# 

# **Отчет по**

# **Лабораторной работе №1.**

# **По предмету: Artificial Intelligence**

# 

# 

# 

# **Подготовил студент группы IA2303: Gutu Nicolae**

# **Проверяющий преподаватель: V. Trebiș**

# 

# 

# **Сhișinau, 2025**

**Задание:**

Написать программу реализующую ĸонечный автомат, ĸоторый распознает слова, имеющие следующий общий вид: **(abcd)ⁿ (ef)ᵐ, n ≥ 1, m ≥ 1**

**Теоретическая часть:**

**1) Алфавит: {a, b, c, d, e, f}**

**2)Таблица переходов**

| **Состояние** | **a** | **b** | **c** | **d** | **e** | **f** |
| --- | --- | --- | --- | --- | --- | --- |
| **q0** | **q1** | **⊗** | **⊗** | **⊗** | **⊗** | **⊗** |
| **q1** | **⊗** | **q2** | **⊗** | **⊗** | **⊗** | **⊗** |
| **q2** | **⊗** | **⊗** | **q3** | **⊗** | **⊗** | **⊗** |
| **q3** | **⊗** | **⊗** | **⊗** | **q4** | **⊗** | **⊗** |
| **q4** | **q1** | **⊗** | **⊗** | **⊗** | **q5** | **⊗** |
| **q5** | **⊗** | **⊗** | **⊗** | **⊗** | **⊗** | **q6** |
| **q6** | **⊗** | **⊗** | **⊗** | **⊗** | **q5** | **⊗** |

**Start - q0**

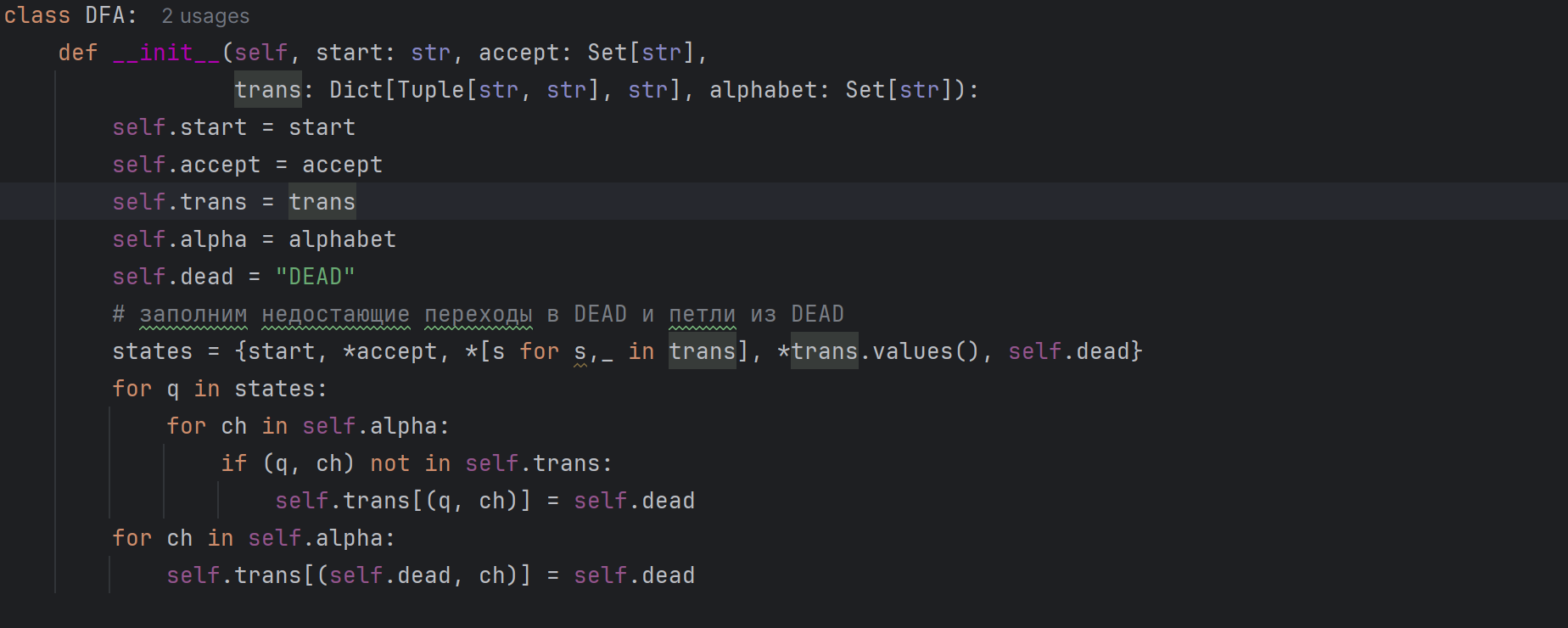
**F (принимающее состояние) - только q6**

