**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

Факультет инфокоммуникационных сетей и систем (ИКСС)

Лабораторная работа №3

по курсу

«Теория автоматов и формальных языков»

Группа: ИКПИ-14

Выполнил студент: Хохлов Т. В.

Принял преподаватель: Леонова М.Д.

Место для подписи \_\_\_\_\_\_\_\_\_\_\_\_\_

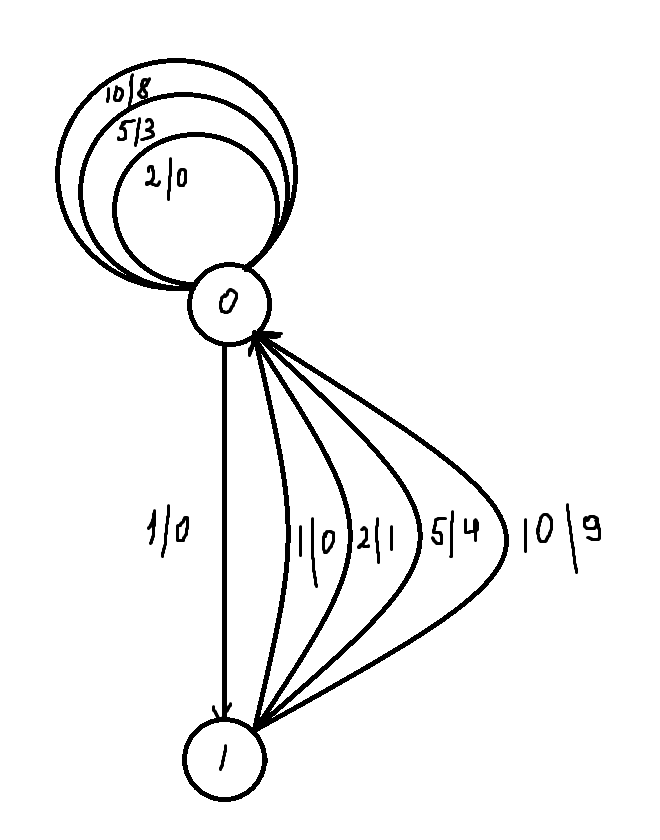
Санкт-Петербург

2023 г.

**Задание:**

Вариант 1. Вендинговый автомат принимает монеты 1, 2, 5, 10 рублей. Товар стоит 2 рубля. Реализовать через автомат Мили.

**Диаграмма МИЛИ:**

****

**Программа:**

class VendingMachine:

def \_\_init\_\_(self):

self.state = 0

def insert\_coin(self, coin):

if self.state == 0:

if coin == 1:

self.state = 1

return -1

elif coin == 2:

return 0

elif coin == 5:

return 3

elif coin == 10:

return 8

elif self.state == 1:

if coin == 1:

self.state = 0

return 0

elif coin == 2:

self.state = 0

return 1

elif coin == 5:

self.state = 0

return 4

elif coin == 10:

self.state = 0

return 9

def output\_state(self, change):

if self.state == 0:

if change >= 0:

print(f"Сумма: 0 рублей. Выдаем товар. Сдача: {change}")

elif change < 0:

print("Сумма: 0 рублей. AFK.")

elif self.state == 1:

print("Сумма: 1 рубль. AFK.")

if \_\_name\_\_ == "\_\_main\_\_":

vm = VendingMachine()

while True:

try:

coin = int(input("Вставьте монету (1, 2, 5, 10): "))

if coin not in [1, 2, 5, 10]:

print("Неверная монета. Попробуйте снова.")

continue

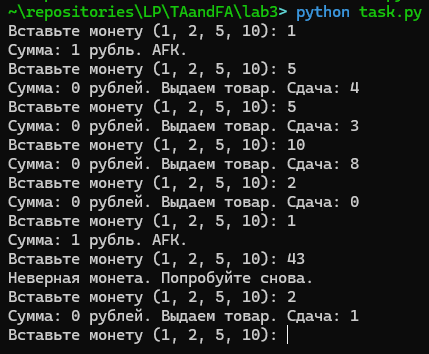
change = vm.insert\_coin(coin)

vm.output\_state(change)

except ValueError:

print("Пожалуйста, введите числовое значение.")

**Пример работы программы:**



**Вывод:**

Я научился проектировать и реализовывать автоматы Мура, а также создавать схемы состояний для моделирования работы устройств, таких как вендинговые автоматы. В процессе работы я приобрел навыки программирования на Python и научился использовать диаграммы состояний для визуализации переходов между состояниями автомата.