



Особенности изучения темы "Исполнители и алгоритмы" на уроках информатики в 7 классе: практические приемы для профильного уровня.

Мирная Мария Ивановна,
учитель информатики
МАОУ СОШ № 111 г. Краснодара

Тема «Исполнители и алгоритмы»

- закладывает фундамент алгоритмического мышления — навыка, востребованного в современном мире цифровых технологий;
- формирует логическое мышление, учит анализировать условия и прогнозировать результаты;
- обучает структурированию информации, это помогает систематизировать мысли и избегать хаотичности в рассуждениях;
- развивает внимание к деталям, так как малейшая ошибка в алгоритме (пропущенная команда, неверная последовательность) приводит к сбою

Особенности профильного уровня

На профильном уровне изучение темы выходит за рамки знакомства с понятиями.

Ученики:

- углублённо изучают алгоритмические конструкции (следование, ветвление, цикл);
- осваивают метод последовательной детализации;
- учатся анализировать и оптимизировать алгоритмы;
- применяют алгоритмы для решения сложных задач.

Использование учебных исполнителей Кумир. Робот

Новая программа - Кумир

Программа Редактирование Вставка Выполнение Окна Практикум Чертежник Робот Инфо »

```
1 алг
2 нач
3 .
4 кон
```

Робот - нет файла

Стр: 1, Кол: 1

```
алг
нач
влево
вниз
нц 3 раз
вправо
кц
нц 4 раз
закрасить
вниз
кц
кон
```

```
алг
нач
влево
вниз
нц пока справа свободно
вправо
кц
нц пока не справа свободно
закрасить
вниз
кц
кон
```

Использование учебных исполнителей Кумир. Черепаха

К Новая программа* - Кумир

Программа Редактирование Вставка Выполнение Окна Практикум Чертежник Робот Инфо >

1 использовать Черепаха
2 алг
3 нач
4 . опустить хвост
5 . нц 15 раз
6 . . вперед(5)
7 . . вправо(108)
8 . кц
9 кон

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

Черепаха

Сетка:
1.0000
 авто
Масштаб:
1.85:1
+ 1:1 -
Весь чертеж

Робот Черепаха

Анализ Выполнено шагов: 65

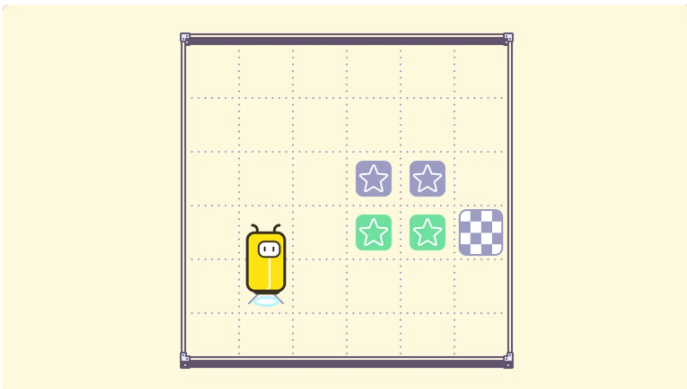
Стр: 15, Кол: 1 рус

Использование учебных исполнителей Яндекс Учебник. Робот (Blockly)

Клетки поля можно красить

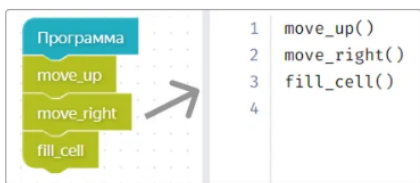
Одни клетки (зелёные) были покрашены до начала работы программы. А другие (серые) должен закрасить робот.

Клетка, в которой должен остановиться робот в конце работы программы, отмечена финишным флажком.



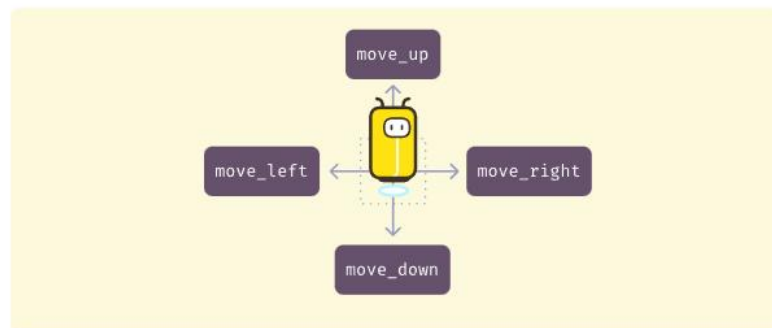
Код на Python

При составлении программы из блоков код дублируется на языке Python в правой части экрана.

