



ЗОЛОТОЕ  
СЕЧЕНИЕ

ФОНД ПОДДЕРЖКИ  
ТАЛАНТЛИВЫХ ДЕТЕЙ  
И МОЛОДЕЖИ

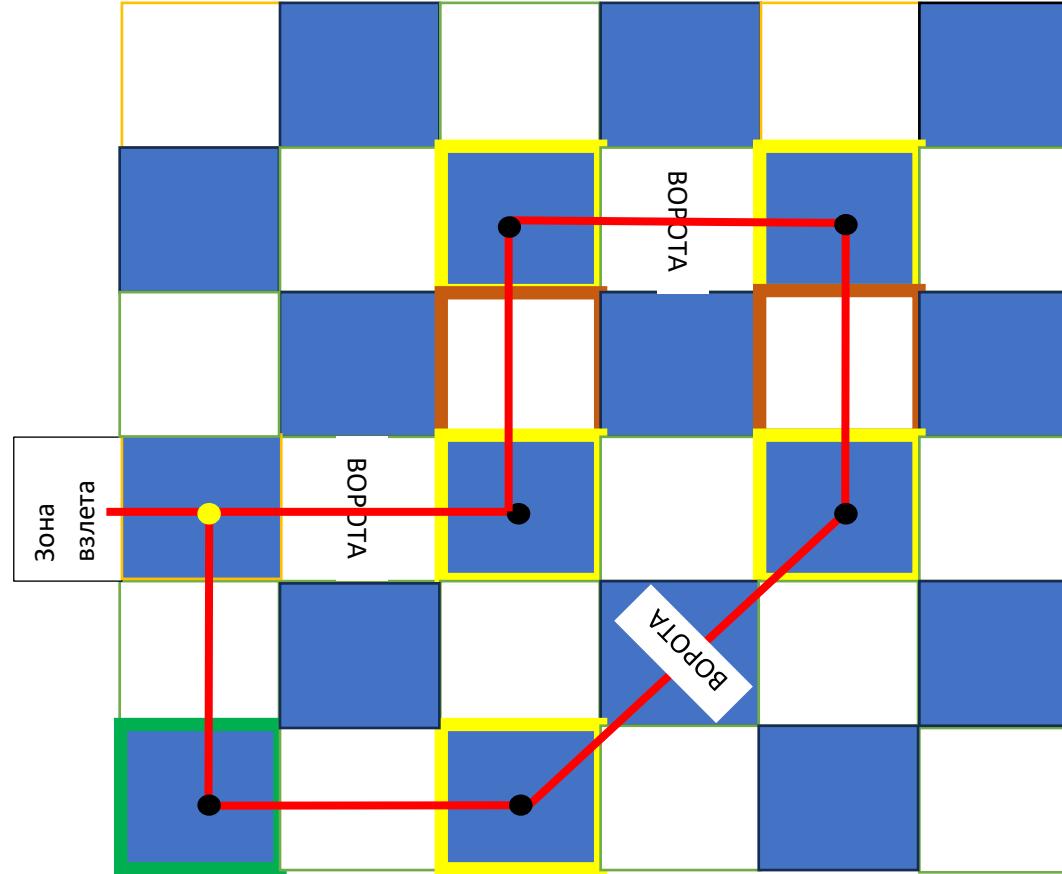
**Разбор заданий муниципального этапа  
всероссийской олимпиады школьников  
по труду (технологии)  
«Программирование пилотного задания  
беспилотного летательного аппарата», 9 класс**

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

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



# Макет полетного поля



муниципальный этап 2025/2026



# Полетное задание

1. Взлететь из произвольной зоны взлёта (базовая высота полета 1 клетка – 50 см).
2. Пролететь через ворота, расположенные через одну клетку от зоны взлёта.
3. Зависнуть над точкой, удерживать позицию 2 секунды.
4. Выполнить поворот на 90° влево, подняться на высоту 30 см и пролететь башню (высота башни 1,5 клетки – 75 см), расположенную в следующей клетке и занять высоту до осуществления подъема.
5. Зависнуть и удерживать позицию 3 секунды.
6. Выполнить поворот на 90° вправо.
7. Пролететь через ворота, расположенные в следующей клетке от зоны зависания.
8. Зависнуть над следующей точкой, удерживая позицию 2 секунды.
9. Выполнить поворот на 90° вправо.
10. Подняться на высоту 30 см и пролететь башню (высота башни 1,5 клетки – 75 см), расположенную в следующей клетке и занять высоту до осуществления подъема.

