы одно из простых условий.
3. Составное условие (**не A**) будет выполняться, когда не выполняется простое условие **A**.



```
пока ( условие )  
{  
    Действие 1;  
    Действие 2;  
}
```



```
пока ( условие )  
{  
    Действие 1;  
}  
Действие 2;
```



Восстановить лабиринт (F5)

Задача для

Программа

```
{  
пока ( впереди_свободно )  
    {  
        вперед ( 1 );  
        посади;  
    }  
направо;  
вперед ( 1 );  
}
```

**Цикл
выполняется
пока робот не
упрется в стену**



**ы на все
лабиринта**

Программа

```

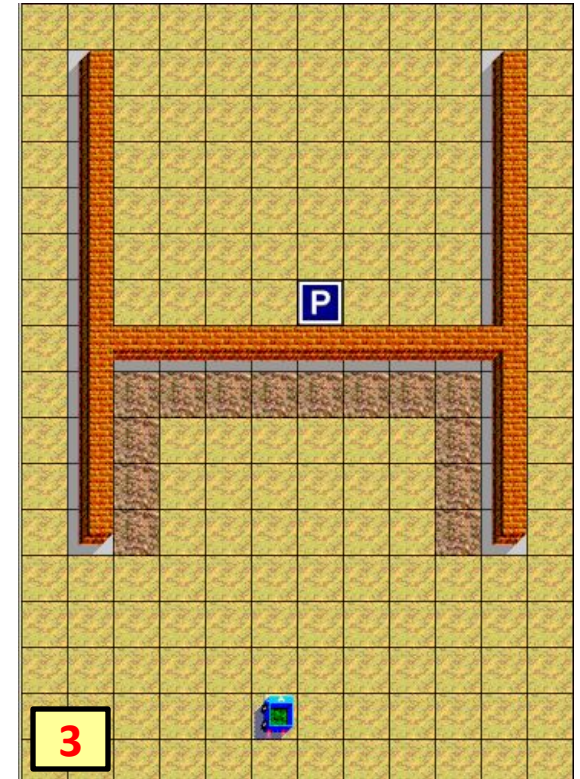
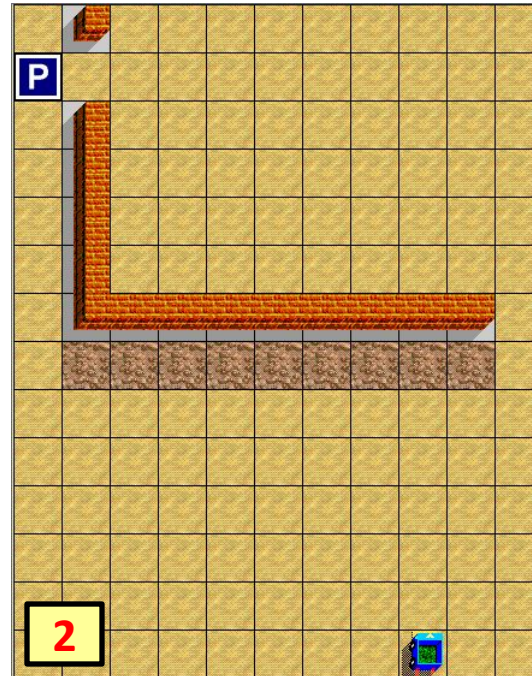
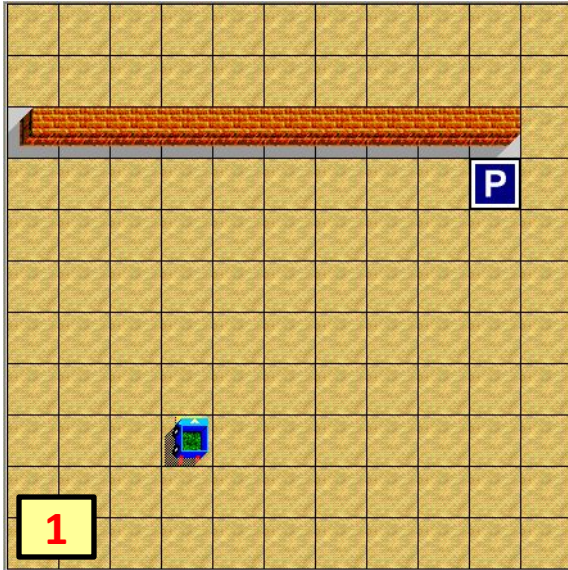
{
  вперед ( 1 );
  пока ( грядка )
  {
    посади;
    вперед ( 1 );
  }
  направо;
  вперед ( 1 );
  направо;
  вперед ( 1 );
}