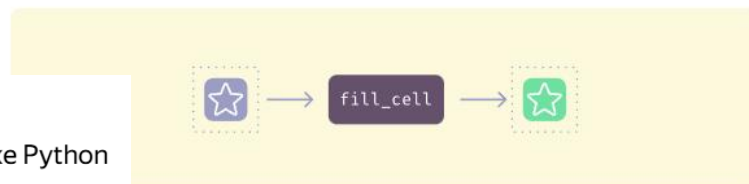


Команды для робота

```
move_up шаг вверх  
move_left шаг влево  
move_right шаг вправо  
move_down шаг вниз
```



fill_cell закрасить клетку

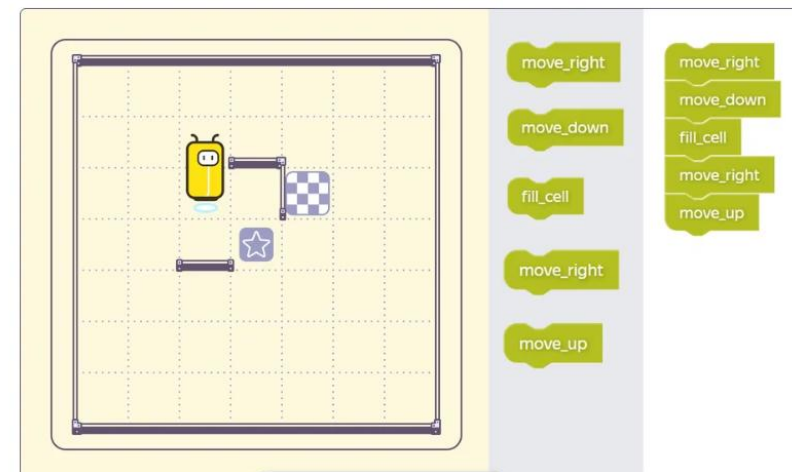


Как писать программы

Программа для робота состоит из последовательности команд. Пока мы будем использовать в программах только 5 команд: move_left, move_right, move_up, move_down и fill_cell.

Команды можно перетаскивать с панели, находящейся рядом с полем ввода. Каждую команду можно разместить в любом месте уже составленной программы.

Команды выполняются последовательно, начиная с первой, сверху вниз.





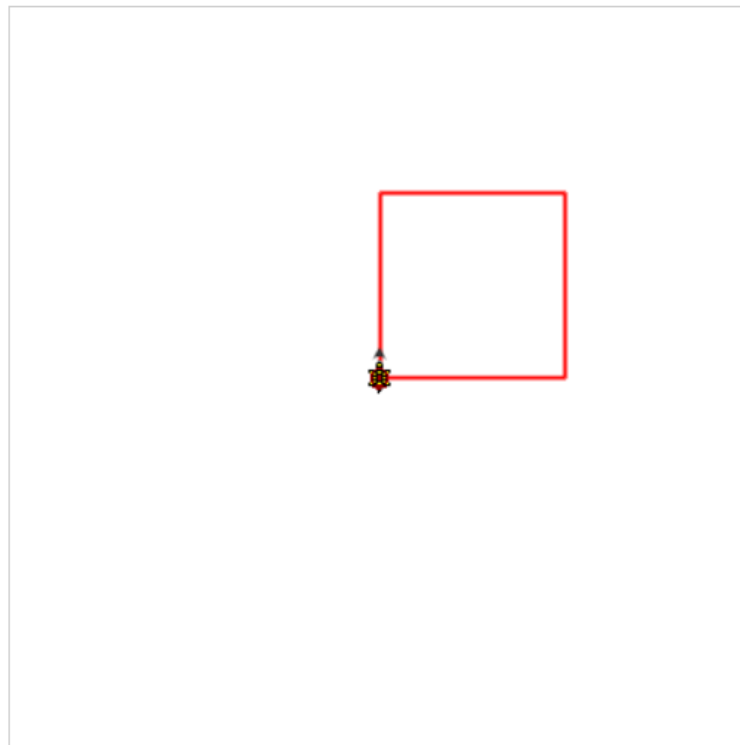
Использование учебных исполнителей Черепаша (Blockly): свободное рисование

← (Я) ↻ kpolyakov.spb.ru

Черепаша: свободное рисование

Черепаша: свободное рисование

  Ты можешь добавить в программу **92 блока**.



