



**«Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)»**

---

**ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

**КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)**

**НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА**

**О Т Ч Е Т**

**по домашней работе**

**Название:** Автомат – “замок”.

**Дисциплина:** Научная организация инженерного труда.

Студент

ИУ6-44Б  
(Группа)



2022/05/17

(Подпись, дата)

Д.Р. Подручный

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Ю.И. Бауман

(И.О. Фамилия)

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
ОСНОВНАЯ ЧАСТЬ.....	4
Спецификация автомата .....	4
Полученный конечный автомат.....	5
Реализация конечного автомата .....	7
Тестирование программы.....	8
ЗАКЛЮЧЕНИЕ .....	12
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	13
ПРИЛОЖЕНИЕ А .....	14

## **ВВЕДЕНИЕ**

В данной работе выполнена реализация цифрового автомата – “замок”. Конечный цифровой автомат эмитирует работу кодового замка сейфа.

Существуют 2 способа реализации автомата, а именно программный и аппаратный. Программная реализация выполняется на любом языке высокого уровня. Аппаратная реализация – предусматривает построение схемы устройства памяти для запоминания текущего состояния автомата, в роли которых обычно используют триггеры.

В настоящей работе использован программный способ реализации цифрового автомата, так как этот способ подразумевает вариативность реализации, возможность отладки и тестирования в процессе разработки программы. В программах (в отличие от аппаратной реализации цифровых автоматов) можно расширять функциональность по мере изменения целей, под которые она разрабатывается.

Задание (10 вариант): разработать и реализовать конечный автомат – “замок”. Программа эмитирует работу кодового замка сейфа.

Цель работы – закрепить навыки реализации конечных цифровых автоматов. Для реализации поставленной цели необходимо выполнить следующие задачи.

Задачи:

- Изучить задание в соответствие со своим вариантом;
- Описать автомат, соответствующий условию задачи;
- Изучить способы реализации цифровых автоматов;
- Выбрать один из способов реализации автоматов;
- Реализовать описанный цифровой автомат.

## ОСНОВНАЯ ЧАСТЬ

Необходимо реализовать конечный цифровой автомат “замок”. Кодовый замок подразумевает собой возможность открытия сейфа. Чтобы реализовать данный цифровой автомат, требуется реализовать возможность установки пароля, ввода пароля, открытия и закрытия сейфа. Открытие сейфа происходит при вводе пароля идентичного установленному (пользователь не может закрыть сейф без установленного пароля). При вводе некорректного пароля, количество попыток для открытия уменьшается. Всего имеется три попытки, чтобы ввести корректный пароль, иначе сейф будет заблокирован навсегда. Пароль состоит из пяти цифр, которые вводятся при нажатии кнопок.

На основе проведенного анализа задания составим конечный цифровой автомат.

### Спецификация автомата

#### 1. Состояния автомата.

- 1.1.  $q_0$  – замок открыт, пароль не установлен;
- 1.2.  $q_1$  – замок закрыт, пароль установлен;
- 1.3.  $q_2$  – замок открыт, пароль установлен;
- 1.4.  $q_3$  – замок закрыт, осталось 2 попытки для ввода корректного пароля;
- 1.5.  $q_4$  – замок закрыт, осталась 1 попытка для ввода корректного пароля;
- 1.6.  $q_5$  – замок закрыт, сейф заблокирован навсегда.

#### 2. Входные сигналы.

- 2.1.  $a$  – корректный пароль для установки;
- 2.2.  $b$  – некорректный пароль для установки;
- 2.3.  $c$  – корректный введенный пароль для открытия сейфа;
- 2.4.  $d$  – некорректный введенный пароль для открытия сейфа;
- 2.5.  $e$  – закрытие сейфа.

### 3. Выходные сигналы автомата.

- 3.1. 1 – пароль не установлен;
- 3.2. 2 – сейф открыт;
- 3.3. 3 – сейф закрыт, осталось 3 попытки ввода;
- 3.4. 4 – сейф закрыт, осталось 2 попытки ввода;
- 3.5. 5 – сейф закрыт, осталась 1 попытка ввода;
- 3.6. 6 – сейф заблокирован.