```

Цикл
выполняется пока
под роботом не
закончатся грядки



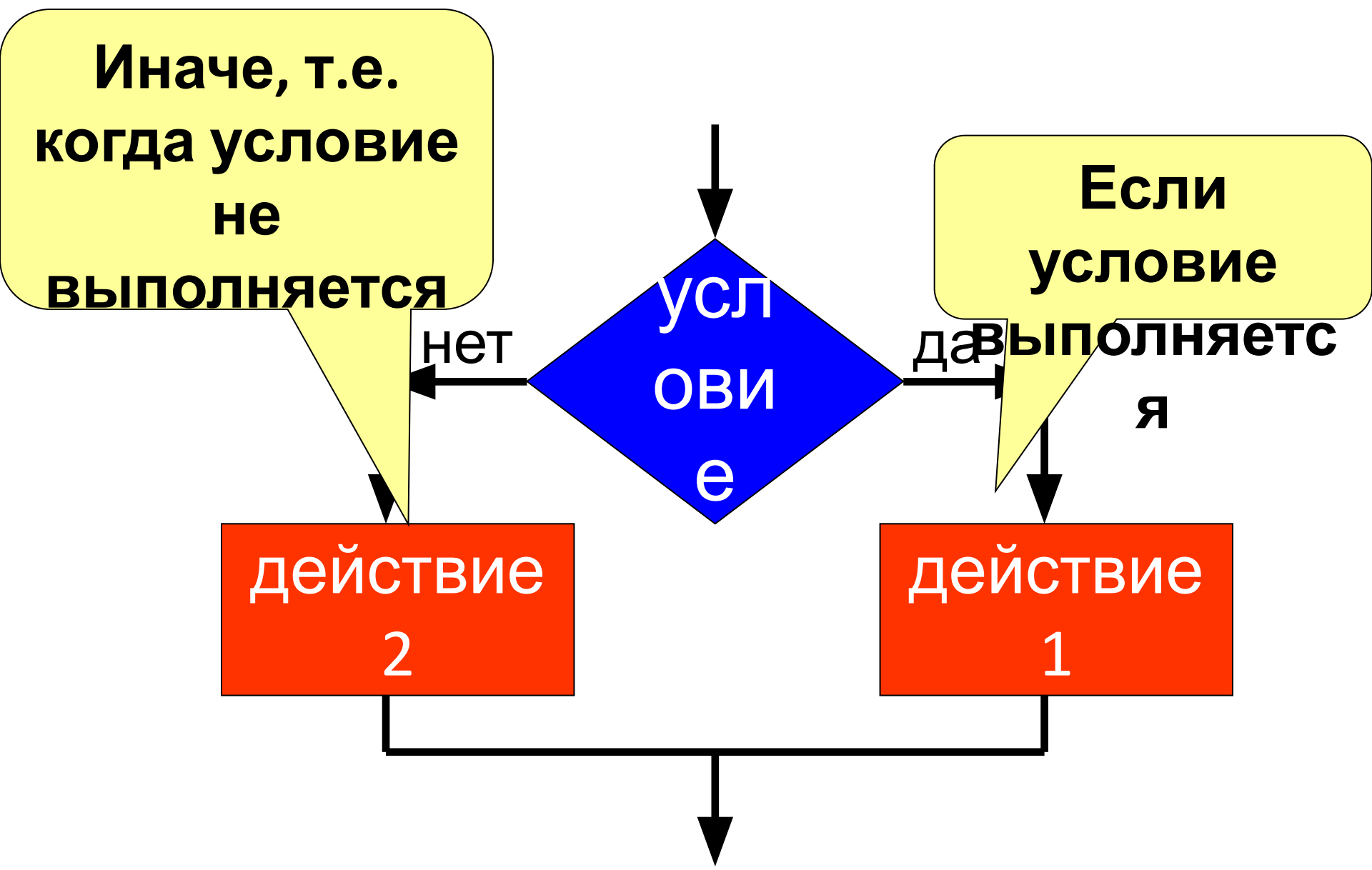
Задачи для самостоятельного решения



Занятие 4. Ветвления

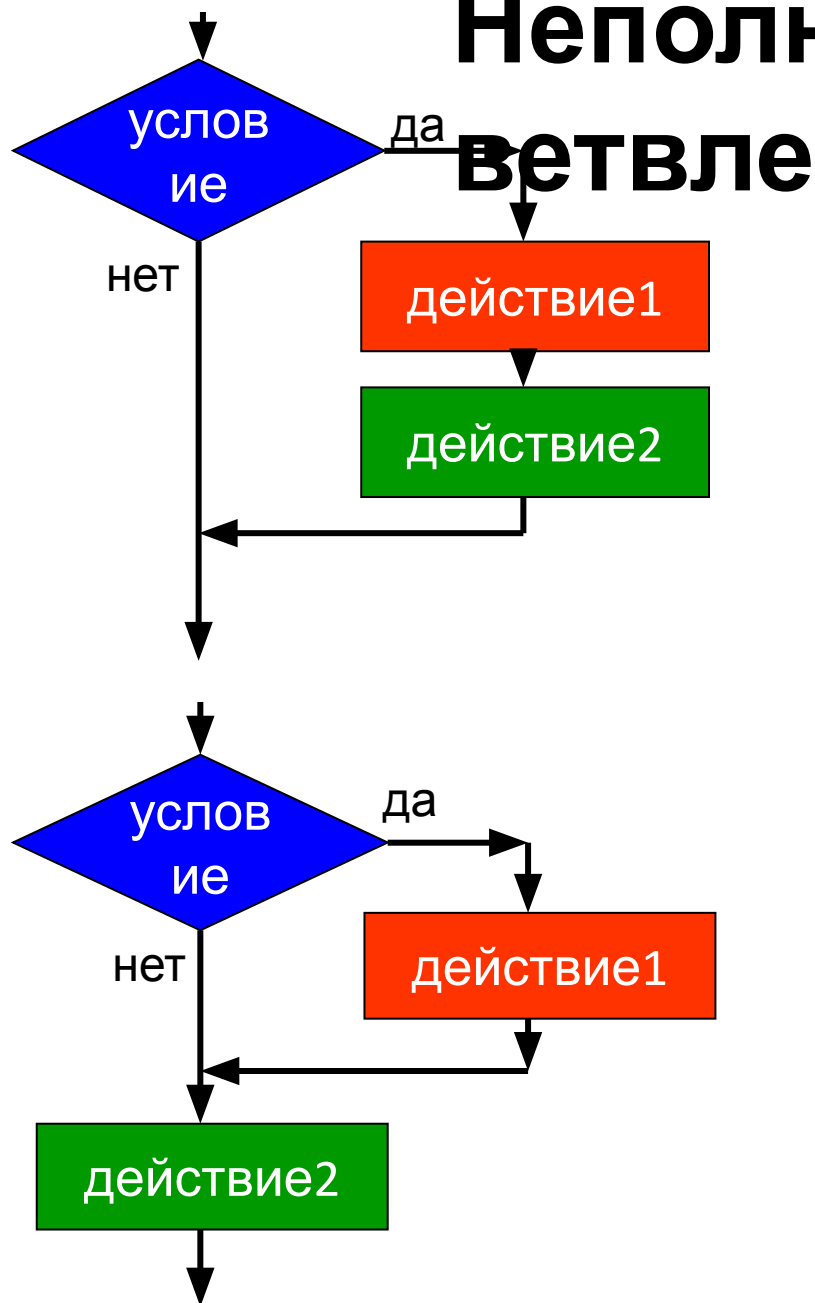
Полное

```
if ( условие )
{
    действие 1;
}
else
{
    действие 2;
}
```



Неполное

ветвление



```

если ( условие )
{
    действие 1;
    действие 2;
}
  
```

```

если ( условие )
{
    действие 1;
}
действие 2;
  
```

Программа

{

{



й

Информация [X]

i Задача решена верно. Робот выполнил задание и вернулся на базу.

