

План-конспект урока по информатике

Дата: 08.12.2023

Класс: 7

Учитель: Кунашко Анна Андреевна

Тема: Исполнитель Робот

Цели урока:

Образовательные: формирование у учащихся представления о исполнителе Робот, командах исполнителя, об использовании алгоритмической конструкции следования для исполнителя Робот.

Развивающие: содействовать развитию алгоритмического мышления; создание ситуации для формирования коммуникативных умений, рефлексивных способностей, самостоятельности учащихся.

Воспитательные: воспитание трудолюбия, ответственности за результаты своего труда; воспитание культуры делового общения при совместной работе в группе, нацеленность на результативность обучения.

Тип урока: урок усвоения и закрепление новых знаний.

Оборудование: класс компьютерной техники, мультимедийный проект.

Литература: Котов, В. М. Информатика: учеб. пособие для 7 кл. учреждений общ. сред. образования с белорус. и рус. обучения / В. М. Котов, А. И. Лапо, Е. Н. Войтехович. – Минск: Нар. асвета, 2017.

Структура урока:

1. Организационный момент (1 мин).
2. Этап актуализации знаний (4 мин).
3. Целемотивационный этап (2 мин).
4. Этап изучения новой темы (15 мин).
5. Физкультминутка (2 мин).
6. Проверка понимания и закрепления изученного материала (15 мин).
7. Информация о домашнем задании (1 мин).
8. Подведение итогов урока (2 мин).
9. Рефлексия (3 мин).

Ход урока:

1. Организационный момент.

Здравствуйте, дети!

Сегодня на уроке мы начнем изучать новую тему. Запишите сегодняшнюю тему в тетрадь: «Исполнитель Робот».

2. Этап актуализации знаний.

Начнем наш урок с проверки понимания пройденной темы.

1. Что такое алгоритм? *Понятная и конечная последовательность точных действий (команд), формальное выполнение которых позволяет получить решение поставленной задачи.*
2. Какие способы записи алгоритмов вам известны? *Словесный, графический, программный.*
3. Что называют алгоритмической конструкцией следования? *Последовательность команд алгоритма, которые выполняются в том порядке, в котором они записаны.*

4. Какой алгоритм называется вспомогательным? *Алгоритм, целиком используемый в составе другого алгоритма.*
5. Для чего нужны вспомогательные алгоритмы? *Вспомогательный алгоритм решает некоторую подзадачу основной задачи. Вызов вспомогательного алгоритма в программе заменяет несколько команд одной.*

3. Целемотивационный этап.

Сегодня мы изучаем тему «Исполнитель Робот». Мы должны за сегодня выполнить некоторые цели: знать: — что такое исполнитель Робот, команды исполнителя; уметь: — строить алгоритмические конструкции следования для исполнителя Робот.

4. Этап изучения новой темы.

9.1. Роботы в жизни человека

Человек с глубокой древности мечтал об искусственном создании, которое могло бы выполнять его приказы. Сегодня эта мечта стала реальностью — в жизни людей появились роботы. Они способны выполнять практически любую работу, доступную человеку, а также делать многие вещи, которые людям выполнить сложно или вообще невозможно. Роботы используются на производстве и в быту, могут работать в сфере услуг и развлечений. Есть роботы, похожие на человека, а есть совсем непохожие.

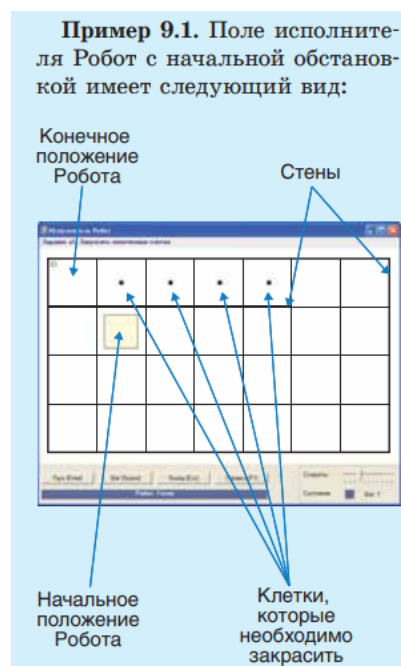
Робот — автоматическое устройство, которое действует по заранее составленной программе.

Робот получает информацию о внешнем мире от датчиков — аналогов органов чувств живых организмов — и предназначен для осуществления различных операций.

9.2. Среда обитания и система команд исполнителя Робот

Средой обитания исполнителя Робот является прямоугольное клетчатое поле. Размеры поля, как и для исполнителя Чертежник, задаются командой $Field(n, m)$. Первоначально Робот находится в центральной клетке поля.

Между некоторыми клетками, а также по периметру поля находятся стены. Робот может передвигаться по полю и закрашивать указанные клетки. Большой желтый квадрат внутри клетки означает начальное положение Робота, маленький — конечное (*пример 9.1*).



Поле Робота, на котором определено положение стен, начальное и конечное положение исполнителя, называют **обстановкой**.