# Полетное задание

11. Зависнуть над точкой, удерживая 2 секунды.
12. Выполнить поворот на  $45^\circ$  вправо.
13. Пролететь через ворота, расположенные в следующей клетке от зоны зависания.
14. Зависнуть над следующей точкой, удерживая позицию 2 секунды.
15. Выполнить поворот на  $45^\circ$  вправо.
16. Пролететь вперед на расстояние двух клеток и совершить посадку (импровизация забора груза) на 4 секунды.
17. Осуществить взлет на высоту 1 клетки, повернуть на  $90^\circ$  вправо и пролететь вперед на расстояние двух клеток.
19. Осуществить посадку и выгрузку груза в контрольной точке.

# 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 задания 1-4

Для дальнейшего написания программы вам необходимо использовать выше написанные функции

```
# 1. Взлёт из произвольной зоны (базовая высота 50 см, 1 клетка)
takeoff()
change_altitude(50) # 50 см = 1 клетка
time.sleep(1)

# 2. Пролёт через ворота, расположенные через одну клетку
move_forward(1)
move_forward(1) # Ворота через одну клетку от зоны взлёта

# 3. Зависнуть над точкой, удерживать 2 сек
hover(2)

# 4. Поворот на 90° влево, подняться на 30 см, пролететь башню (75 см)
rotate(90)
change_altitude(30) # Подъём на 30 см
move_forward(1) # В следующей клетке – башня
change_altitude(75) # Занять высоту башни (75 см)
```

# Написание основного кода задания 5-8

# 5. Зависнуть и удерживать 3 секунды  
`hover(3)`

# 6. Поворот на 90° вправо  
`rotate(-90)`

# 7. Пролететь через ворота в следующей клетке  
`move_forward(1)`

# 8. Зависнуть над точкой, удерживать 2 секунды  
`hover(2)`

# Написание основного кода задания 9-13

# 9. Поворот на 90° вправо

```
rotate(-90)
```

# 10. Подняться на 30 см, пролететь башню в следующей клетке, занять высоту

```
change_altitude(30)
```

```
move_forward(1)
```

```
change_altitude(75) # На высоте башни
```

# 11. Зависнуть и удерживать 2 секунды

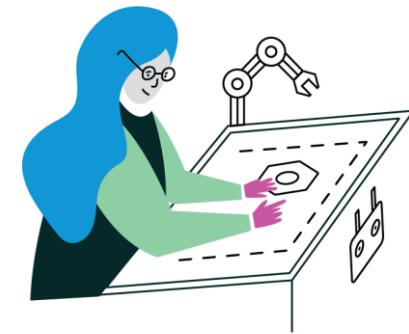
```
hover(2)
```

# 12. Поворот на 45° вправо

```
rotate(-45)
```

# 13. Пролететь через ворота в следующей клетке

```
move_forward(1)
```

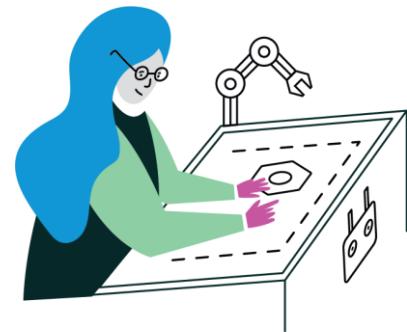


# Написание основного кода задания 14-16

```
# 14. Зависнуть над точкой, удерживать 2 сек
hover(2)

# 15. Поворот на 45° вправо
rotate(-45)

# 16. Пролететь вперед на 2 клетки и выполнить посадку (забор груза) на 4
сек
move_forward(2)
print("Забор груза (импровизация).")
perform_task("Забор груза")
time.sleep(4)
```



# Написание основного кода задания 17-19

```
# 17. Взлет на высоту 1 клетки (50 см)
change_altitude(50)
# Повернуть на 90° вправо
rotate(-90)
# Пролететь вперед на 2 клетки
move_forward(2)

# 19. Посадка и выгрузка груза
land()
perform_task("Выгрузка груза")
print("Миссия завершена.")
```