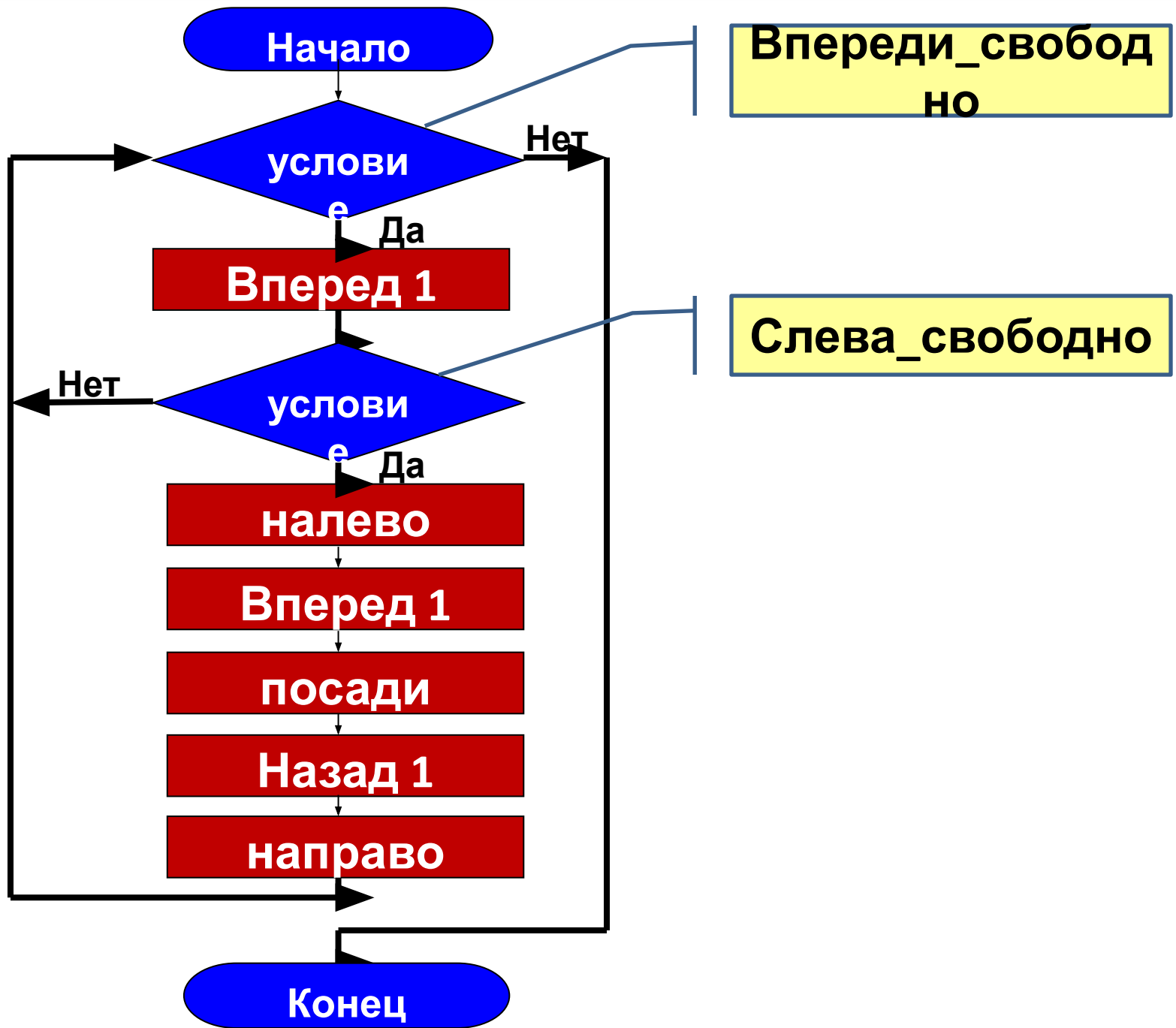
OK

}

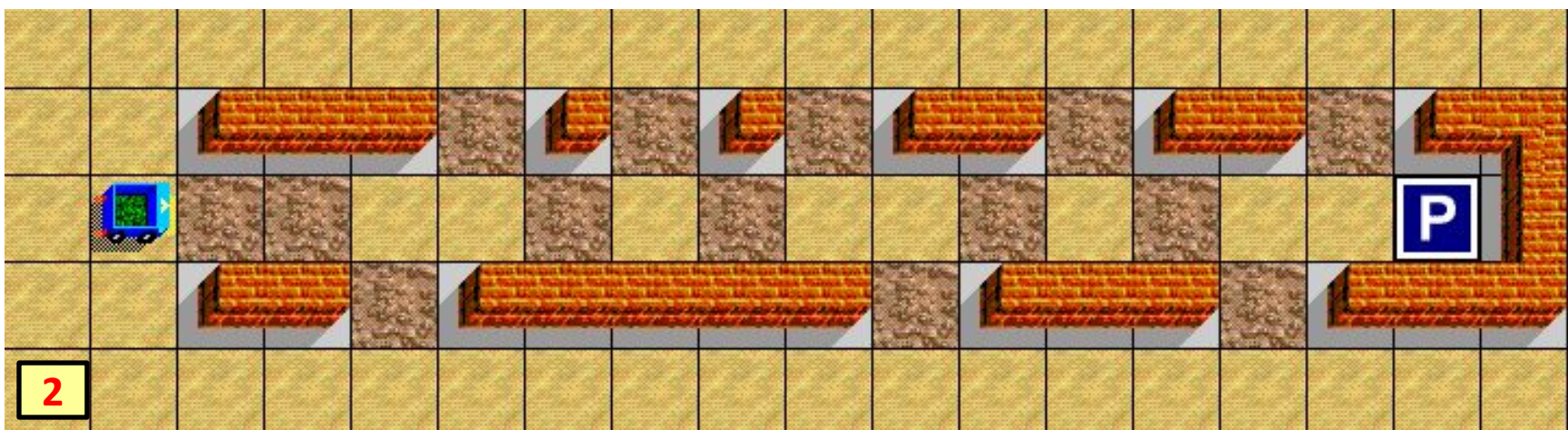
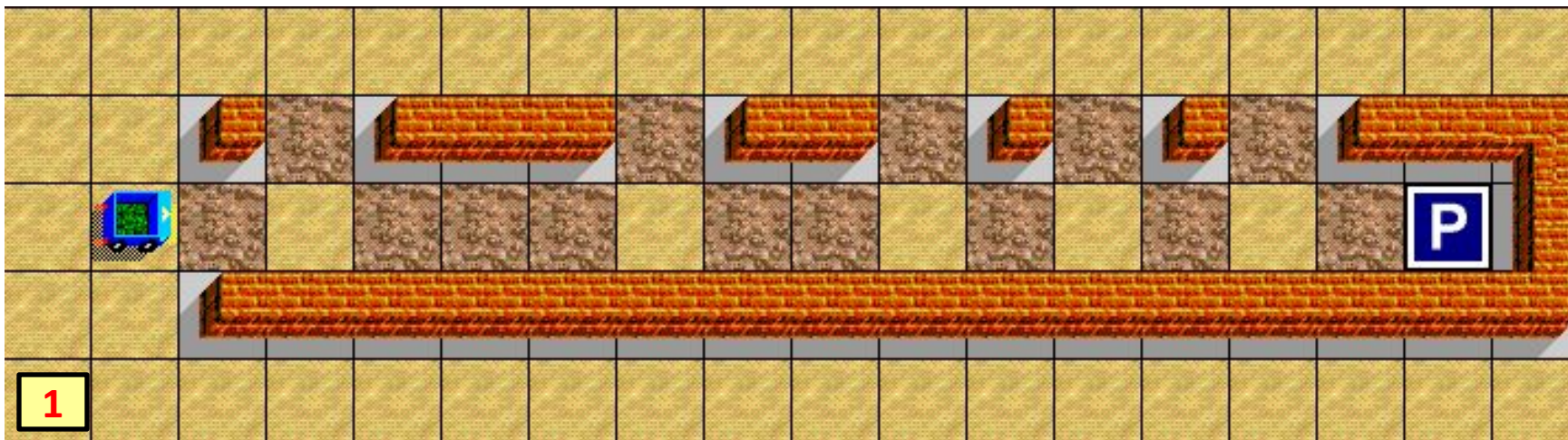
}