### Полученный конечный автомат

Составим таблицу, описывающую конечный автомат, на основе текста задания по результатам анализа (таблица 1).

Таблица 1 – описание конечного автомата

Состояние	$\delta$					$\lambda$				
	a	b	c	d	e	a	b	c	d	e
<b>q<sub>0</sub></b>	q <sub>1</sub>	q <sub>0</sub>	-	-	-	3	1	-	-	-
<b>q<sub>1</sub></b>	-	-	q <sub>2</sub>	q <sub>3</sub>	-	-	-	2	4	3
<b>q<sub>2</sub></b>	q <sub>1</sub>	q <sub>2</sub>	-	-	q <sub>1</sub>	3	2	-	-	3
<b>q<sub>3</sub></b>	-	-	q <sub>2</sub>	q <sub>4</sub>	-	-	-	2	5	-
<b>q<sub>4</sub></b>	-	-	q <sub>2</sub>	q <sub>5</sub>	-	-	-	2	6	-
<b>q<sub>5</sub></b>	-	-	-	-	-	-	-	-	-	-

Теперь представим описанный конечный автомат в виде графа переходов (рисунок 1).

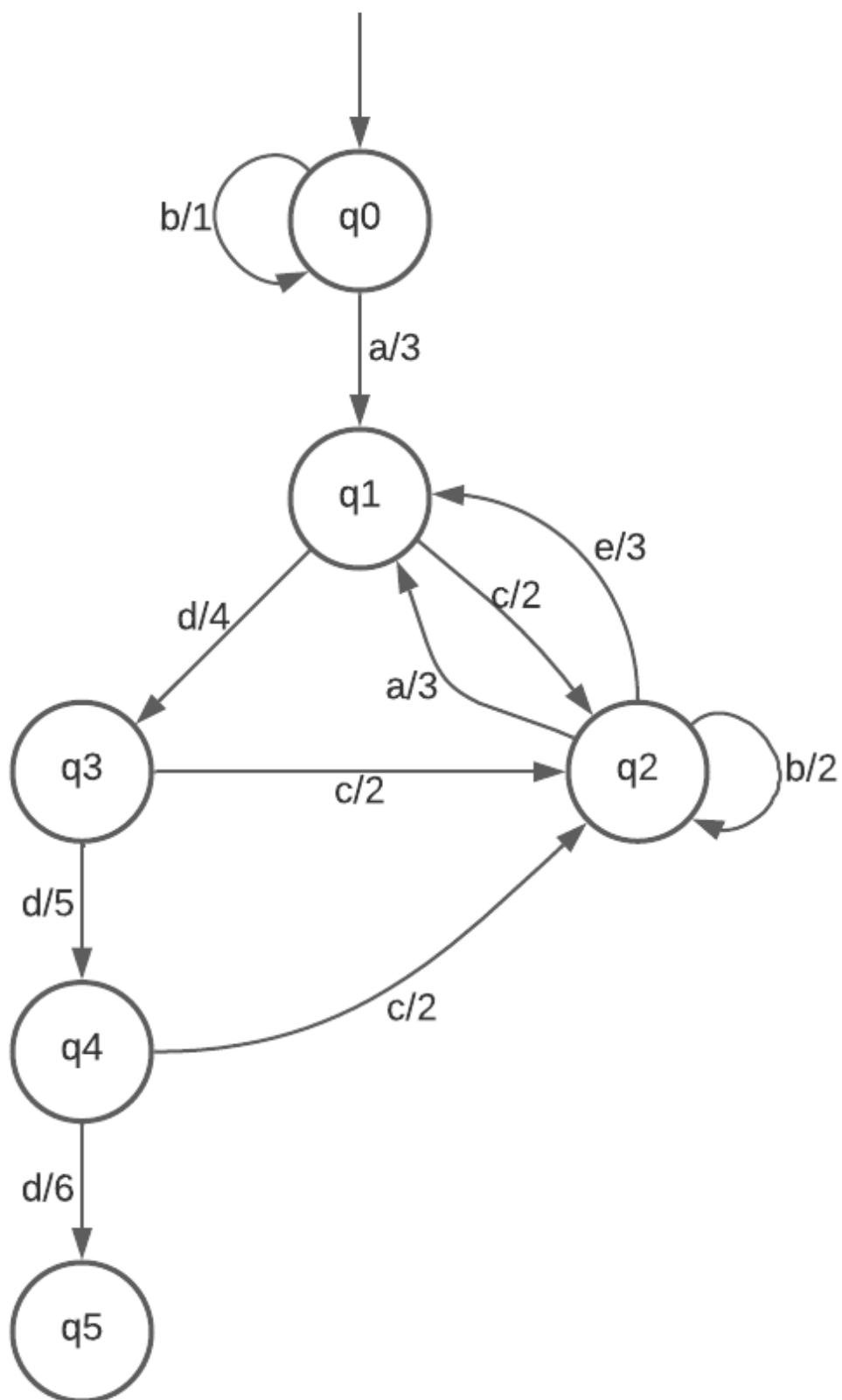


Рисунок 1 – граф переходов конечного автомата

## Реализация цифрового автомата “Замок”

Для реализации описанного конечного цифрового автомата была написана программа на языке программирования Python с использованием библиотеки PyQt 5 [1]. Также использовалась среда Qt Designer [3] для разработки интерфейса программы.

Ниже представлен интерфейс программы. Подробный код всего приложения содержится в Приложении А.