- Черепаша
- Цвет
- Циклы
- Переменные
- Математика
- Процедуры

Программа

- опусти перо
- цвет пера ■
- повторить 4 раз
- выполнить
 - вперёд на 100
 - поверни вправо на 90°

Программа на языке Python

```
1 penDown()  
2 penColour('#ff0000')  
3 for count in range(4):  
4     forward(100)  
5     right(90)  
6
```



✕ Сброс

Выполнено операций: 10

Пошаговая сложность задач

Задачи выстраиваются по принципу «от простого к сложному»:

Базовый уровень: составить алгоритм для перемещения исполнителя в точку.

Средний уровень: добавить условия (ветвления) — например, обойти препятствие.

Продвинутый уровень: использовать циклы и вспомогательные алгоритмы — например, нарисовать узор.

«Человек-исполнитель»: один ученик задаёт команды, другой выполняет их буквально (например, пройти до доски). Это наглядно демонстрирует важность точности алгоритма.

Соревнования: кто быстрее и эффективнее решит задачу для исполнителя.

Квесты

Урок 0. Квест в Яндекс Музее: история развития вычислительной техники

Зал 7. Умные устройства и искусственный интеллект

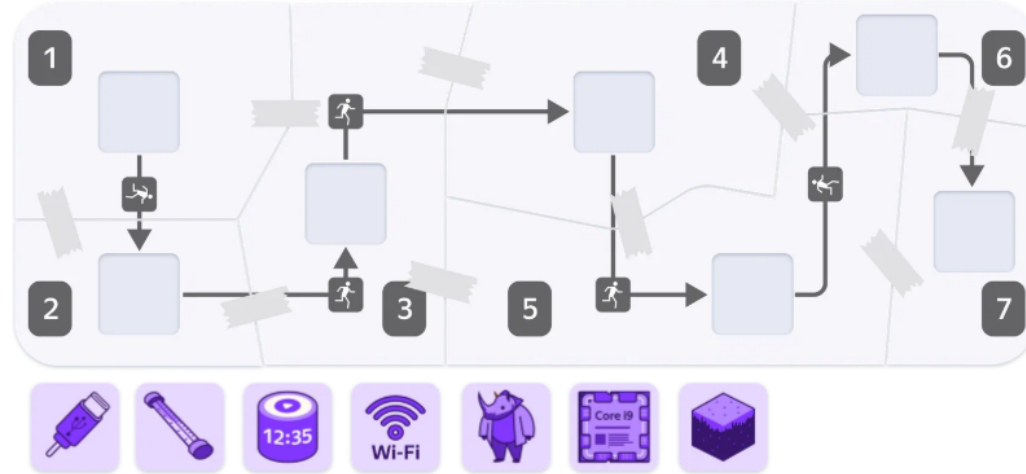
Найдите в тексте один неверный факт и объясните, почему это ошибка.

Пропустим несколько больших шагов в развитии компьютеров и перейдём к двум самым современным: умным устройствам и нейросетям.

Карта Яндекс Музея

0 из 2

Соберите карту хроноошибок по всем залам музея.



Для быстрого перемещения воспользуйтесь телепортами в отдельные залы:

[Зал 1. Первые вычислительные устройства](#)

[Зал 2. Ламповая эпоха](#)

[Зал 3. Эпоха транзисторов](#)

[Зал 4. Эпоха интегральных схем](#)

[Зал 5. Золотая эпоха](#)

[Зал 6. Появление интернета](#)

● ○ ○ Ответить

Работа с разными формами представления алгоритмов

Ученики учатся переводить алгоритмы из одной формы в другую:

Словесное описание — тренировка чёткости формулировок.

Блок-схемы — развитие структурного мышления.

Программный код — переход к языкам программирования.

Ученикам предлагаются алгоритмы с ошибками. Задача — найти и исправить их. Это развивает критическое мышление и внимательность.

Задание 1. Сумма чётных чисел

Условие. Напишите программу, которая считывает 5 целых чисел и выводит сумму чётных из них. Если чётных чисел нет, вывести 0.

Ошибочный код (Python):

```
python ⇨ Переносить ↶ Свернуть 📄 Копировать

sum = 0
for i in range(5):
    num = int(input())
    if num % 2 == 1: # Ошибка здесь
        sum += num
print(sum)
```

Вопросы для учеников:

1. Что выведет программа при вводе чисел: 2, 3, 4, 5, 6?
2. В чём заключается ошибка?
3. Исправьте код.

Основы программирования – предмет в учебном плане



Каталог ▾

Моё обучение

Преподавание



+ Новый курс

📖 Курсы

📄 Уроки

🎓 Классы ▾

Мои классы

Классы



7Ж_ОП_111: "Поколение Python": курс для начинающих

👤 23

2. Ввод-вывод данных

 37/209

Данный модуль посвящен знакомству с Python и основам ввода-вывода данных. Рассматриваются функции `print()` и `input()`, параметры `sep` и `end`, переменные, оформление кода по PEP 8, а также целочисленная арифметика.