}

**засаживаем там
грядку**

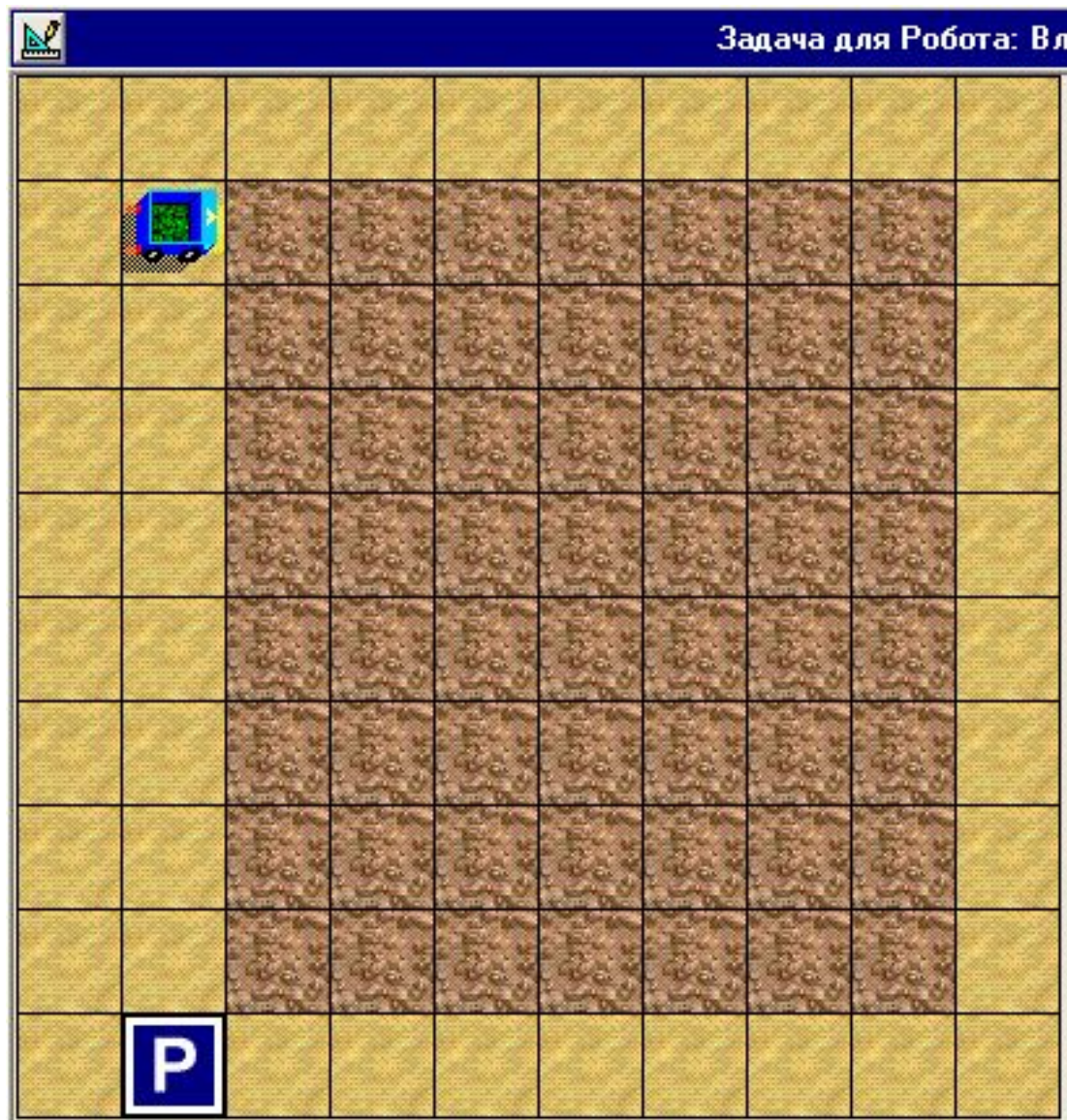


Задачи для самостоятельного решения



Занятие 5. Вложенные циклы

Задача 6. Робот должен посадить цветы на все грядки и прийти до Базы. (Лабиринт имеет размеры 8x7).