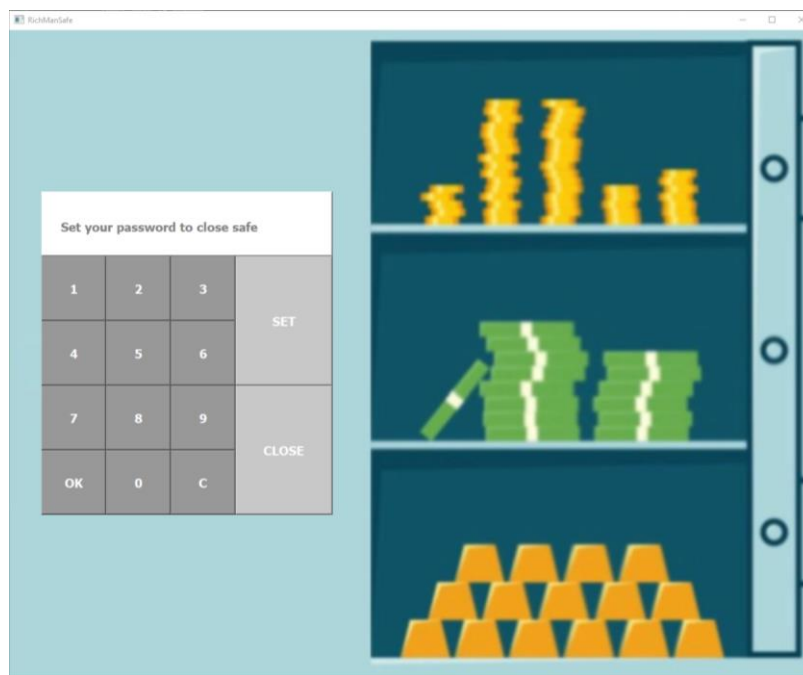


Рисунок 2 – сейф открыт

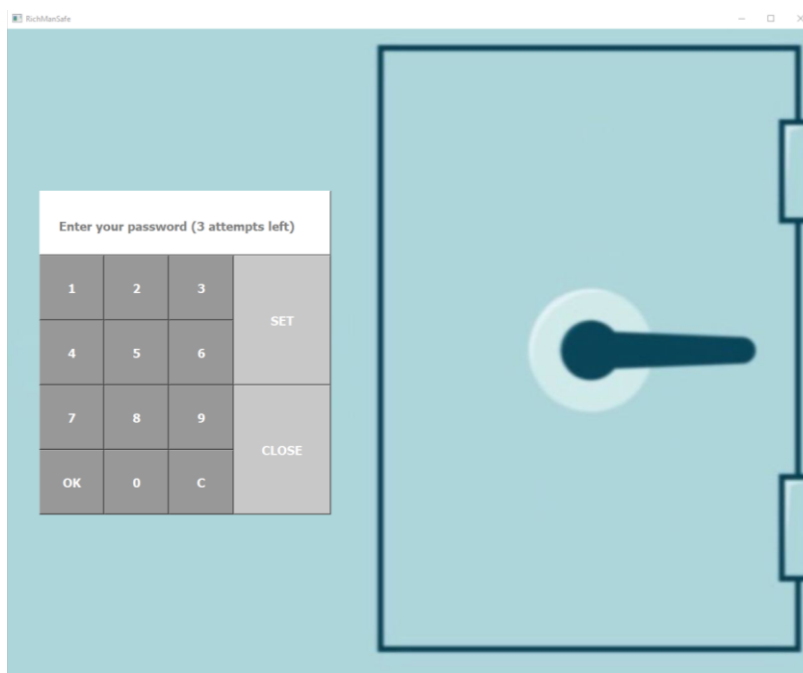


Рисунок 3 – сейф закрыт

## Тестирование программы

Протестируем написанную программу (рисунки 4 - 20).

Для успешного тестирования необходимо проверить все переходы конечного автомата, изображенные на графе переходов.

