

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

Отчет по лабораторной работе №2
по дисциплине
«Теория автоматов»

Выполнил студент гр. ИВТб-2301-05-00

_____/Макаров С.А./

Преподаватель

_____/Мельцов В.Ю./

Киров 2025

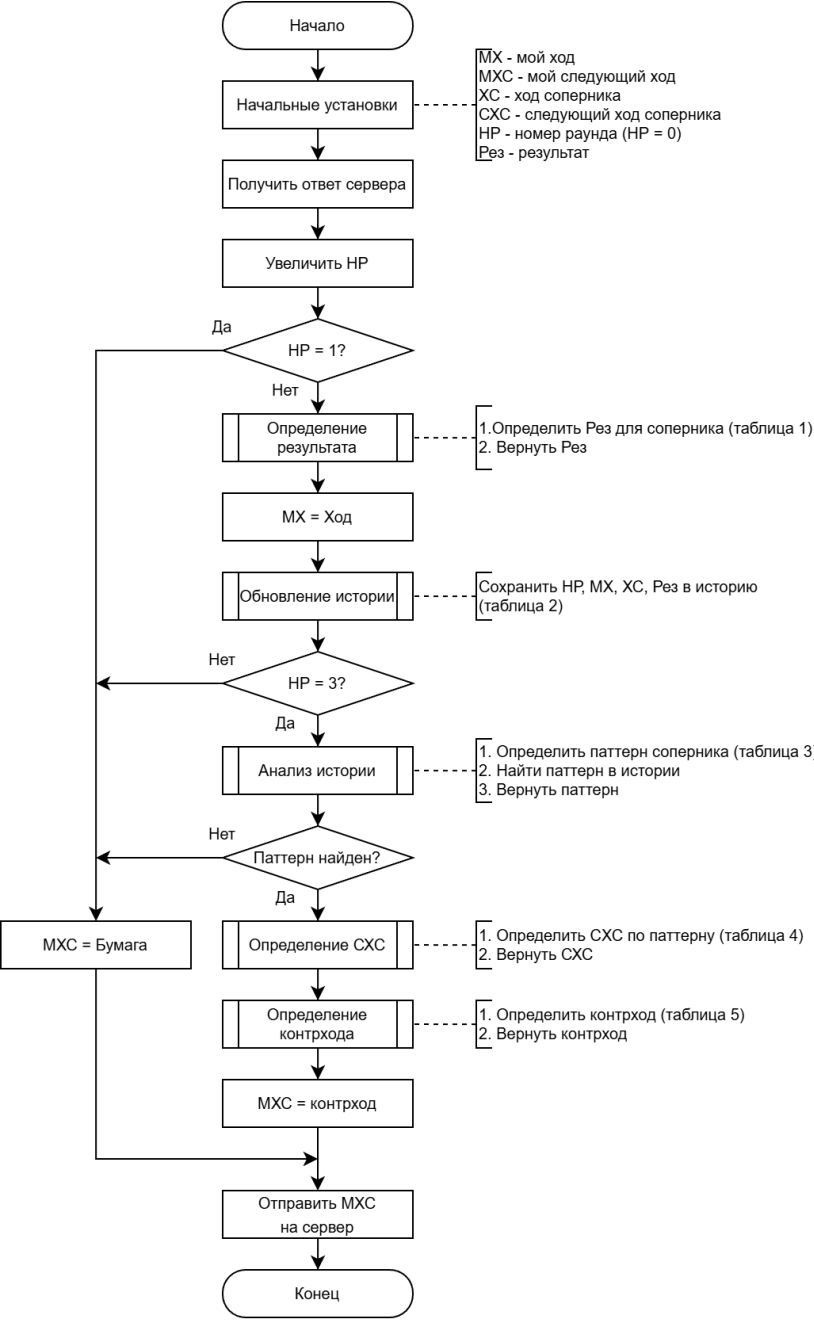
Цель

Получить навыки разработки алгоритма, реализующего автомат, который на основании истории ходов соперников может предсказывать его следующий ход в игре «Камень, Ножницы, Бумага».

Задание

Разработать алгоритм работы бота для игры «Камень, Ножницы, Бумага», разработать программу, принять участие в турнире.