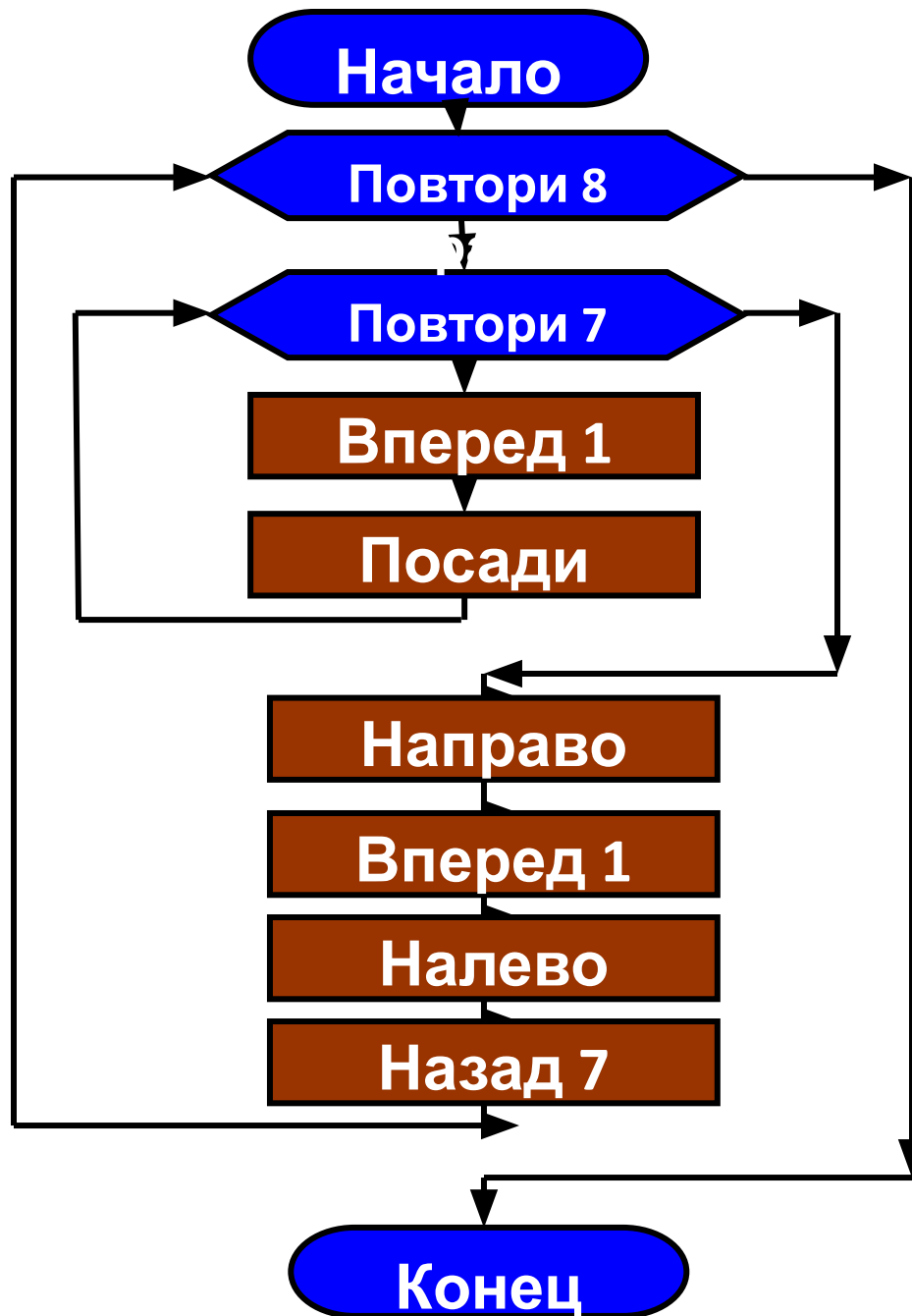


Программа

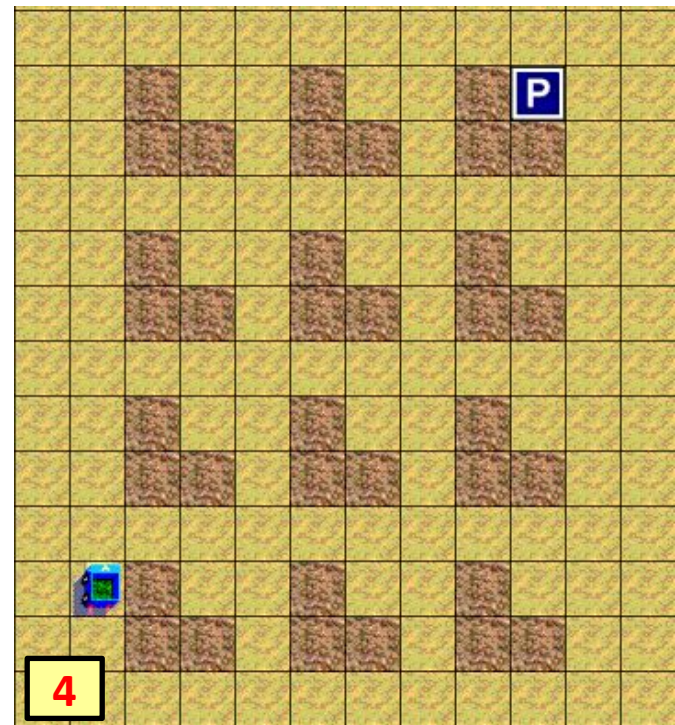
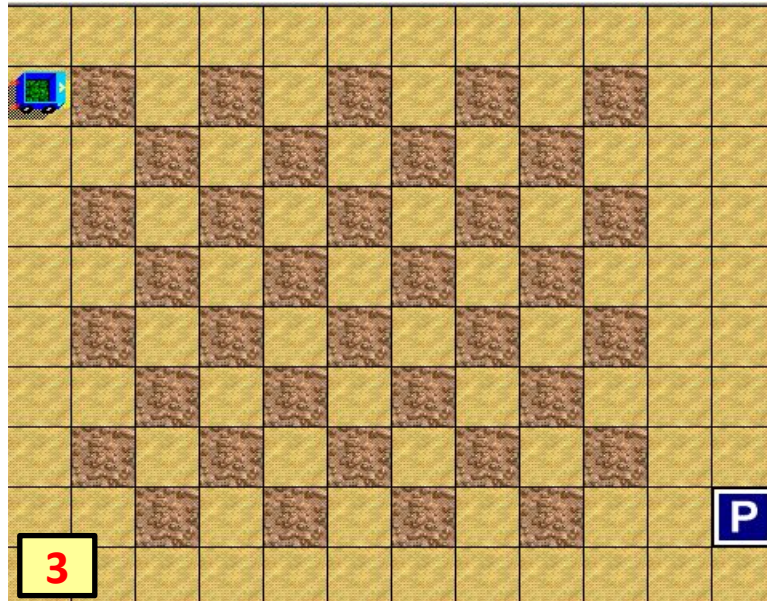
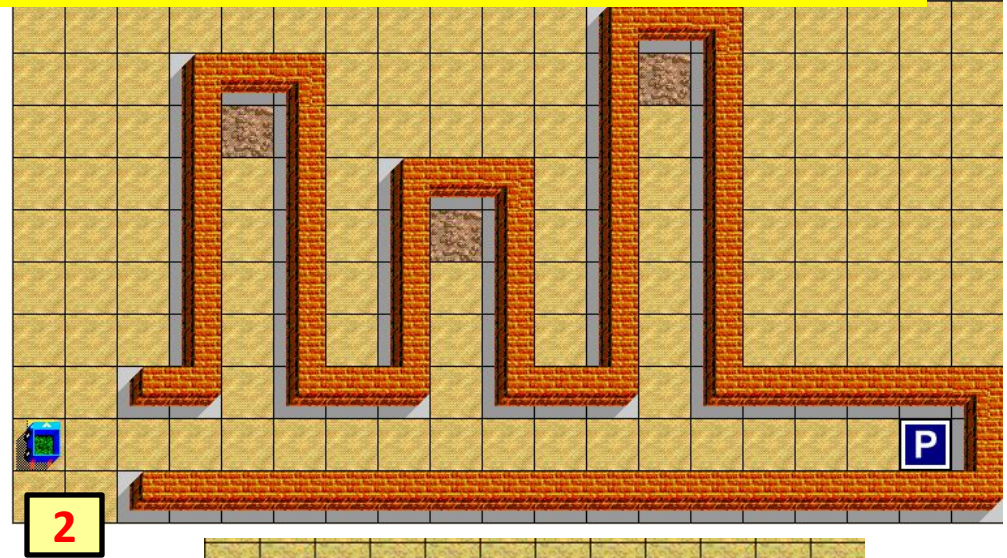
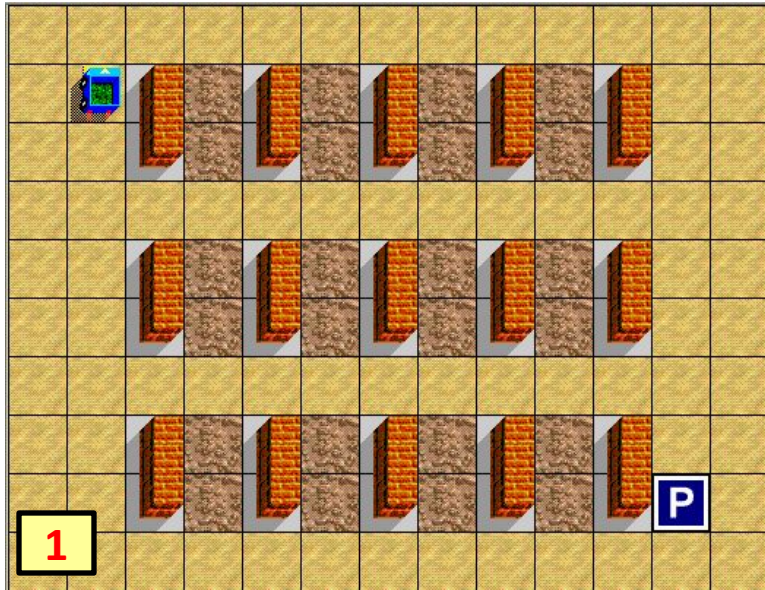
```
{  
  повтори (8)  
  {  
    повтори (7)  
    {  
      вперед (1);  
      посади;  
    }  
    направо;  
    вперед (1);  
    налево;  
    назад (7);  
  }  
}
```