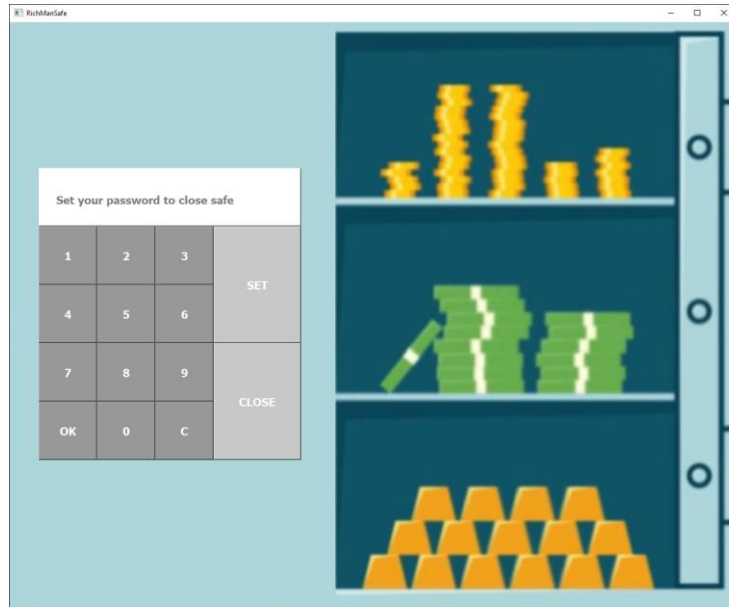


Рисунок 4 – начальное состояние. Сейф открыт, пароль не установлен (состояние  $q_0$  конечного автомата)

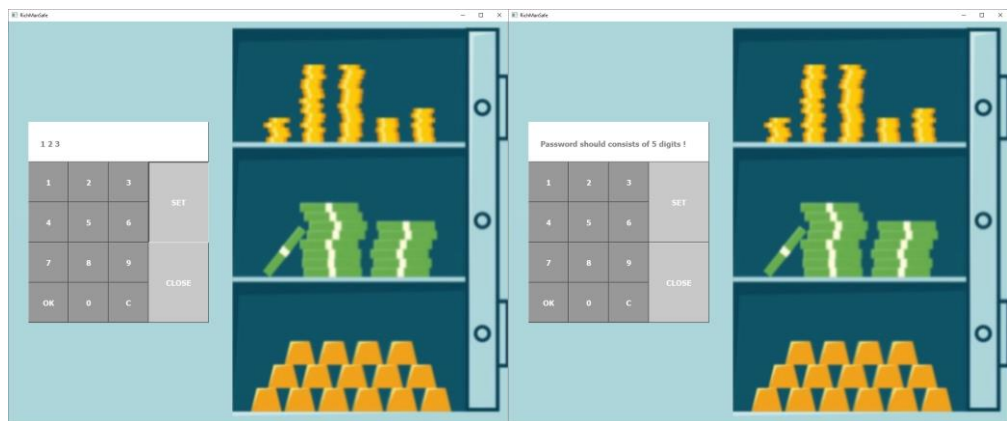


Рисунок 5 - 6 – попытка установить пароль меньше 5 цифр (состояние  $q_0$  конечного автомата не поменялось)





