

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс № 3 «GridMaster: Разработка интерпретатора  
для двумерного перемещения»**

---

**1. Условия**

В эпоху цифровизации и роботизации развитие навыков программирования и алгоритмизации становится все более важным. Во многих современных школьных учебных планах появляются курсы по программированию и компьютерным наукам. Один из захватывающих методов погружения учащихся в эту тему — это задача создания интерпретатора для программирования движения исполнителя по двумерной сетке. Разработка интерпретатора для движения исполнителя по двумерной сетке, представляет собой задачу, применимую в областях программирования, робототехники и искусственного интеллекта, где важно умение создавать эффективные алгоритмы и управлять сложными системами. Участникам олимпиады предлагается разработать интерпретатор, способный перемещать исполнителя по полю 21x21 (стартовая координата (0; 0)).

**2. Техническое задание**

Требуется разработать язык программирования для управления движением исполнителя по двумерной сетке, а также интерпретатор для этого языка. Язык должен поддерживать основные конструкции программирования, включая алгоритмы, циклы, процедуры, условные операторы и переменные. В рамках задачи участникам необходимо создать среду исполнения, которая будет хранить и воспроизводить последовательность команд и координат исполнителя. Среда исполнения должна иметь возможность загрузки команд для исполнителя из .txt файла. Область видимости переменных в этом интерпретаторе глобальная, что означает, что переменные, объявленные в любом месте программы, будут доступны во всех последующих частях программы. Переменные можно объявлять в процедурах, и они сохраняют свои значения на протяжении всей

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс № 3 «GridMaster: Разработка интерпретатора  
для двумерного перемещения»**

---

программы. Переменные, используемые в блоках IFBLOCK и REPEAT, также имеют глобальную область видимости и будут сохранять свои значения после завершения этих блоков.

Приведем список команд и их описание (N - натуральное число с допустимым диапазоном от 1 до 1,000):

Команда	Описание
RIGHT N	Переместить исполнителя на N клеток вправо.
LEFT N	Переместить исполнителя на N клеток влево.
UP N	Переместить исполнителя на N клеток вверх.
DOWN N	Переместить исполнителя на N клеток вниз.
IFBLOCK DIR	Проверить препятствие в направлении DIR (RIGHT, LEFT, UP, DOWN). Препятствием являются края сетки. Если есть препятствие, выполнить следующие команды до ENDIF.
ENDIF	Завершить блок команд после IFBLOCK DIR.
REPEAT N	Повторить следующую команду (или блок команд до ENDREPEAT) N раз.
ENDREPEAT	Завершить блок команд после REPEAT N.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс № 3 «GridMaster: Разработка интерпретатора  
для двумерного перемещения»**

SET X = N	Задать значение переменной X равным N.
PROCEDURE NAME	Начать определение процедуры* с заданным именем.
ENDPROC	Завершить определение процедуры.
CALL NAME	Вызвать ранее определенную процедуру по имени.

\*Процедура - это набор инструкций, который может быть вызван по имени в любом месте программы. Процедуры используются для группировки инструкций в блоки, что делает код более организованным и удобным для понимания и отладки.

А также пример элементарного скрипта на полученном языке:

```
REPEAT 3
  IFBLOCK RIGHT
    RIGHT 5
  ENDIF
ENDREPEAT

DOWN 10

PROCEDURE MoveLeftAndUp
  LEFT 2
  UP 2
```

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс № 3 «GridMaster: Разработка интерпретатора  
для двумерного перемещения»**

ENDPROC

CALL MoveLeftAndUp

*Функциональное задание:*

Обязательная функциональность программы:

- Ввод и выполнение команд в соответствии с установленными правилами языка, которые были разработаны для обучения основам программирования учащихся.
- Поддержка основных конструкций языка, включая циклы и условные операторы.
- Возможность объявления и использования переменных.
- Разработка пользовательского интерфейса с окном для редактирования команд, окном для отображения исполнителя на двумерной сетке, а также кнопками "Run" и "Stop" для управления выполнением команд.
- Возможность загрузки исходного кода из .txt файла.
- Поддержка вложенности конструкций языка, таких как вложенные циклы и условные блоки, что позволяет создавать более сложные алгоритмы и улучшает понимание учащимися структур контроля.

