

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В. И. Ульянова (ЛЕНИНА)
Кафедра информационной безопасности

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Модели безопасности компьютерных систем»
Тема: Модернизация матрицы доступа

Студентки гр. 1361

Галунина Е.С.

Токарева У.В.

Преподаватели

Шкляр Е.В.

Шульженко А.Д.

Санкт-Петербург

2024

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Студентки: Галунина Е.С., Токарева У.В.

Группа 1361

Тема: Информационный поток по памяти

Задача:

Доработать программу администратора из лабораторной работы №2, чтобы изменять матрицу доступа посредством команд `grant`, `create`, `remove`.

Технические требования:

- 1) Команда `grant` выдает заданному набору субъектов прав доступа на заданный набор объектов.
- 2) Команда `create` создает новый субъект с доступом к заданному набору объектов (набор может быть пустым). Если субъект уже существует, `create` работает как `grant`. Если некоторых объектов из заданного набора не существует, их нужно автоматически создать.
- 3) Команда `remove` удаляет у заданного набора субъектов права на заданный набор объектов.
- 4) Предусмотрите режимы `grant_all`, чтобы выдать или забрать у заданного набора субъектов права доступа на все существующие объекты.
- 5) В коде программы `grant`, `create` и `remove` должны быть реализованы как отдельные функции и принимать соответствующие аргументы.
- 6) Предусмотрите обработку ошибок.

Старые функции из лабораторной работы №2 должны работать:

- 1) Ввод имен субъектов и объектов. Имена регистронезависимы, то есть `G` и `g` – разные объекты. Предусмотрите обработку ошибок при вводе.
- 2) Сохранение и загрузка матрицы доступа из файла.
- 3) Отображение матрицы доступа в окне программы и возможность ее интерактивного изменения без перезагрузки программы.

Для проверки используем программу пользователя из лабораторной работы №2.

Важно: обе программы остаются оконными с пользовательским интерфейсом. Язык программирования любой. Все необходимые для работы текстовые поля должны быть подписаны. Консольные приложения в этой работе не применяются.

Дата сдачи отчета: 04.04.2024

Дата защиты отчета: 04.04.2024

Студентки

Галунина Е.С.

Токарева У.В.

Преподаватели

Шкляр Е.В.

Шульженко А.Д.

ТЕОРЕТИЧЕСКИЕ ПОЛОЖЕНИЯ

Матрица доступа (МД) – готовая модель, позволяющая регламентировать доступ к информационным ресурсам компании, на основании которой можно оценить состояние и структуру защиты данных в информационных системах. В матрице четко устанавливаются права для каждого субъекта по отношению ко всем объектам информации.

Визуально это можно представить в качестве некоего массива данных со множеством ячеек, которые формируются пересечением строки, указывающей на субъект и столбика, указывающего на объект. Получается, что при таком подходе к управлению доступом ячейка содержит определенную запись, характерную для пары субъект-объект и указывает на режим доступа, разрешенный или запрещенный, или его характеристику для каждого конкретного случая. В матрице доступа к информационным ресурсам столбец отождествляется с перечнем контроля доступа, строка выполняет роль профиля доступа присущего объекту.

ХОД РАБОТЫ

1. К программе администратора, которая была создана в прошлой лабораторной работе, были добавлены следующие функции: создает новый субъект с доступом к заданному набору объектов (*create access right*), выдать или забрать у заданного набора субъектов права доступа на все существующие объекты (*grant access right* и *remove all access right*, соответственно), а также переделана функция *grant access right*, которая теперь может выдать заданному набору субъектов права доступа на заданный набор объектов, и *remove*, которая удаляет у заданного набора субъектов права на заданный набор объектов. Обновленный интерфейс программы представлен на рисунке 1.

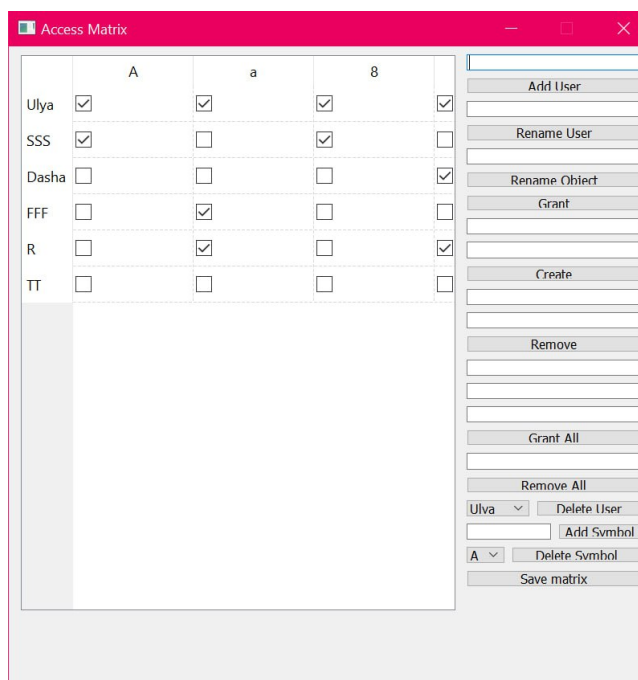


Рисунок 1 – Интерфейс программы администратора

1.1. Далее рассмотрим каждую из новых функций в отдельности. Начнем с функции *grant access right* позволяет теперь выдавать заданному набору субъектов права доступа на заданный набор объектов. Для этого достаточно ввести нужные нам субъекты в поле разделив их запятой и аналогично сделать со списком объектов, после чего нужно нажать кнопку *Grant*. Результат представлен на рисунке 2.