Рисунок 7 - 8 – попытка установить пустой пароль (состояние  $q_0$  конечного автомата не поменялось)

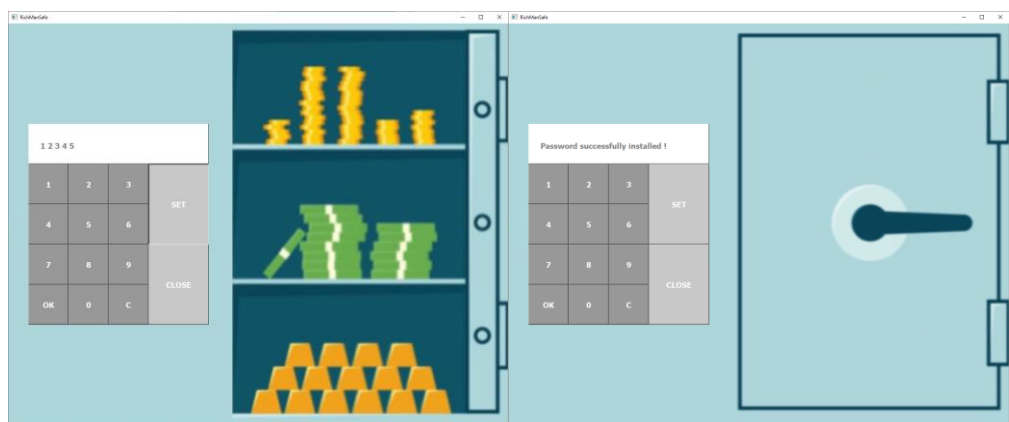


Рисунок 9 - 10 – введен корректный пароль для установки. Сейф закрылся, пароль успешно установлен (состояние  $q_1$  конечного автомата)

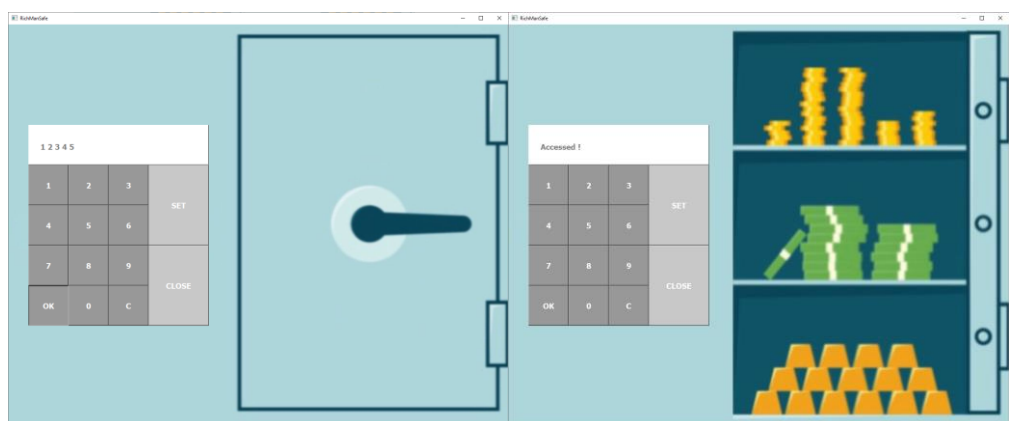


Рисунок 11 - 12 – ввод правильного пароля для открытия сейфа (из состояния  $q_1$  перешли в состояние  $q_2$ )

