



ЗОЛОТОЕ  
СЕЧЕНИЕ

ФОНД ПОДДЕРЖКИ  
ТАЛАНТЛИВЫХ ДЕТЕЙ  
И МОЛОДЕЖИ

ВС{З}Ш

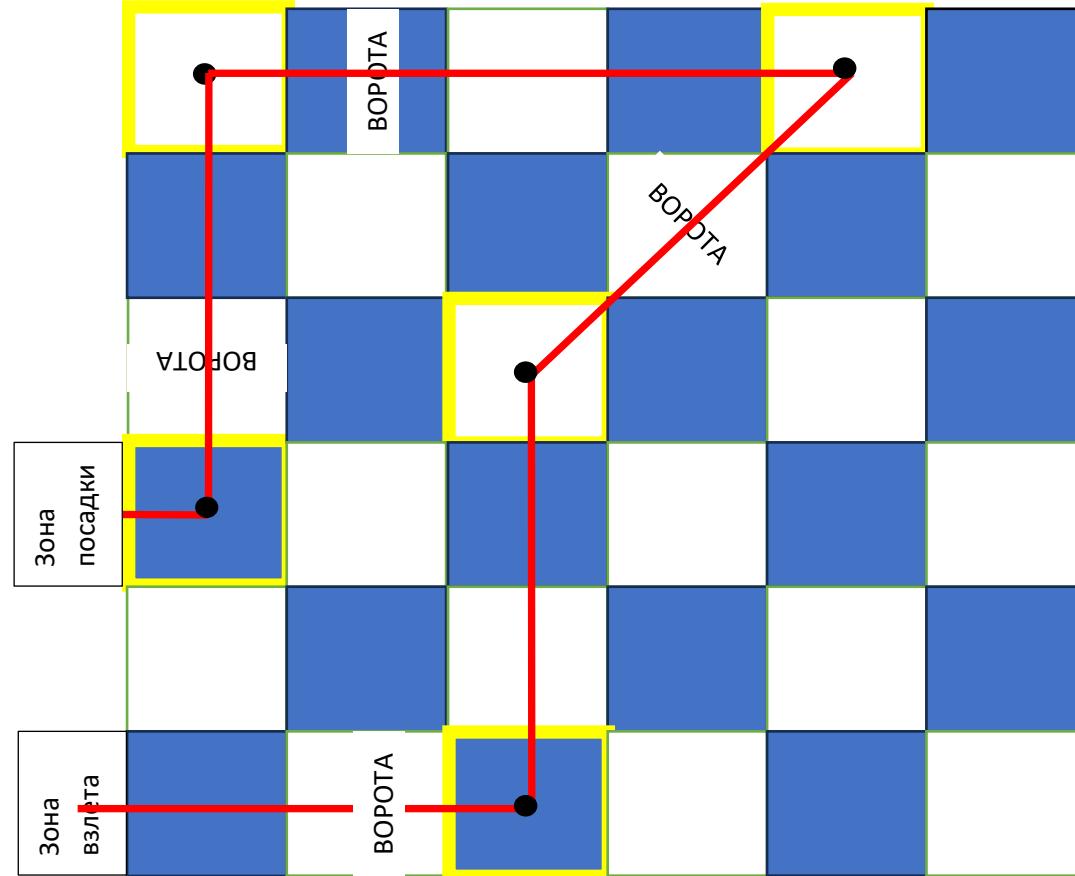
**Разбор заданий муниципального этапа  
всероссийской олимпиады школьников  
по труду (технологии)  
«Программирование пилотного задания  
беспилотного летательного аппарата», 7-8 класс**

**2025/2026 учебного года  
в Свердловской области**

Разработчик –  
Кощеева Елена Сергеевна, доцент  
УрГПУ,  
Омельченко Никита Сергеевич,  
магистрант УрГПУ



# Макет полетного поля



# Полетное задание

- 1. Взлёт из произвольной зоны.
- 2. Пролететь через ворота, расположенные через одну клетку от зоны взлёта.
- 3. Зависнуть над первой точкой и удерживать позицию 2 секунды.
- 4. Выполнить поворот на  $90^\circ$  влево.
- 5. Двинуться вперёд на 3 клетки, зависнуть над следующей точкой, удерживая позицию 3 секунды.
- 6. Выполнить круговой разворот (на  $360^\circ$ ), зависнуть ещё 2 секунды.
- 7. Выполнить поворот на  $45^\circ$  вправо, после чего пролететь через ворота.
- 8. Зависнуть над точкой за воротами, удерживая позицию 2 секунды.
- 9. Повернуть на  $135^\circ$  влево и пролететь через ворота (через 2 клетки).
- 10. После прохождения ворот выполнить зависание над точкой и тонкий маневр поворота на  $90^\circ$  влево.
- 11. Пролететь через ворота (через 1 клетку).
- 12. Зависнуть в точке, удерживая позицию 2 секунды.
- 13. Выполнить финальный поворот на  $90^\circ$ , и посадить дрон в зону посадки.