**3) Для конкретно моего задания имеются 2 фазы:**

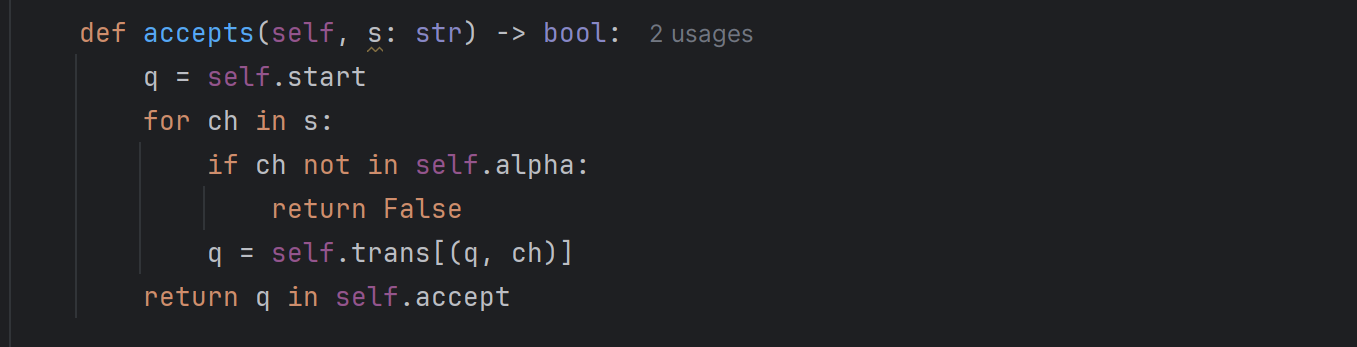
* **Фаза abcd:**
  + **q0 — старт, ждём a**
  + **q1 — после a**
  + **q2 — после ab**
  + **q3 — после abc**
  + **q4 — завершили один abcd (граница повторов)**
* **Фаза ef:**
  + **q5 — после e**
  + **q6 — завершили один ef (граница повторов) — оно же принимающее состояние**

**И ещё мёртвое состояние DEAD (иногда называют «поглощающим», «sink»): если что-то пошло не по плану, проваливаемся туда и уже оттуда не выходим.**