Ограничения на вложенность конструкций:

Максимальный уровень вложенности конструкций (например, циклов, условных операторов и процедур) должен быть ограничен до 3 уровней. Это значит, что в любой момент код может иметь не более трех вложенных конструкций.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс № 3 «GridMaster: Разработка интерпретатора  
для двумерного перемещения»**

---

**Пример:**

```
REPEAT 5
  IFBLOCK RIGHT
    REPEAT 3
      RIGHT 2
    ENDREPEAT
  ENDIF
ENDREPEAT
```

*Дополнительная функциональность:*

Создание и вызов процедур:

- Учащиеся должны иметь возможность создавать и вызывать процедуры для группировки и повторного использования кода.

Обработка ошибок и исключений:

- Система должна обрабатывать ошибки и исключения, такие как попытка выхода исполнителя за пределы поля, и предоставлять информативные сообщения об ошибках.

Сохранение и загрузка последовательности команд:

- Предоставить возможность сохранения и загрузки написанных программ в базу данных вместо логирования отдельных команд.

Импорт и экспорт сохраненных программ:

- Учащиеся могут импортировать и экспортировать (из БД в .txt) сохраненные программы для дальнейшего воспроизведения и анализа.

Визуализация движения исполнителя:

- Поддержка графического интерфейса с анимацией для визуального отображения движения исполнителя по сетке.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс № 3 «GridMaster: Разработка интерпретатора  
для двумерного перемещения»**

---

- В интерфейсе должны присутствовать окно для редактирования команд, окно с отображением движения исполнителя по сетке, а также кнопки "run" и "stop" для управления выполнением программы.

**3. Требования к документации**

- **Титульный лист:** с указанием названия кейса и перечислением членов команды.

- **Анализ технических требований:** детализация спецификаций и интерпретация задачи. (Интерпретация задачи в данном контексте означает понимание и объяснение целей и требований, предъявляемых к разработке интерпретатора для программирования движения исполнителя по двумерной сетке. Это включает в себя анализ функциональных и дополнительных требований, определение основных и дополнительных возможностей системы, а также понимание желаемого пользовательского взаимодействия и выходных данных, которые должен предоставить разработанный интерпретатор.)

- **Обоснование используемых программных средств:** выбор языков программирования, фреймворков и других инструментов, и их обоснование.

- **Структурная и функциональная схемы программного продукта:** визуализация основных модулей, классов, функций и их взаимодействия.

- **Схема организации данных:** структура данных, диаграммы базы данных или другие релевантные элементы.

- **Описание проведенных испытаний в соответствии с регламентом кейса:** снимки экрана и/или запись экрана с демонстрацией работы программы.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс № 3 «GridMaster: Разработка интерпретатора  
для двумерного перемещения»**

---

- **Программный код:** ссылка на *Git-репозиторий* с полным набором файлов проекта.
- **README файл:** содержит подробное описание запуска программного обеспечения для разработчика и конечного пользователя.
- **Методы доставки кода:** необходимо предоставить удобные методы для доставки программного обеспечения пользователю. Это может включать в себя готовые установочные пакеты, докер-контейнеры или другие методы, упрощающие процесс установки без необходимости собирать код или устанавливать среду разработки.
- **Подробное описание проекта в репозитории:** объяснение архитектуры, основных модулей и их функциональности, а также любые другие релевантные детали.
- **Видеоролик,** демонстрирующий функционирование разработанного программного продукта в соответствии с регламентом испытаний. На видео или записи экрана необходимо продемонстрировать выполнение каждого, описанного в регламенте, испытания в соответствии с условиями. Видео должно однозначно подтверждать авторство участников (во время записи ролика необходимо четко произнести название команды, ФИО участников, номер школы, ФИО руководителя). Видеоролик располагается на стороннем видеохостинге (ВКонтакте, Rutube и др.), ссылка на видеоролик располагается в репозитории в системе контроля версий (может использоваться github, gitlab, bitbucket и др.).

#### **4. Регламент испытаний**

1) Проверка всех функциональных требований, указанных в техническом задании. Каждое требование должно быть проверено на корректность и соответствие спецификациям.

