**Кейс «Спасение марсохода: отладка кода в экстремальных условиях»**

***(Учебно-методический комплект для анализа алгоритмов)***

**Цели кейса:**

1. Образовательные:
   * Научить школьников анализировать базовые структуры алгоритмов (следование, ветвление, циклы).
   * Развить навыки поиска логических ошибок без непосредственного исправления кода.
   * Познакомить с принципами тестирования программного обеспечения.
2. Практические:
   * Сформировать умение читать и интерпретировать код на Python.
   * Отработать методику пошаговой трассировки программ.
3. Мотивационные:
   * Показать применение программирования в реальных космических миссиях.
   * Вовлечь через игровую механику «спасения миссии».

**Описание кейса.**

**Сюжетная рамка:** Ученики становятся инженерами ЦУПа, которые получают аварийный сигнал от марсохода «Пионер-X». Из-за песчаной бури в его программном обеспечении возникли критические ошибки. Задача — проанализировать три проблемных фрагмента кода, чтобы специалисты могли отправить исправления.

**Компоненты:**

1. 3 задания (по одному на каждый тип алгоритма):
   * Линейный: Расчет координат движения;
   * Ветвление: Проверка температуры;
   * Цикл: Поиск воды в образцах.
2. Интерактивные таблицы для пошагового анализа:
   * Трассировка значений переменных;
   * Визуализация выполнения условий.
3. Бонус-материалы:
   * Стилизованные бейджи «Космический инженер»;
   * Звуковое сообщение из ЦУПа.

**Методические особенности:**

* Контекстная интеграция: Связь с реальными миссиями NASA
* Дифференциация: Задания имеют 2 уровня сложности (базовый/продвинутый).
* Форматы работы: Индивидуальный анализ + командное обсуждение.

**Результаты применения.**

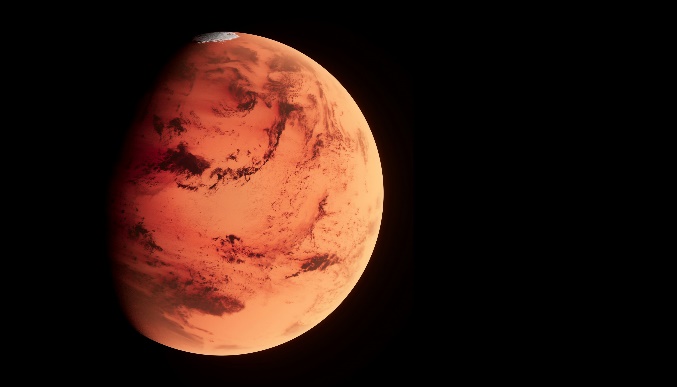
1. Для учеников:
   * Умение декомпозировать алгоритмы на составляющие.
   * Понимание важности точности условий в программировании.
2. Для учителей:
   * Готовый инструмент для уроков информатики в 7-9 классах.
   * Шаблон для создания аналогичных кейсов (например, про спутники или роботов-разведчиков).

Файлы кейса:

* [DOCX] Сценарий урока;
* [PPT] Презентация с визуализацией задач;
* [PDF] Распечатка для учеников;
* [HTML] Бонус-материал: стилизованные бейджи;
* [MP3] Бонус-материал: звуковое сообщение из ЦУПа.