Повторяем эти действия для 8 строк

Возвращаем робота в начало следующей строки



Задачи для самостоятельного решения



Занятие 6. Процедуры

```

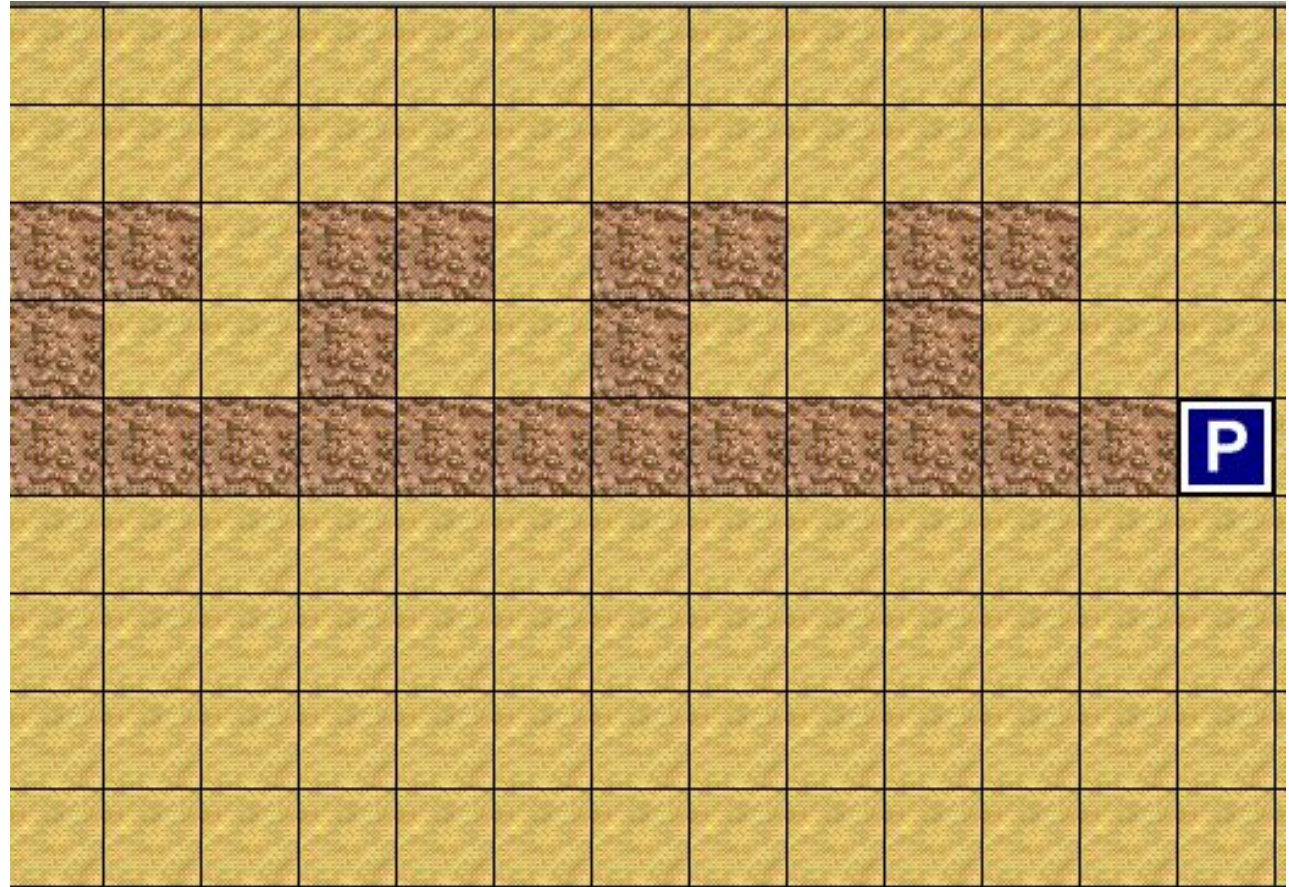
Программа
{
  вперед ( 1 );
  повтори ( 5 )
  {
    Процедура;
  }
}

```

```

Процедура
{
  вперед ( 1 );
  налево;
  вперед ( 2 );
  налево;
  посади;
  вперед ( 1 );
  посади;
  налево;
  вперед ( 1 );
  посади;
  вперед ( 1 );
  посади;
  налево;
  вперед ( 1 );
  посади;
  вперед ( 1 );
  посади;
  вперед ( 1 );
}

```



Задача 7. Робот должен посадить цветы на все грядки и дойти до Базы.