# Разбор задания

Разбор задания будет приведен в примере кода на языке программирования PYTHON

Для начала инициализируем начальное положение робота

```
import time
import math

class Drone:
    def __init__(self, x=0, y=0, altitude=0, direction=0):
        """
        Инициализация дрона.
        x, y: координаты в плоскости (клетки)
        altitude: высота (см)
        direction: направление (градусы, 0 - север, 90 - восток,
        180 - юг, 270 - запад)
        """
        self.x = x
        self.y = y
        self.altitude = altitude
        self.direction = direction # по часовой
        print(f"Дрон инициализирован в точке ({self.x}, {self.y}),
высота {self.altitude} см, направление {self.direction}°." )
```

Для удобства создадим функции для осуществления перемещения в каждом из направлений

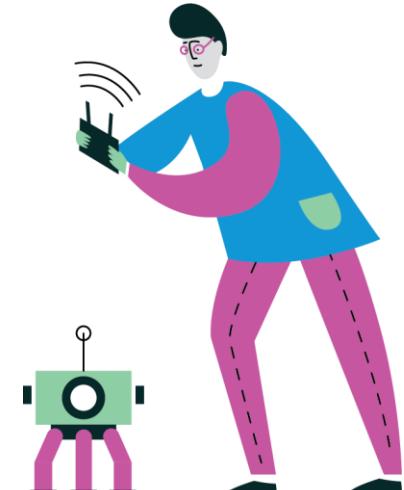
```
def take_off(self, height_cm):
    print(f"Взлёт на высоту {height_cm} см.")
    self.altitude = height_cm
    time.sleep(1)

def move_forward(self, cells):
    # Перемещение вперед в зависимости от направления
    rad = math.radians(self.direction)
    dx = round(math.cos(rad))
    dy = round(math.sin(rad))
    self.x += dx * cells
    self.y += dy * cells
    print(f"Перемещение вперед на {cells} клеток к точке
({self.x}, {self.y}).")
    time.sleep(1)

def turn_left(self, degrees=90):
    self.direction = (self.direction - degrees) % 360
    print(f"Поворот налево на {degrees}°, новое направление
{self.direction}°.")
    time.sleep(0.5)

def turn_right(self, degrees=90):
    self.direction = (self.direction + degrees) % 360
    print(f"Поворот направо на {degrees}°, новое направление
{self.direction}°.")
    time.sleep(0.5)
```

**take\_off** функция взлета на заданную высоту  
**move\_forward** функция перемещения вперед  
**turn\_left** функция поворота налево  
**turn\_right** функция поворота направо



**hover** функция зависания над точкой  
**rotate\_around** функция вращения на 360 градусов  
**change\_altitude** функция изменения высоты  
**land** функция посадки

```
def hover(self, seconds):
    print(f"Зависание над точкой ({self.x}, {self.y}),
удержание {seconds} секунд.")
    time.sleep(seconds)

def rotate_around(self, degrees):
    # Имитация кругового разворота
    print(f"Круговой разворот на {degrees}°. ")
    self.direction = (self.direction + degrees) % 360
    time.sleep(1)

def change_altitude(self, delta_cm):
    self.altitude += delta_cm
    print(f"Изменение высоты на {delta_cm} см. Текущая высота
{self.altitude} см.")
    time.sleep(1)

def land(self):
    print("Посадка в зоне назначения.")
    self.altitude = 0
    time.sleep(1)
```

# Написание основного кода ВСОШ задания 1-5

Для дальнейшего написания программы вам необходимо использовать выше написанные функции

*# 1. Взлет из произвольной зоны*

```
drone.take_off(50)
```

*# 2. Пролететь через ворота, расположенные через одну клетку  
(через 2 клетки)*

```
drone.move_forward(2)
```

*# 3. Зависнуть над первой точкой, удерживать 2 секунды*

```
drone.hover(2)
```

*# 4. Поворот на 90° влево*

```
drone.turn_left(90)
```

*# 5. Вперед на 3 клетки, зависнуть 3 секунды*

```
drone.move_forward(3)
```

```
drone.hover(3)
```

# Написание основного кода задания 6-8

# 6. Круговой разворот (на 360°), зависание 2 секунды

```
drone.rotate_around(360)  
drone.hover(2)
```

# 7. Поворот на 45° вправо, пролет через ворота (через 2  
клетки)

```
drone.turn_right(45)  
drone.move_forward(2)
```

# 8. Зависание над точкой, удержание 2 секунды

```
drone.hover(2)
```

# Написание основного кода задания 9-13

# 10. После прохождения ворот выполнить зависание и тонкий маневр поворота на 90° влево

```
hover(2)
```

```
rotate(90)
```

# 11. Пролететь через ворота (через 1 клетку)

```
move_forward(1)
```

# 12. Зависнуть в точке, удерживая позицию 2 секунды

```
hover(2)
```

# 13. Выполнить финальный поворот на 90° вправо, и посадить дрон

```
rotate(-90)
```

```
land()
```