1	2.1 Введение. Знакомство с Python	8 / 8
2	2.2 Команды <code>print()</code> и <code>input()</code> , переменные	23 / 52
3	2.3 Параметры <code>sep</code> и <code>end</code> , PEP8	5 / 38
4	2.4 Целочисленная арифметика. Часть 1	0 / 49
5	2.5 Целочисленная арифметика. Часть 2	1 / 62

Тема урока: работа с целыми числами

1. Целочисленный тип данных
2. Преобразование строки к целому числу
3. Операции над целыми числами
4. Частые ошибки

Аннотация. Урок посвящен изучению работы с целыми числами. Изучим основные операции, а также научимся преобразовывать строки в числа.

Целые числа

Все предыдущие программы, которые мы писали, работали со строковыми данными. Действительно, команда `input()` считывает строку текста. Однако во многих случаях нам нужно работать именно с числами. Чтобы в Python создать переменную целого типа данных, нужно опустить кавычки при объявлении переменной. Рассмотрим следующий код:

```
num1 = 7           # num1 - это число
num2 = 10          # num2 - это число
num3 = num1 + num2 # num3 - это число

print(num3)
```

В результате выполнения такой программы будет выведено число `17`.

Сумма трёх чисел


Напишите программу, которая считывает три целых числа и выводит на экран их сумму. Каждое число записано в отдельной строке.

Формат входных данных

На вход программе подаются три целых числа, каждое на отдельной строке.

Формат выходных данных

Программа должна вывести сумму введённых чисел.

▶ Тестовые данные 

Sample Input 1:

```
9
11
2
```

Sample Output 1:

```
22
```

Напишите программу. Тестируется через stdin → stdout

Из всех попыток **70%** верных

Time Limit: 15 секунд

Memory Limit: 256 MB

Python 3.6



1	
2	
3	
4	
5	
6	

Максимум 5 баллов за решение.

Отправить

Запустить код


Табель успеваемости курса "Поколение Python": курс для начинающих

7Ж_ОП_111

Поиск по ID, email, имени учащегося

Искать

Итог	Введение. Знакомство с Python				Команды print() и input(), переменные																Параметры sep и end, PEP8											
	Q3	Q4	Q5	Q6	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q2	Q3	Q4	Q5	Q6	Q7	Q8	
Итого: 2438	1	1	1	5	1	1	1	1	1	5	5	5	5	1	1	1	1	1	1	1	1	5	5	5	5	1	1	1	1	1	1	1
1094	1	1	1	5	1	1	1	1	1	5	5	5	5	1	1	1	1	1	1	1	1	5	5	5	5	1	1	1	1	1	1	1
283	1	1	1	5	1	1	1	1	1	0	5	5	5	1	1	1	1	1	1	1	5	5	5	5	1	1	1	1	1	1	1	
158	1	1	1	5	1	1	1	1	1	5	5	5	5	1	1	1	1	1	1	1	5	5	5	5	1	1	1	1	1	1	1	
158	1	1	1	5	1	1	1	1	1	–	–	–	–	1	1	1	1	1	1	1	0	–	–	–	1	1	1	1	1	1	1	
83	1	1	1	5	1	1	1	1	1	5	5	5	5	1	0	1	1	0	–	–	–	–	–	–	1	1	1	1	1	1	1	
61	1	1	1	5	1	1	1	1	1	5	5	5	5	1	1	1	1	1	1	1	0	0	0	–	1	1	1	1	1	1	1	
59	1	1	1	5	1	1	1	1	1	0	0	–	–	1	1	1	1	1	1	1	–	–	–	–	1	1	1	1	1	1	1	
58	1	1	1	5	1	1	1	1	1	5	5	5	5	1	1	1	1	1	1	1	0	–	–	–	1	1	1	1	1	1	1	
40	–	–	–	–	1	1	1	1	1	5	5	5	5	1	1	1	1	0	1	1	0	–	–	–	–	–	–	–	–	–	–	
37	–	–	–	–	1	1	1	1	1	5	5	5	5	1	1	1	1	0	1	1	0	–	–	–	1	1	1	1	1	1	0	



Тема «Исполнители и алгоритмы» не просто знакомит школьников с информатикой — она развивает универсальные когнитивные навыки: логику, внимание, креативность, умение структурировать информацию и решать задачи поэтапно. Эти качества полезны в учёбе, будущей профессии и повседневной жизни.