Программа

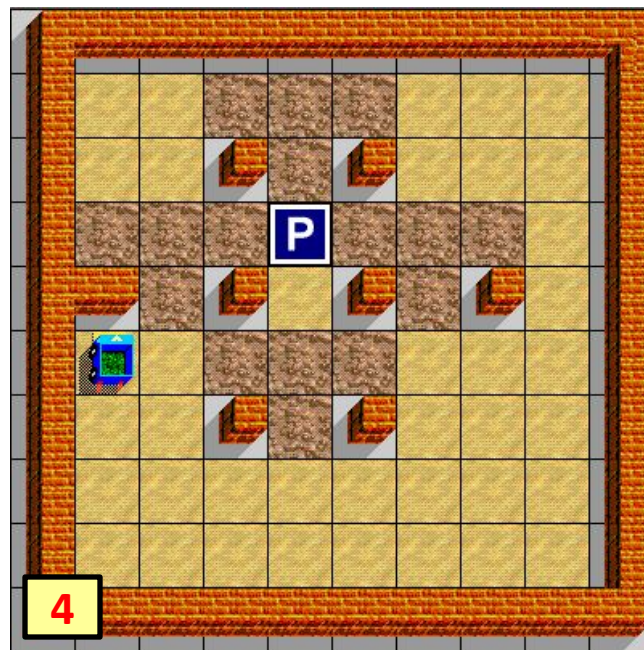
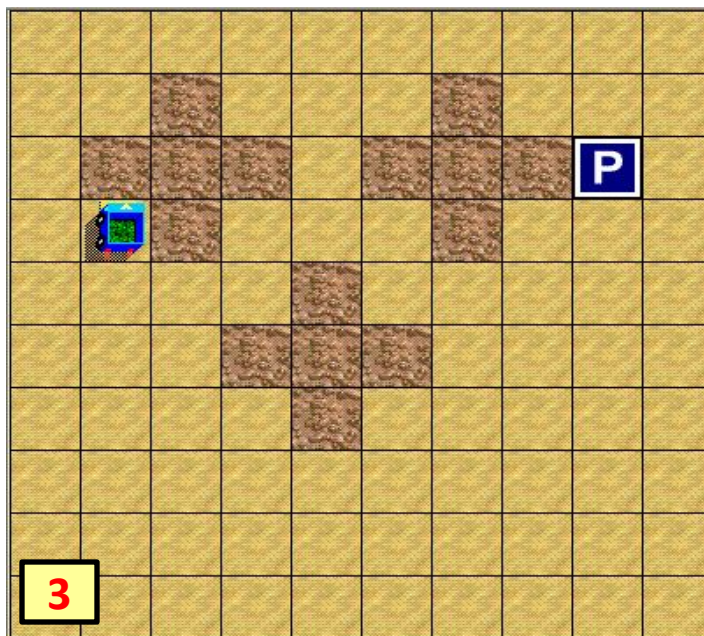
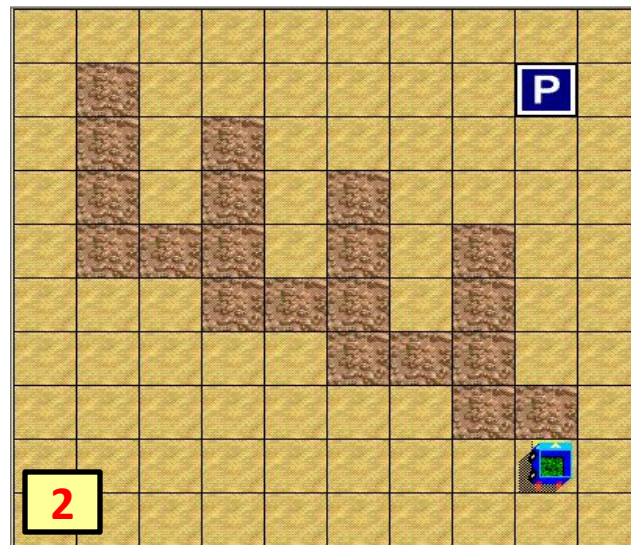
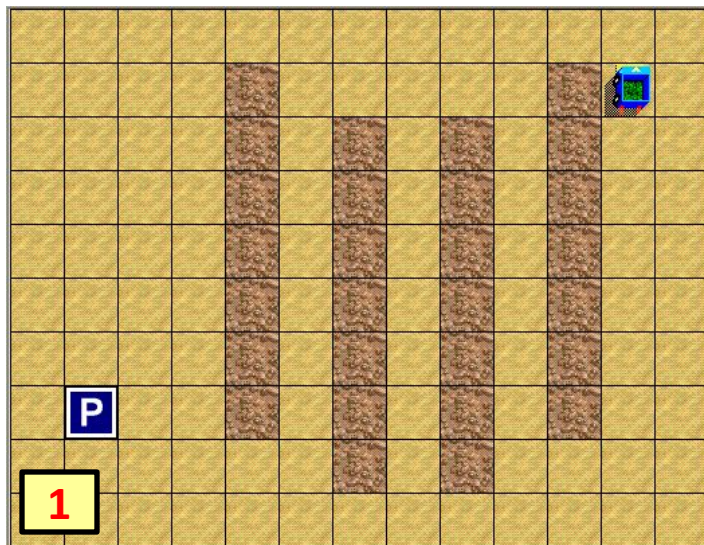
```
{  
вперед ( 1 );  
повтори ( 5 )  
  {  
    Процедура;  
  }  
}
```