**АНАЛИЗ КОДА МАРСОХОДА: СПАСАЕМ МИССИЮ**

***Экстренное сообщение из ЦУПа:***

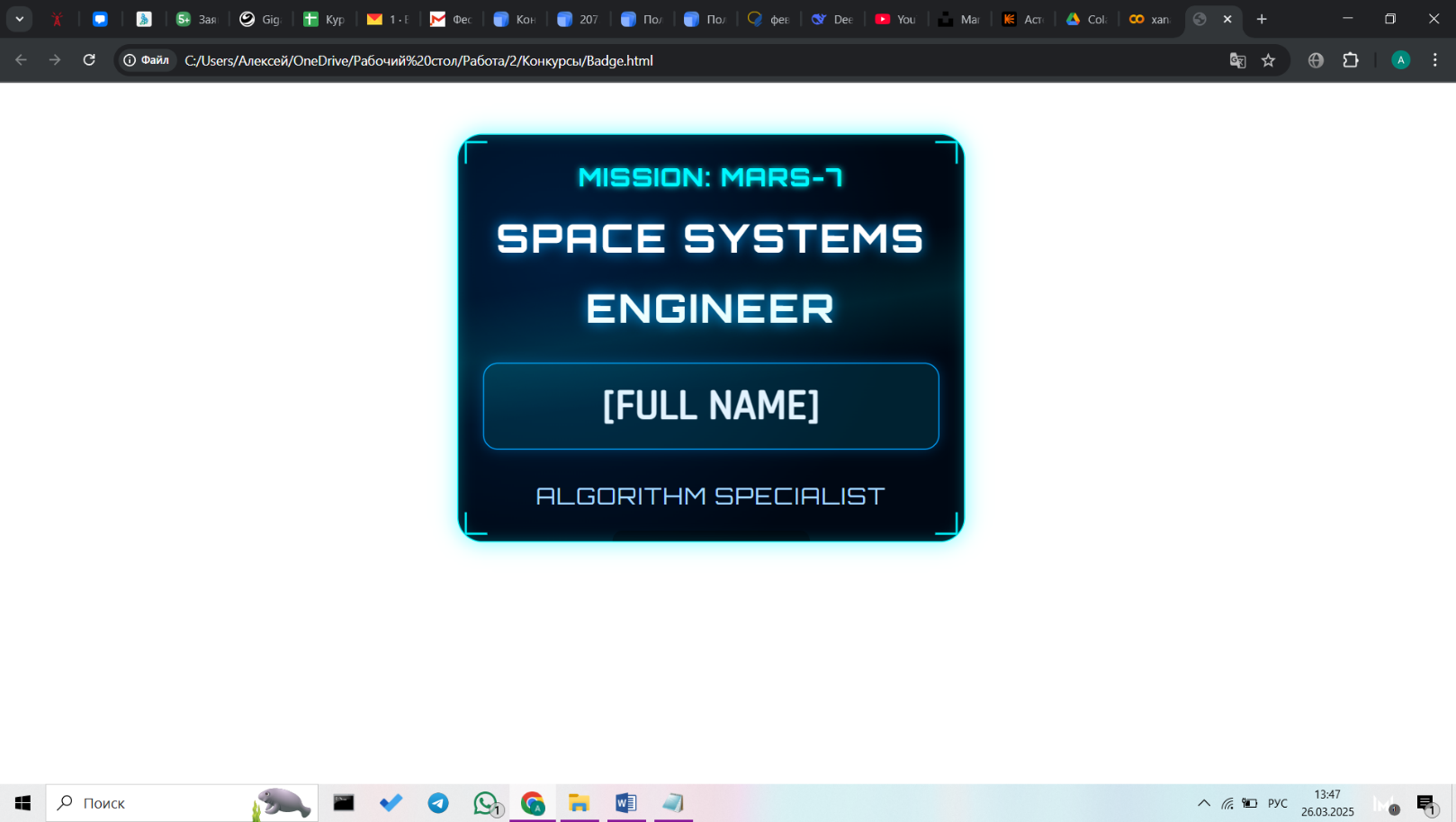
*****"Внимание всем операторам! Это Центр управления космическими миссиями. Только что получен сигнал бедствия от марсохода 'Пионер-X', который находится в долине Маринер на Марсе. Из-за песчаной бури в его программное обеспечение закрались критические ошибки!*

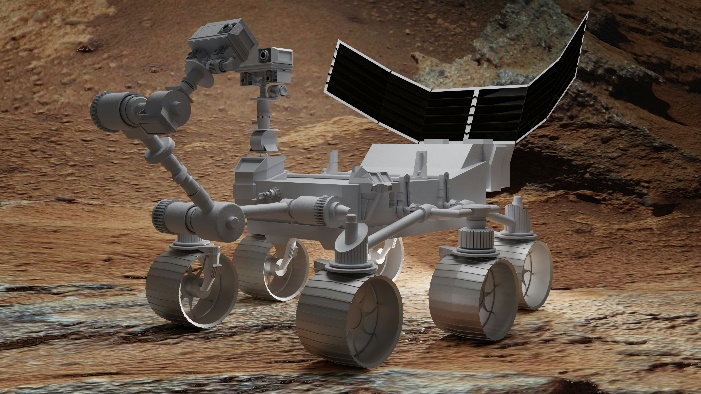
*Робот не может правильно анализировать образцы грунта, его система навигации дает сбои, а температурные датчики передают противоречивые данные. Если мы не исправим код в ближайшие 24 часа, миссия стоимостью 2 миллиарда долларов окажется под угрозой!*

*Все студенты Академии юных космических инженеров срочно привлекаются к анализу проблемных алгоритмов. Ваша задача — изучить три ключевых фрагмента кода, понять логику их работы и выявить потенциально опасные места. Позже на основе ваших отчетов специалисты отправят исправленные программы на Марс!"*

