

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1
дисциплины
«Искусственный интеллект в профессиональной сфере»

Выполнил:
Гуляницкий Александр
3 курс, группа ИТС-б-о-22-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи», очная
форма обучения

(подпись)

Проверил:
Воронкин Р. А., доцент департамента
цифровых, робототехнических систем
и электроники

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

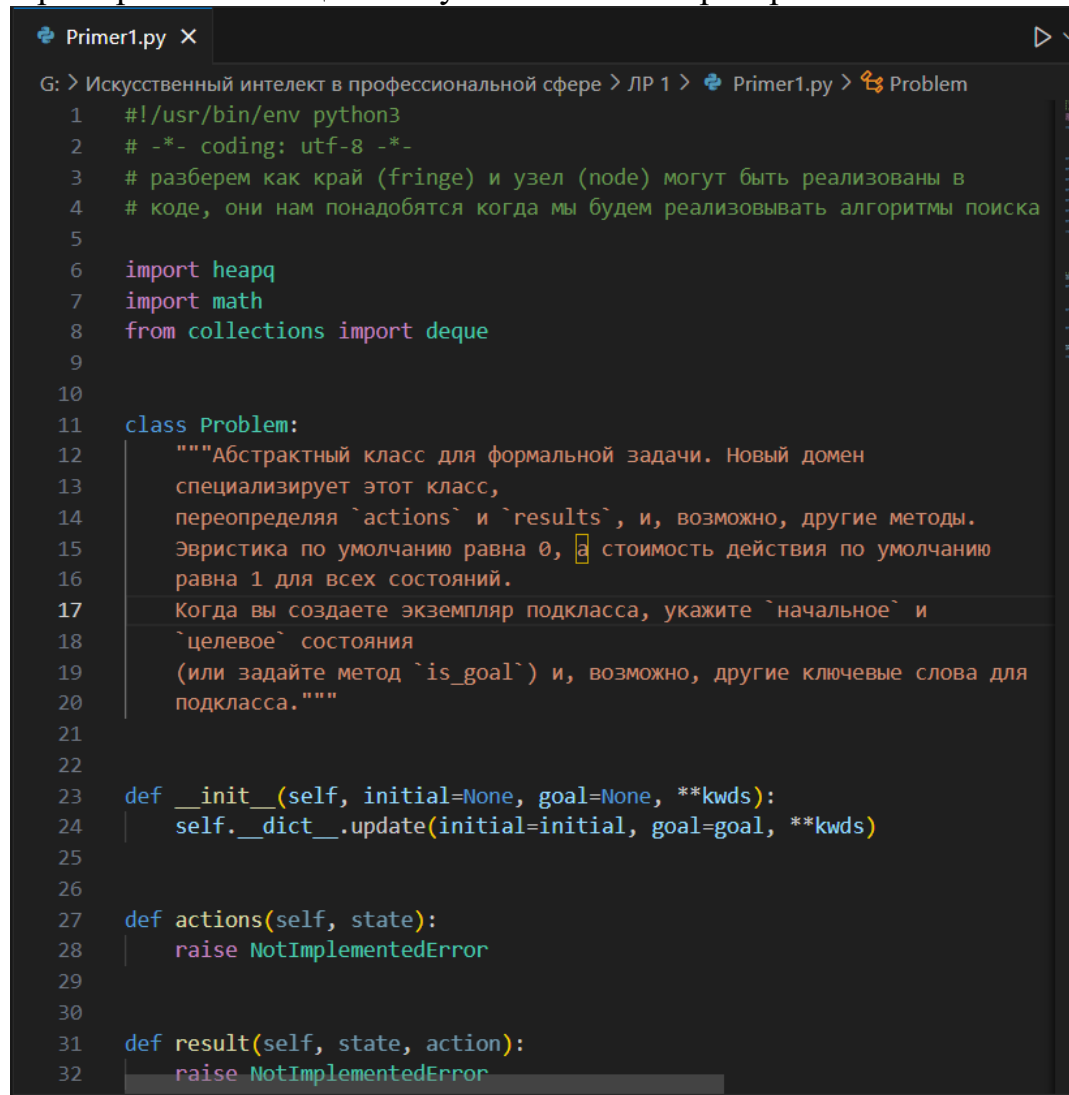
Исследование методов поиска в пространстве состояний.

Цель работы: приобретение навыков по работе с методами поиска в пространстве состояний с помощью языка программирования Python версии 3.x.