Процедура

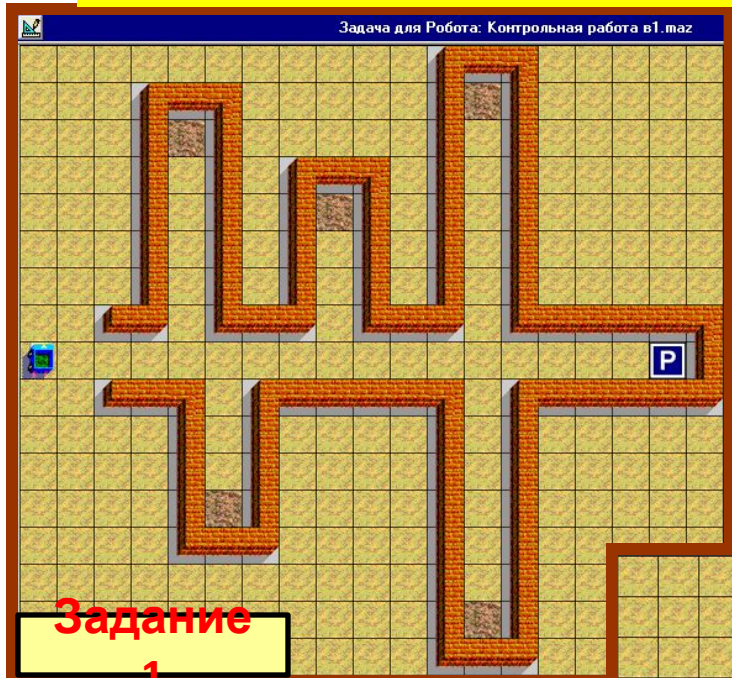
```
{  
вперед ( 1 ); налево;  
вперед ( 2 ); налево;  
посади; вперед ( 1 );  
посади; налево;  
вперед ( 1 ); посади;  
вперед ( 1 ); посади;  
налево; вперед ( 1 );  
посади; вперед ( 1 );  
посади; вперед ( 1 );  
}
```



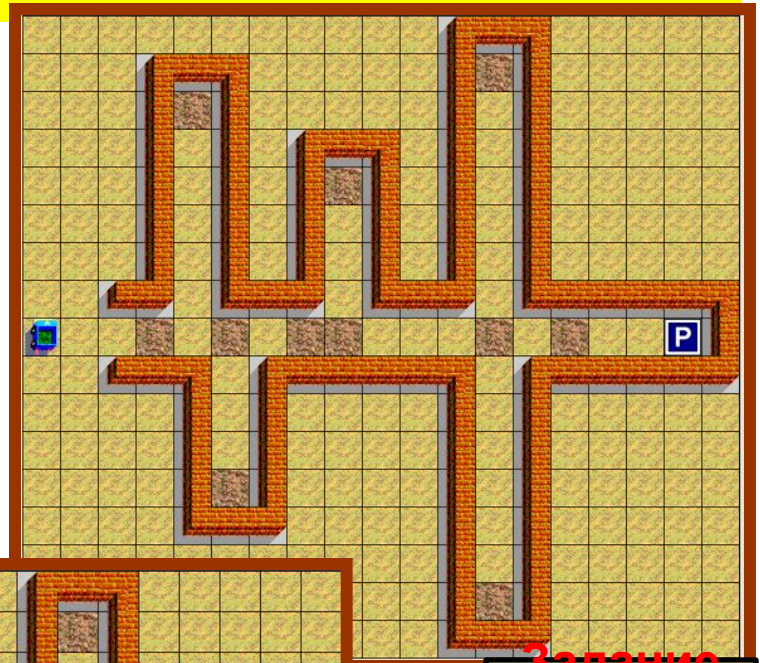
Задачи для самостоятельного решения



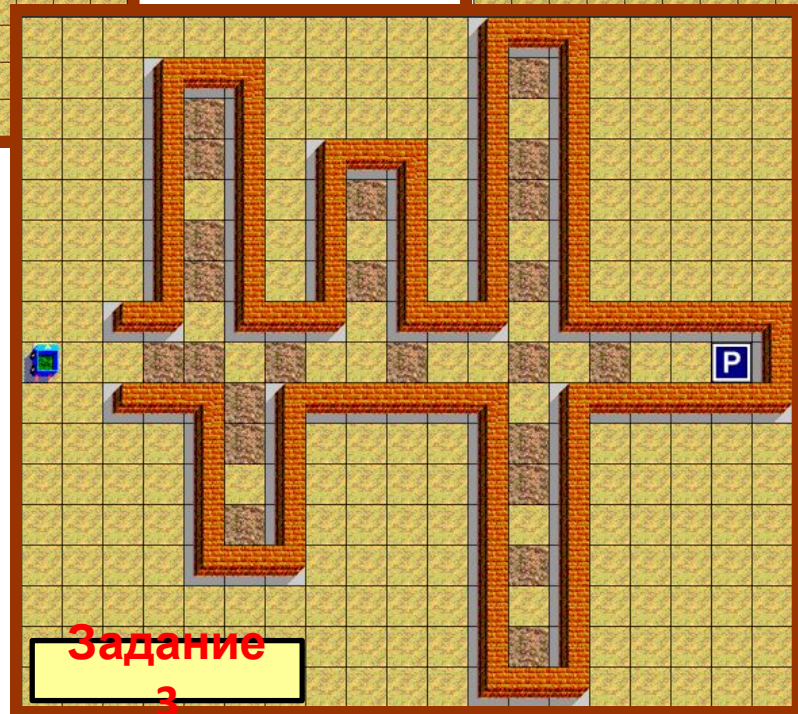
Обстановки для КОНТРОЛЬНОЙ РАБОТЫ



Задание 1



Задание 2



Задание 3

Список литературы и интернет источников



Сайт автора системы программирования «Исполнители» Полякова Константина Юрьевича <http://kpolyakov.spb.ru/>



Разработка уроков и презентация по теме "Исполнитель Робот" в пропедевтическом курсе информатики (7 класс) Автор: Чайченко Сергей Викторович <http://pedsovet.su/load/15-1-0-33553>



Сайт поддержки уроков информатики Д.М. Ушакова. Готовые лабиринты по изучаемым темам. <http://inform239.narod.ru/robot.html>

Список литературы и интернет источников

1. Босова Л. Л. Информатика и ИКТ: учебник для 7 класса. – М.: БИНОМ. Лаборатория знаний, 2010.
2. Босова Л. Л. Информатика и ИКТ: рабочая тетрадь для 7 класса. – М.: БИНОМ. Лаборатория знаний, 2011
3. Босова Л. Л., Босова А. Ю. Уроки информатики в 5–7 классах: методическое пособие. – М.: БИНОМ. Лаборатория знаний, 2007.
4. Поурочные разработки для 7 класса (4 четверть) – авторская мастерская Босовой Л.Л.
<http://metodist.lbz.ru/authors/informatika/3/>
5. Задачи для робота (Кумир) Автор Удалова Т.Л.
<http://www.licey.net/kumir/robot>
(Легко переделать для «Исполнителей»)