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс № 3 «GridMaster: Разработка интерпретатора  
для двумерного перемещения»**

---

2) Проверка возможности запуска команд из .txt файла. Участники должны демонстрировать, как их разработанный интерпретатор считывает и исполняет команды из внешнего текстового файла.

3) Проверка воспроизведения движения исполнителя по сохраненным координатам. После выполнения команд из текстового файла, система должна позволять сохранять координаты движения исполнителя и затем корректно воспроизводить данный маршрут.

4) Проверка работы интерпретатора на различных задачах. Необходимо тестировать программу на нескольких предоставленных организаторами скриптах из файлов формата .txt для убедительности работы интерпретатора.

5) Проверка обработки исключительных ситуаций.

а) Выход Исполнителя за Границы Сетки:

i) Попытка переместить исполнителя за пределы поля 21x21 (например, команда RIGHT 22 из позиции (0,0)).

b) Неверный Формат Команды:

i) Команды с неправильным синтаксисом (например, RIGH 5 вместо RIGHT 5).

ii) Команды с несуществующими операциями (например, JUMP 5).

c) Некорректные Значения Параметров:

i) Команды с недопустимыми значениями (например, LEFT -5 или UP 1001).

d) Обращение к Неопределенным Переменным:

i) Использование необъявленных переменных в командах (например, SET X = Y, где Y не была объявлена).

e) Ошибки в Циклах и Условных Блоках:

i) Бесконечные циклы (например, REPEAT 0).



**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс № 3 «GridMaster: Разработка интерпретатора  
для двумерного перемещения»**

---

ii) Неправильно оформленные циклы или условные блоки (например, отсутствует ENDREPEAT или ENDIF).

f) Ошибки в Определении и Вызове Процедур:

i) Вызов несуществующих процедур.

ii) Ошибки в определении процедуры (например, повторное объявление процедуры с тем же именем).

g) Превышение Максимального Уровня Вложенности Конструкций:

i) Программы, превышающие ограничение вложенности в 3 уровня.

h) Ошибки Чтения из Файлов:

i) Невозможность считывания команд из поврежденных или отсутствующих .txt файлов.

i) Проблемы Сохранения и Загрузки Программ:

i) Невозможность сохранить или загрузить программу из базы данных.

Для участников кейса по разработке интерпретатора движения исполнителя по двумерной сетке важно уделить особое внимание подготовке видеоматериалов, демонстрирующих корректную обработку каждой из вышеуказанных исключительных ситуаций.

**5. Примерный перечень средств и инструментов для выполнения задания**

1. [Что такое программирование?](#)
2. [Основы языка Python](#)
3. [Обучение Python для детей и подростков](#)
4. [Основы веб-разработки](#)
5. [Учебник по JavaScript](#)
6. [Понимание топологической сортировки](#)

**МОСКОВСКАЯ ПРЕДПРОФЕССИОНАЛЬНАЯ  
ОЛИМПИАДА ШКОЛЬНИКОВ**  
**Профиль «Информационные технологии»**  
**Командный кейс № 3 «GridMaster: Разработка интерпретатора  
для двумерного перемещения»**

---

7. [Топологическая сортировка в Python](#) (Топологическая сортировка - это процесс сортировки узлов ориентированного графа в линейный порядок, при котором каждый узел предшествует тем узлам, которые он направляет. В контексте нашего кейса, топологическая сортировка может быть использована для определения порядка выполнения процедур и операций, особенно при наличии зависимостей (например, если одна процедура зависит от результатов другой))
8. [Введение в базы данных](#)
9. [Работа с SQLite в Python](#)
10. [Работа с JSON в Python](#)
11. [Git How To](#) - Интерактивное введение в Git.
12. [GitHub Learning Lab](#) - Официальные интерактивные курсы от GitHub.
13. [GitHub Guides](#) - Краткие руководства по основным функциям GitHub.
14. [PLY \(Python Lex-Yacc\)](#) - Инструмент для написания лексических анализаторов и парсеров.
15. [Lark](#) - Современная библиотека парсинга для Python.
16. [autopep8 GitHub](#)
17. [Prettier GitHub](#)
18. [Google Java Format GitHub](#)