Ссылка на репозиторий: <https://github.com/Alexander-its/LR1.git>

Ход работы:

Пример 1. Реализация на Python поиска в пространстве состояний.



```
Primer1.py X
G: > Искусственный интеллект в профессиональной сфере > ЛР 1 > Primer1.py > Problem
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  # разберем как край (fringe) и узел (node) могут быть реализованы в
4  # коде, они нам понадобятся когда мы будем реализовывать алгоритмы поиска
5
6  import heapq
7  import math
8  from collections import deque
9
10
11 class Problem:
12     """Абстрактный класс для формальной задачи. Новый домен
13     специализирует этот класс,
14     переопределяя `actions` и `results`, и, возможно, другие методы.
15     Эвристика по умолчанию равна 0, а стоимость действия по умолчанию
16     равна 1 для всех состояний.
17     Когда вы создаете экземпляр подкласса, укажите `начальное` и
18     `целевое` состояния
19     (или задайте метод `is_goal`) и, возможно, другие ключевые слова для
20     подкласса."""
21
22
23 def __init__(self, initial=None, goal=None, **kwargs):
24     self.__dict__.update(initial=initial, goal=goal, **kwargs)
25
26
27 def actions(self, state):
28     raise NotImplementedError
29
30
31 def result(self, state, action):
32     raise NotImplementedError
```

Рисунок 1 – Код для выполнения примера 1

Задание 1. Решим задачу коммивояжёра методом полного перебора.

```
IDZ.py x
G: > Искусственный интеллект в профессиональной сфере > ЛР 1 > IDZ.py > ...

18 def traveling_salesman(distance_matrix):
29     if current_distance < min_distance:
30         min_distance = current_distance
31         best_route = route
32
33     return best_route, min_distance
34
35
36 # Пример использования
37 if __name__ == "__main__":
38     # Матрица расстояний между городами для 10 узлов
39     distance_matrix = [
40         [0, 634, 246, 420, 98, 181, 853, 462, 457, 199],
41         [634, 0, 719, 545, 732, 815, 1487, 1096, 983, 833],
42         [246, 719, 0, 174, 183, 100, 938, 708, 703, 284],
43         [420, 545, 174, 0, 357, 274, 1112, 882, 877, 458],
44         [98, 732, 183, 357, 0, 83, 755, 560, 555, 101],
45         [181, 815, 100, 274, 83, 0, 838, 643, 638, 184],
46         [853, 1487, 938, 1112, 755, 838, 0, 462, 674, 654],
47         [462, 1096, 708, 882, 560, 643, 462, 0, 212, 661],
48         [457, 983, 703, 877, 555, 638, 674, 212, 0, 656],
49         [199, 833, 284, 458, 101, 184, 654, 661, 656, 0],
50     ]
51
52     best_route, min_distance = traveling_salesman(distance_matrix)
53     print(f"Лучший маршрут: {best_route}")
54     print(f"Минимальное расстояние: {min_distance}")
```

Рисунок 2 – Код для выполнения задания 1

Блок схема:

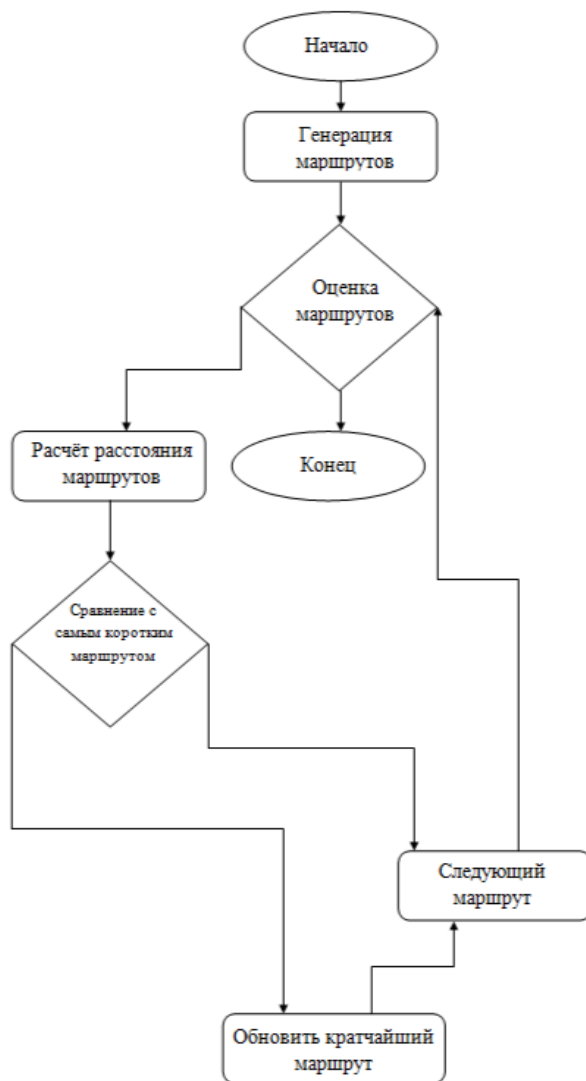


Рисунок 5 – Блок-схема задания №1

Вывод: в ходе выполнения лабораторной работы приобрели навыки по работе с методами поиска в пространстве состояний с помощью языка программирования Python версии 3.x.