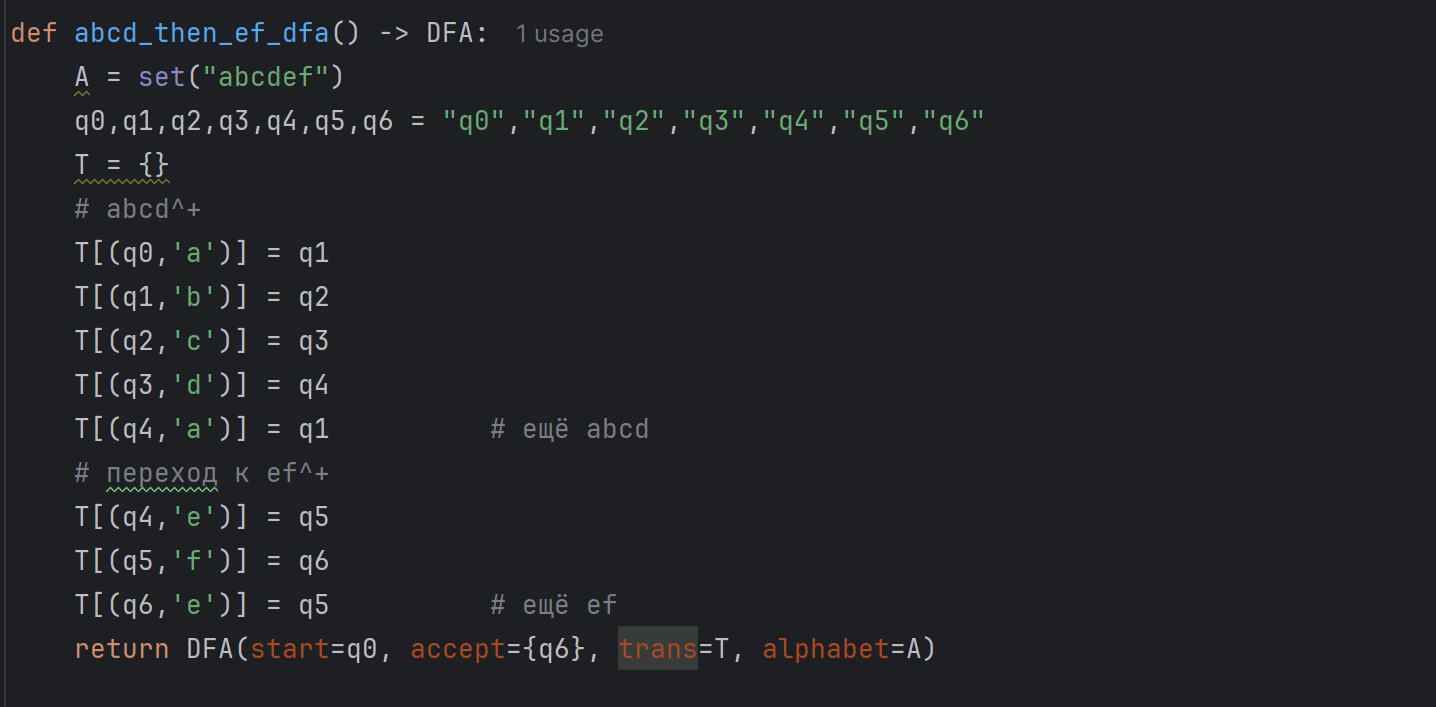
**4) Python - code:**

****

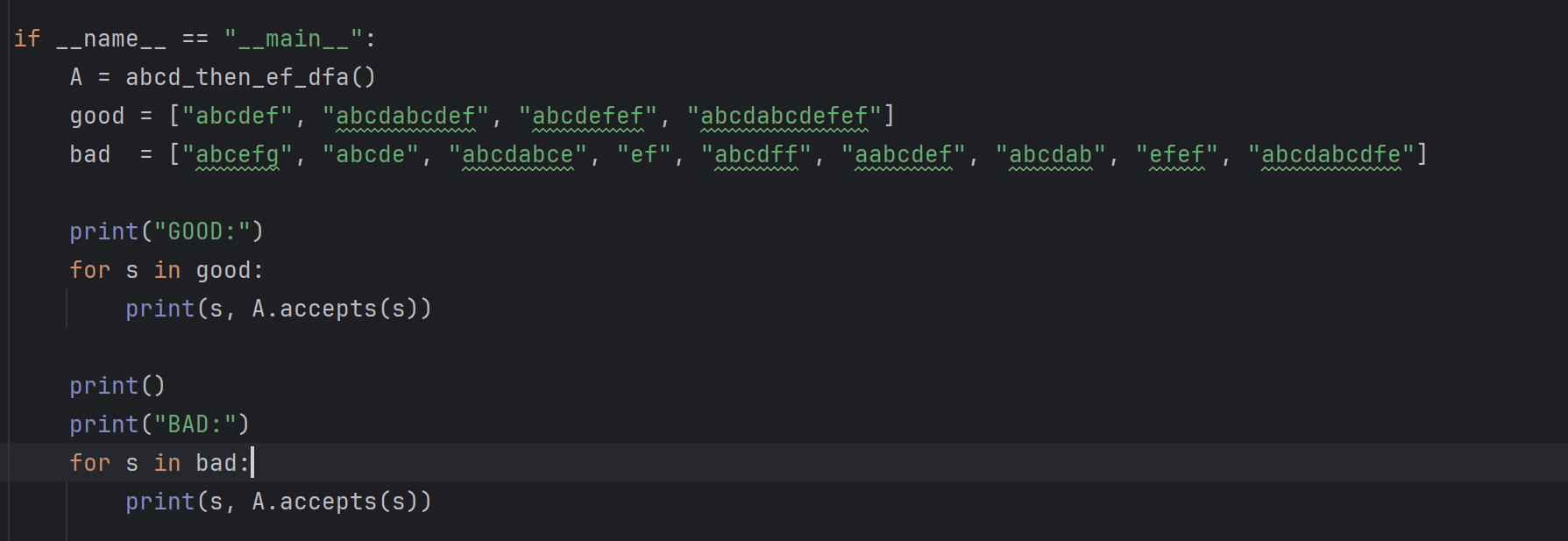
* **DFA -** симулятор детерминированного конечного автомата
* Поля:Start, accept, trans, alphabet, dead-end - стандартное множество полей DFA
* В циклах собираются все встречающиеся состояния, а DEAD-end’ы указывают на самих себя, чтоб туда можно было попасть, но нельзя было выйти



* Каждый символ переводит автомат в следующее состояние (по таблице trans).
* Если встретился символ вне алфавита — сразу False.
* Если после прохода всех состояний мы остались в множестве ACCEPT, то значит проверка пройдена и введенная строка соответствует условию



* Конструктор автомата abcd\_then\_ef\_dfa() (Он задаёт сам язык)
* Логика:
* q0→q1→q2→q3→q4 читает один блок abcd.
* Из q4 два пути:
  + a → снова в q1 (начать следующий abcd);
  + e → перейти к фазе ef
* Принимающим делаем **q6** — это гарантирует, что последним прочитан символ f, т.е. был пройден минимум один **ef**.



* На вход подаются для проверки 2 списка строк.

6) Результат работы программы:



Если просмотреть результат - видно, что все было обработано верно.

Таким образом,был построен детерминированный конечный автомат (DFA) со стадиями для блоков abcd и ef, корректность работы которого была подтверждена тестами.