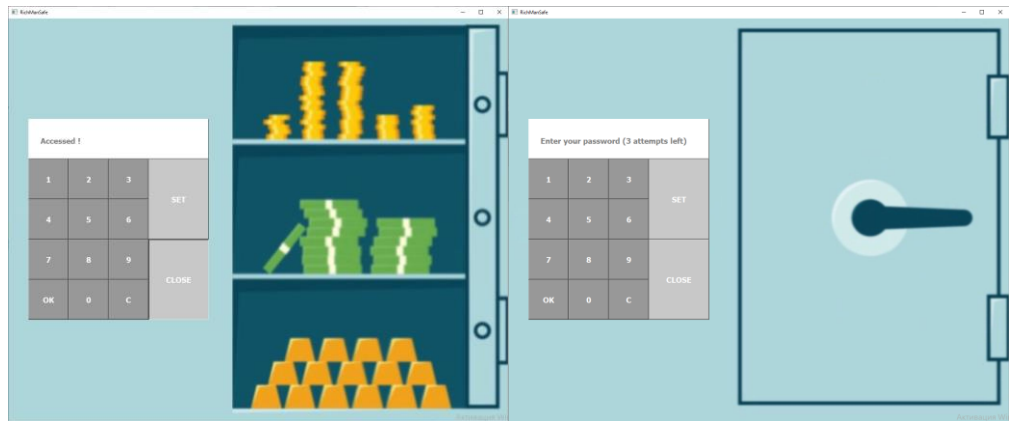


Рисунок 13 - 14 – закрытие сейфа (из состояния  $q_2$  перешли в состояние  $q_1$ )

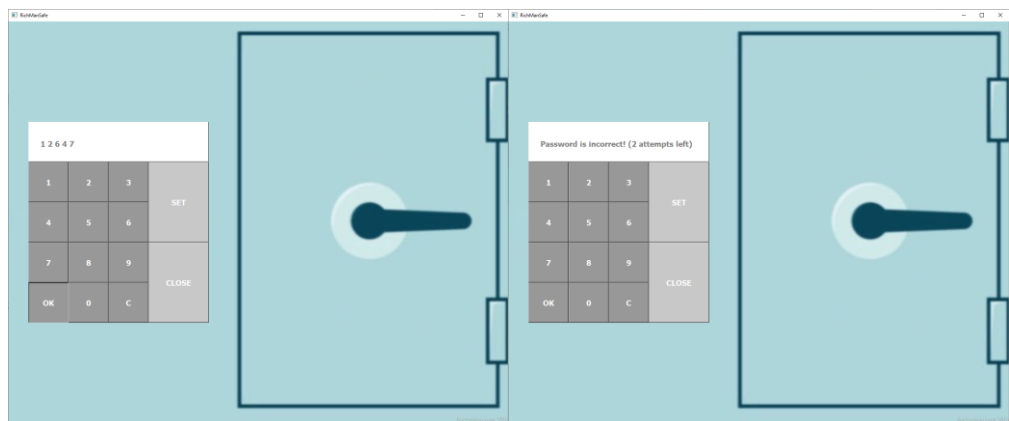


Рисунок 15 - 16 – введен неправильный пароль (перешли в состояние  $q_3$ )

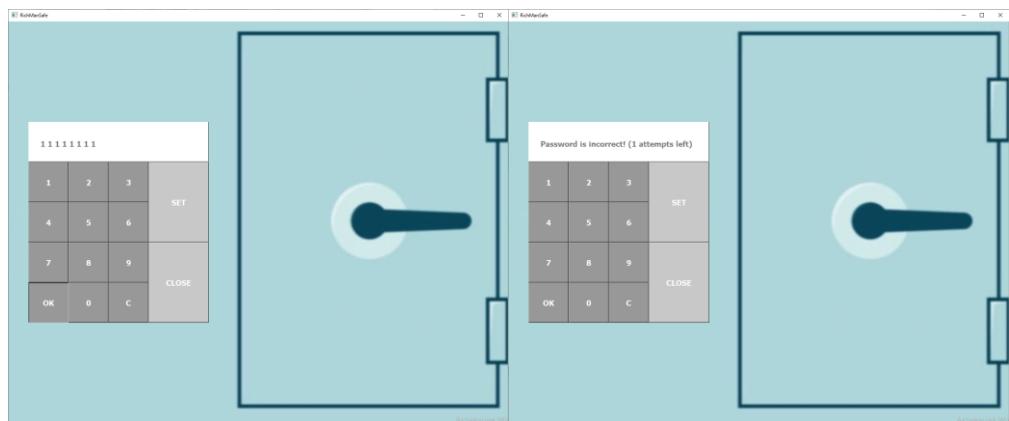


Рисунок 17 - 18 – второй раз введен неправильный пароль (перешли в состояние  $q_4$ )