****

**Детали миссии**

1. **Объект**: Марсоход "Пионер-X"
   * Год запуска: 2024
   * Место посадки: Кратер Езеро
   * Задачи: поиск следов древней жизни, анализ грунта, картографирование
2. **Хронология ЧП**:
   * **Сол 145**: Началась песчаная буря
   * **Сол 146**: Первые сбои в навигации
   * **Сол 147**: Ошибки в анализе образцов
   * **Сол 148 (СЕЙЧАС)**: Критическое состояние
3. **Ваша роль**:
   * Аналитик алгоритмов 1-го класса
   * Доступ: Только чтение кода (исправления будут делаться позже)
   * Цель: Составить подробные отчеты по каждому фрагменту

**Технические параметры**

* **Язык программирования**: Python 3.9 (специальная космическая версия)
* **Ограничения**:
  + Связь с Марсом: Задержка 15 минут
  + Энергия: 40% заряда батарей
  + Память: 128 Мб свободно

*Ваш анализ напрямую повлияет на спасение миссии! Лучшие отчеты будут отправлены в NASA и отмечены сертификатами 'Юный космический инженер'*

*Расшифровки:*

*ЦУП – Центр управления полетами*

*Сол – марсианские сутки = 24ч 39мин*

**Задание 1: Движение по координатам**

**Код**:

x = 5

y = 3

x = x + 2

y = y - 1

print("Текущие координаты:", x, y)

Проанализируйте программу и заполните таблицу:

|  |  |  |  |
| --- | --- | --- | --- |
| **Шаг** | **Переменная/Действие** | **Значение** | **Пояснение** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Вопросы**:

1. Куда переместится робот, если начальные координаты будут x = 0, y = 0?
2. Как изменится код, чтобы робот двигался влево?

**Задание 2: Проверка температуры**

**Код**:

temperature = -15

if temperature < -20:

print("Включить аварийный обогрев!")

elif temperature < 0:

print("Включить стандартный обогрев")

else:

print("Температура в норме")

Проанализируйте программу и заполните таблицу:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Шаг** | **Переменная/Действие** | **Тип операции** | **Логическое выражение** | **Значение** | **Пояснение** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Вопросы**:

1. Что выведет программа, если temperature = -25?
2. Какой диапазон температур покрывает elif?

**Задание 3: Поиск воды в образцах**

**Код:**

samples = ["песок", "пыль", "лёд", "камень"]

found = False

for sample in samples:

if sample == "лёд":

found = True

print("Образец льда найден!")

if not found:

print("Лёд не обнаружен")

Проанализируйте программу и заполните таблицу:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Итерация цикла** | **Значение параметра sample** | **Проверка условия sample == "лёд":** | **Действие** | **Значение переменной found** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Вопросы**:

1. Сколько раз проверится условие if sample == "лёд"?
2. Что выведет программа, если убрать строку found = True?

**Заполненные таблицы (Ответы)**

Задание 1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Шаг** | **Переменная/Действие** | **Значение** | **Пояснение** |
| 1 | x | 5 | Начальная координата **x** |
| 2 | y | 3 | Начальная координата **y** |
| 3 | x = x + 2 | 7 | Робот сдвинулся вправо на 2 |
| 4 | y = y - 1 | 2 | Робот сдвинулся вниз на 1 |
| 5 | print("Текущие координаты:", x, y) | 7, 2 | Новые координаты |

Задание 2.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Шаг** | **Переменная/Действие** | **Тип операции** | **Логическое выражение** | **Значение** | **Пояснение** |
| 1 | temperature | Присваивание | - | -15 | Текущая температура |
| 2 | if temperature < -20: | Условие | -15 < -20 | False | Проверка условия |
| 3 | print("Включить аварийный обогрев!") | Вывод | - | - | Включение аварийного обогрева |
| 4 | elif temperature < 0: | Условие | -15 < 0 | True | Проверка условия |
| 5 | print("Включить стандартный обогрев") | Вывод | - | - | Включение стандартного обогрева |
| 6 | else: | Альтернатива | - | - | Иначе выполнить |
| 7 | print("Температура в норме") | Вывод | - | Температура в норме | Сообщение о норме температуры |

Задание 3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Итерация** | **Значение параметра sample** | **Проверка условия sample == "лёд":** | **Действие** | **Значение переменной found** |
| 1 | “песок” | False | - | False |
| 2 | “пыль” | False | - | False |
| 3 | “лёд” | True | found = True  Вывод | True |
| 4 | “камень” | False | - | True |