Решение



Макаров Станислав ИВТ6-2301

Происходит анализ двух предыдущих ходов. Если паттерн был найден, то выдается контрход. Если паттерн не найден в качестве хода используется бумага.

Таблица 1 - Определение результата хода соперника

| | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|
| MX | К | К | К | Н | Н | Н | Б | Б | Б |
| XC | К | Н | Б | К | Н | Б | К | Н | Б |
| Рез | н | в | п | п | н | в | в | п | н |

Таблица 2 - История ходов (Рез - результат соперника)

| | | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|----|
| № | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| MX | Б | Б | Б | Б | Б | Б | Н | К | Б | Б |
| XC | К | Н | Б | Н | К | Н | Б | Н | Н | К |
| Рез | п | в | н | в | п | в | п | п | в | п |

Таблица 3 - Анализ истории (Рез - результат соперника)

| | | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|----|
| № | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| MX | Б | Б | Б | Б | Б | Б | Н | К | Б | Б |
| XC | К | Н | Б | Н | К | Н | Б | Н | Н | К |
| Рез | п | в | н | в | п | в | п | п | в | п |

Исходный паттерн ходов соперника

Найденный паттерн ходов соперника

Таблица 4 - Определение CXС (Рез - результат соперника)

| | | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|----|
| № | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| MX | Б | Б | Б | Б | Б | Б | Н | К | Б | Б |
| XC | К | Н | Б | Н | К | Н | Б | Н | Н | К |
| Рез | п | в | н | в | п | в | п | п | в | п |

Исходный паттерн ходов соперника

Найденный паттерн ходов соперника

МХС На основе паттерна соперника

CXC На основе паттерна соперника

Таблица 5 - Контрходы

| | | | |
|----------|---|---|---|
| Ход | К | Н | Б |
| Контрход | Б | К | Н |

Исходный код программы, написанный на языке Python представлен ниже:

```
ROCK = 1 # Камень
PAPER = 2 # Бумага
SCISSORS = 3 # Ножницы
```

```
history = [[], [], []]
```

```
def set_parameters(set_count: int, wins_per_set: int) -> None:
    """
```

```
    Вызывается один раз перед началом игры.
```

```
    Передаёт параметры, с которыми запущен турнир.
```

```
    :param set_count: Максимальное количество сетов в игре
```

```
    :param wins_per_set: Требуемое количество побед в сете
```

```
    """
```

```
    pass
```

```
def on_game_start() -> None:
```

```
    """ Вызывается один раз в начале игры. """
```

```
    pass
```

```
def beat(move: int) -> int:
```

```
    """ Возвращает ход, который бьёт переданный. """
```

```
    return {ROCK: PAPER, PAPER: SCISSORS, SCISSORS: ROCK}[move]
```

```
def choose(previous_opponent_choice: int) -> int:
```

```
    """
```

```
    Функция должна вернуть число от 1 до 3, соответствующее фигуре,
    которую выбрал бот
```

```
    (1 - Камень, 2 - Бумага, 3 - Ножницы).
```

```
    Передаваемый параметр previous_opponent_choice - число от 1 до 3, выбор
    противника на предыдущем ходу.
```

```
    Самый первый раз за игру, при первом вызове choose,
```

```
    previous_opponent_choice равен 0 (т.к. предыдущих ходов ещё не было).
```

```
    :param previous_opponent_choice: Код фигуры, выбранной противником на
    предыдущем ходу
```

```

: return: Код фигуры, которую выбирает бот
"""

if previous_opponent_choice == 0:
    return PAPER

history[0].append(previous_opponent_choice)

if len(history[0]) < 3:
    return PAPER

pattern = tuple(history[0][-2:])

for i in range(len(history[0]) - 2):
    if tuple(history[0][i:i+2]) == pattern:
        return beat(history[0][i + 2])

return PAPER

def on_game_end() -> None:
    """ Вызывается один раз в конце игры. """
    pass

```

Вывод

В ходе выполнения лабораторной работы разработан алгоритм работы бота для игры «Камень, Ножницы, Бумага», а также разработана программа на языке Python, реализующий данный алгоритм.