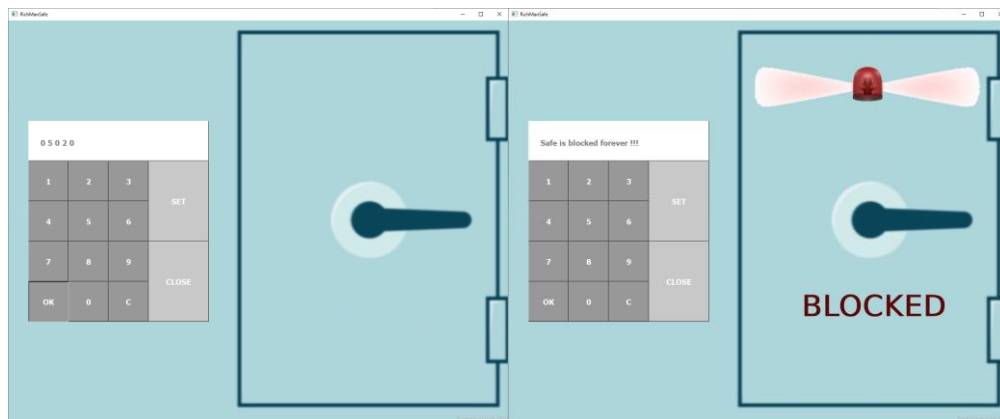


Рисунок 19 - 20 – последняя попытка ввода была неудачной, сейф заблокирован (перешли в состояние  $q_5$ )

Были протестированы практически все переходы состояний конечного автомата, исходя из построенного ранее графа переходов. Не был протестирован ввод некорректного пароля для установки в состоянии  $q_2$ , так как результат будет идентичным результату, показанному на рисунках 5 – 8. Также не был протестирован ввод правильного пароля из состояний  $q_3$  и  $q_4$ , так как результат будет таким же, как на рисунках 11 – 12.

Тестирование показало, что программа работает корректно и соответствует спецификации автомата.

## **ЗАКЛЮЧЕНИЕ**

При выполнении домашнего задания был изучен способ проектирования конечных цифровых автоматов. Изучен программный способ реализации автоматов. Закреплены навыки подготовки и оформления отчета в соответствии с ГОСТ 7.32 [2].

В ходе выполнения домашнего задания был спроектирован и реализован конечный цифровой автомат “замок”, целью которого является открытие сейфа с помощью пароля, состоящего из 5 цифр. Программная реализация данного конечного цифрового автомата была реализована на языке программирования высокого уровня Python с использованием библиотеки PyQt 5. Для создания интерфейса программы была использована среда разработки интерфейсов Qt Designer.

Проведено тестирование, демонстрирующее корректность работы программы.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. PyQt5 (tutorialspoint.com). – [130 л.]. – URL: [https://www.tutorialspoint.com/pyqt5/pyqt5\\_tutorial.pdf](https://www.tutorialspoint.com/pyqt5/pyqt5_tutorial.pdf) (дата обращения: 14.05.2022). – Текст: электронный.
2. ГОСТ 7.32-2017. СИБИД. Отчет о научно-исследовательской работе. Структура и правила оформления = System of standards on information, librarianship and publishing. The research report. Structure and rules of presentation : Национальный стандарт РФ : Введ. 01.07.2018. - М. : Стандартиформ, 2017. - [32 л.]. - URL: <https://docs.cntd.ru/document/1200157208> (дата обращения: 04.05.2022). - Текст: электронный.
3. Qt Designer Manual. – URL: <https://doc.qt.io/qt-5/qtdesigner-manual.html> (дата обращения: 14.05.2022). – Текст: электронный.

## ПРИЛОЖЕНИЕ А

К домашнему заданию прилагается листинг кода приложения.