Для подключения исполнителя Робот в программе прописывается команда **uses Robot**. Готовые задания с обстановками для Робота хранятся в задачнике, встроенном в систему программирования, и вызываются командой *task*. Эта же команда использовалась для Чертежника.

Система команд исполнителя:

Команда	Действие
Right	Перемещает Робота вправо
Left	Перемещает Робота влево
Up	Перемещает Робота вверх
Down	Перемещает Робота вниз
Paint	Закрашивает текущую ячейку

Записать структуру программы:

```
uses Robot;
begin
Field (10,10);
<команды>;
end.
```

Робот может становиться на обычную и на закрашенную клетку, но не может переместиться с клетки на клетку, если между ними стена. Робот не может переместиться за границы поля. Эти действия вызывают ошибку (пример 9.2). Робот может закрасить уже закрашенную клетку. Такое действие ошибку не вызывает.

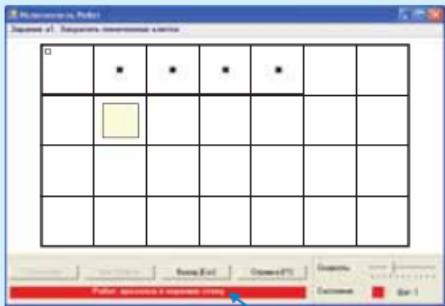
Пример 9.2. Вызов задачи a1 из встроенного задачника:

```
•Program1.pas
uses Robot;
begin
  Task('a1');
end.
```

Запишем в программе команду up.

```
uses Robot;
begin
  Task('a1');
  up;
end.
```

Сверху находится стена, поэтому перемещение Робота вверх невозможно:



Сообщение об ошибке

Запись команд исполнителя:

```
•Program1.pas*
uses Robot;
begin
  Task('a1');
  right;
  right;
end.
```

9.3. Использование алгоритмической конструкции следование для исполнителя Робот

Рассмотрим примеры решения задач для исполнителя Робот.

Пример 9.3. Робот находится на поле размером 3 на 3 клетки. Нужно закрасить все клетки, кроме средней (задача a2).

Для решения задачи Робот должен выполнить следующий алгоритм:

закрасить;

вправо;

закрасить;

вправо;

закрасить;

вниз;

закрасить;

вниз;

закрасить;

влево;

закрасить;

влево;

закрасить;

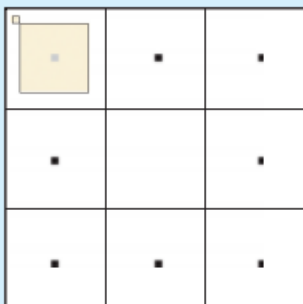
вверх;

закрасить;

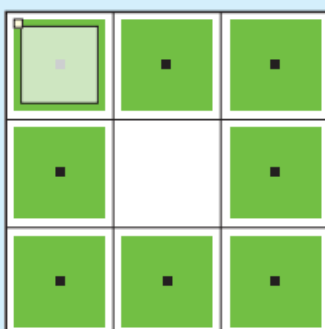
вверх.

В данном алгоритме Робот обходит клетки, двигаясь по часовой стрелке. Тот же результат можно получить, если Робот будет обходить поле против часовой стрелки, изначально двигаясь вниз.

Пример 9.3. Начальная обстановка:



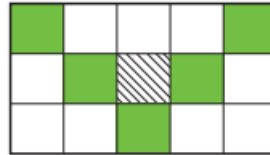
Результат работы программы имеет следующий вид:



Программа для исполнителя Робот:

```
uses Robot;  
begin  
  Task('a2');  
  paint; right;  
  paint; right;  
  paint; down;  
  paint; down;  
  paint; left;  
  paint; left;  
  paint; up;  
  paint; up;  
end.
```

Пример 9.4. Составим программу для закрашивания клеток поля Робота по образцу:



Такой обстановки нет в задачнике, поэтому вначале нужно создать поле Робота размером 5 на 3. Начальное положение Робота на таком поле отмечено заштрихованной клеткой.

Для решения задачи Робот должен выполнить алгоритм:

создать поле;

вниз;

закрасить;

влево;

вверх;

закрасить;

влево;

вверх;

закрасить;

вправо;

вправо;

вправо;

вправо;

закрасить;

влево;

вниз;

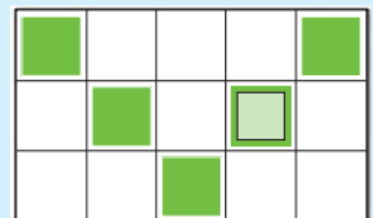
закрасить.

Пример 9.4. В данном случае программа для учебного компьютерного исполнителя Робот может быть составлена таким образом:

```
uses Robot;
begin
  Field(5, 3);
  down;
  paint;
  left;
  up;
  paint;
  left;
  up;
```

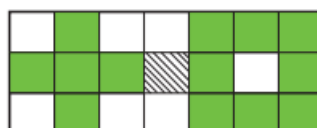
```
  paint;
  right;
  right;
  right;
  right;
  paint;
  left;
  down;
  paint;
end.
```

Результат работы записанной выше программы будет иметь следующий вид:



9.4. Вспомогательные алгоритмы

Пример 9.7. Напишем программу для закрашивания клеток поля Робота по образцу:



Такой обстановки нет в задачнике, поэтому создадим поле 7 на 3. Начальное положение Робота отмечено заштрихованной клеткой.

В данной задаче Робот должен нарисовать две отдельные фигуры: крест и квадрат. Составим два вспомогательных алгоритма.

Вспомогательный алгоритм **крест**:

влево; закрасить;
вниз; влево; закрасить;
вверх; влево; закрасить;
вправо; закрасить;
вверх; закрасить.

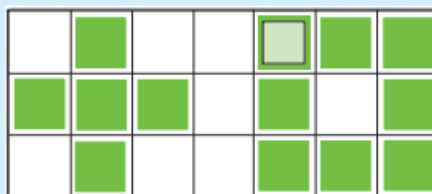
В качестве вспомогательного алгоритма для рисования квадрата можно использовать алгоритм решения задачи а2 (пример 9.3). Для перехода от одной фигуры к другой Робот должен сдвинуться на 3 клетки вправо:

крест;
вправо; вправо; вправо;
квадрат.

Пример 9.7. Программа для учебного компьютерного исполнителя Робот:

```
uses Robot;  
procedure krest;  
begin  
  left; paint;  
  down; left; paint;  
  up; left; paint;  
  right; paint;  
  up; paint;  
end;  
procedure kvadrat;  
begin  
  paint; right;  
  paint; right;  
  paint; down;  
  paint; down;  
  paint; left;  
  paint; left;  
  paint; up;  
  paint; up;  
end;  
begin  
  field(7,3);  
  krest;  
  right; right; right;  
  kvadrat;  
end.
```

Результат работы записанной выше программы будет иметь следующий вид:



5. Физкультминутка.

6. Проверка понимания и закрепления изученного материала.

Выполнить задания 3, 5. Задания 2 и 3 на выбор, если останется время можно выполнить остальное задание.



Задание 3. В системе программирования **Pascal ABC** выполните проверяемое задание а3 для исполнителя Робот (закрасить клетки, отмеченные точками) (рис. 3.9).

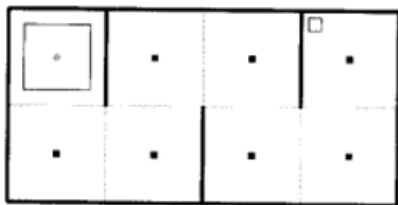


Рис. 3.9



Задание 2. Создайте поле размером 10×10 . Составьте программу создания изображения цветка из закрасенных клеток (рис. 3.14).



Задание 3. Создайте поле размером 17×17 . Составьте программу создания изображения оленя из закрасенных клеток (рис. 3.15).

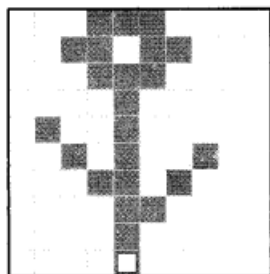


Рис. 3.14

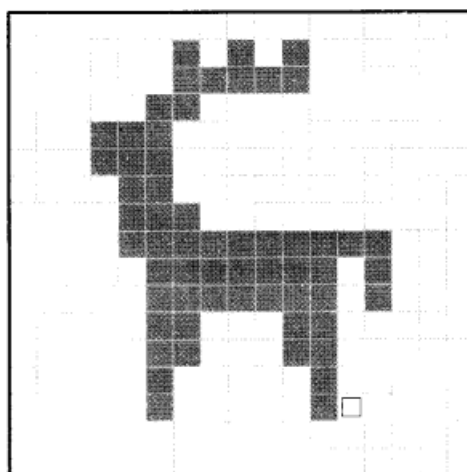


Рис. 3.15

7. Подведение итогов.

Наш урок подходит к концу, давайте поговорим о затруднениях в заданиях. Кому какое задание показалось сложнее?

(оцениваю работу учащихся на уроке выборочно)

8. Информация о домашнем задании.

§ 9

9. Этап рефлексии.

Предлагаю закончить фразы:

«Сегодня я понял, что...»

«Урок научил...»

«Я бы изменил в уроке ...»

Конспект ученика

Исполнитель Робот

08.12.2023

Робот — автоматическое устройство, которое действует по заранее составленной программе.

Команда	Действие
Right	Перемещает Робота вправо
Left	Перемещает Робота влево
Up	Перемещает Робота вверх
Down	Перемещает Робота вниз
Paint	Закрашивает текущую ячейку

```
uses Robot;  
begin  
Field (10,10);  
<команды>;  
end.
```

Оформление классной доски

Анна Андреевна	Исполнитель Робот 08.12.2023 uses Robot; begin Field (10,10); <команды>; end.	§ 9
-------------------	--	-----