### Файл **main.py**

```
from PyQt5 import uic
from PyQt5.QtWidgets import QApplication

Form, Window = uic.loadUiType('safe.ui')

app = QApplication([])
window = Window()
form = Form()
form.setupUi(window)
window.show()

global input_code
input_code = ""
global installed_code
installed_code = ""

def write_number(number):
    global input_code
    if not form.output.text().replace(' ', '').isdigit():
        form.output.setText(number)
        input_code += number
    else:
        form.output.setText(form.output.text() + ' ' + number)
        input_code += number

def reset_password():
    global input_code
    form.output.setText("")
    input_code = ""

def set_password():
    global installed_code
    global input_code
    if input_code == "":
        form.output.setText('Fill the password !')
    elif len(input_code) != 5:
        form.output.setText('Password should consists of 5 digits !')
    input_code = ""
```

```

else:
    installed_code = input_code
    form.output.setText('Password successfully installed !')
    input_code = ""
    form.background_pic.setStyleSheet('QPushButton {background: url('
                                      'C:/Users/79680/PycharmProjects/safe/Pictures/close.jpg);}')
    form.btn_set.setEnabled(False)
    form.btn_close.setEnabled(False)
    form.btn_ok.setEnabled(True)

```

```

def close_safe():
    form.background_pic.setStyleSheet('QPushButton {background: url('
                                      'C:/Users/79680/PycharmProjects/safe/Pictures/close.jpg);}')
    form.output.setText('Enter your password (3 attempts left)')
    form.btn_set.setEnabled(False)
    form.btn_close.setEnabled(False)
    form.btn_ok.setEnabled(True)

```

```

global count
count = 3

```

```

def check_password():
    global installed_code
    global input_code
    global count
    if installed_code == input_code:
        form.output.setText('Accessed !')
        input_code = ""
        form.background_pic.setStyleSheet('QPushButton {background: url('
                                          'C:/Users/79680/PycharmProjects/safe/Pictures/open.jpg);}')
        form.btn_set.setEnabled(True)
        form.btn_close.setEnabled(True)
        form.btn_ok.setEnabled(False)
        count = 3

```

```

else:
    form.output.setText(f'Password is incorrect! ({count - 1} attempts left)')
    input_code = ""
    count -= 1
    if count == 0:
        form.output.setText('Safe is blocked forever !!!')
        form.background_pic.setStyleSheet('QPushButton {background: url('

```

```

'C:/Users/79680/PycharmProjects/safe/Pictures/locked2.jpg);}')
    form.btn_set.setEnabled(False)
    form.btn_close.setEnabled(False)
    form.btn_ok.setEnabled(False)
    form.btn_reset.setEnabled(False)
    form.btn1.setEnabled(False)
    form.btn2.setEnabled(False)
    form.btn3.setEnabled(False)
    form.btn4.setEnabled(False)
    form.btn5.setEnabled(False)
    form.btn6.setEnabled(False)
    form.btn7.setEnabled(False)
    form.btn8.setEnabled(False)
    form.btn9.setEnabled(False)
    form.btn_zero.setEnabled(False)


form.btn1.clicked.connect(lambda: write_number(form.btn1.text()))
form.btn2.clicked.connect(lambda: write_number(form.btn2.text()))
form.btn3.clicked.connect(lambda: write_number(form.btn3.text()))
form.btn4.clicked.connect(lambda: write_number(form.btn4.text()))
form.btn5.clicked.connect(lambda: write_number(form.btn5.text()))
form.btn6.clicked.connect(lambda: write_number(form.btn6.text()))
form.btn7.clicked.connect(lambda: write_number(form.btn7.text()))
form.btn8.clicked.connect(lambda: write_number(form.btn8.text()))
form.btn9.clicked.connect(lambda: write_number(form.btn9.text()))
form.btn_zero.clicked.connect(lambda: write_number(form.btn_zero.text()))


form.btn_reset.clicked.connect(reset_password)
form.btn_set.clicked.connect(set_password)
form.btn_close.clicked.connect(close_safe)
form.btn_ok.clicked.connect(check_password)

if __name__ == "__main__":
    app.exec_()

```