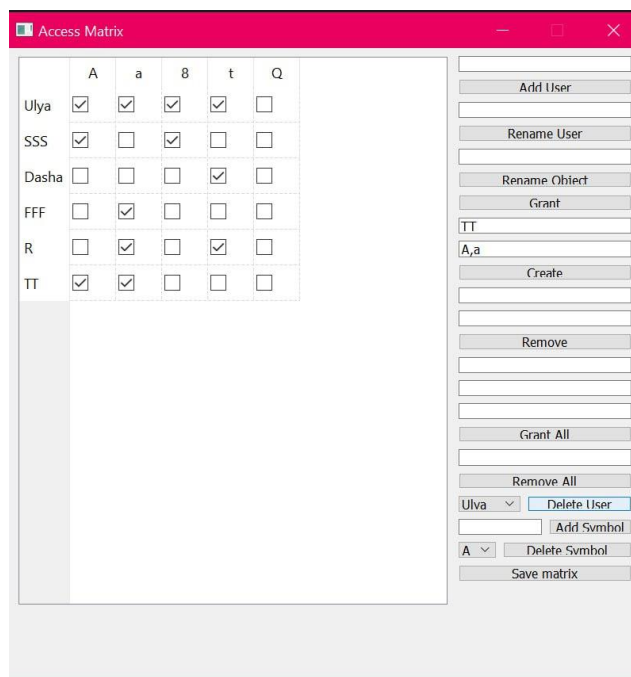


Рисунок 2 – Результат работы функции *Grant*

1.2. Функция функции *create access right*. Данная функция позволяет создавать новые субъекты с доступом к заданному набору объектов. Для этого нужно ввести имена субъектов, которые нужно создать, разделив имена запятой, и соответственно с объектами, далее нажав кнопку «Create», таблица прав доступа обновится, выполнив задачу и выдав всем заданным субъектам права доступа. Результат работы функции представлен на рисунке 3.

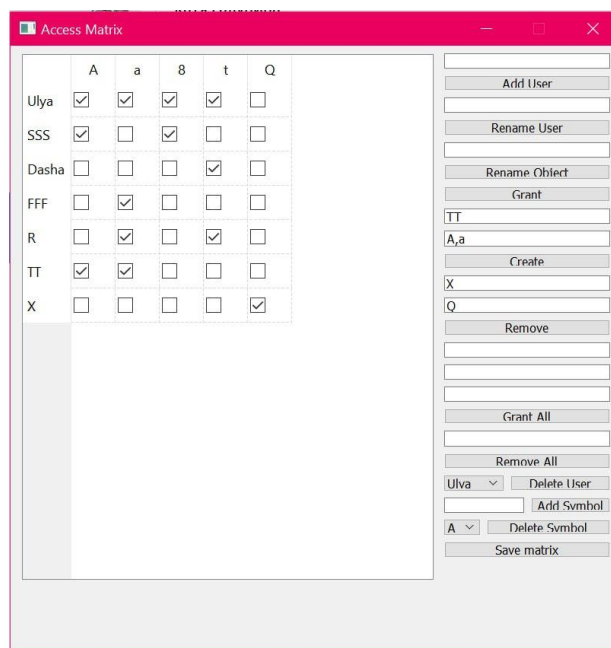


Рисунок 3 – Результат работы функции *Create*

1.3. Функция *remove access right* позволяет теперь забрать у заданного набора субъектов права доступа на заданный набор объектов. Для этого достаточно ввести нужные нам субъекты в поле разделив их запятой и нажать кнопку *Remove*. Результат представлен на рисунке 4.

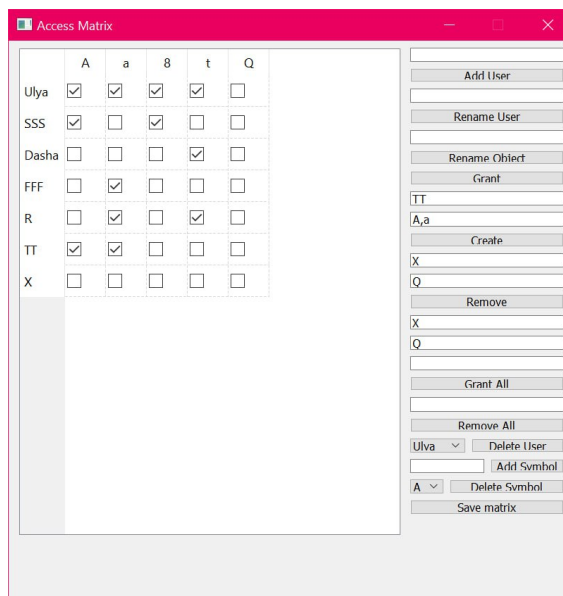


Рисунок 4 – Результат работы функции *Remove*

1.4. Функция *grant all access right* позволяет выдать заданному набору субъектов права доступа на все существующие объекты. Для этого достаточно ввести нужные нам субъекты в поле разделив их запятой и нажать кнопку *Grant All*. Результат представлен на рисунке 5.

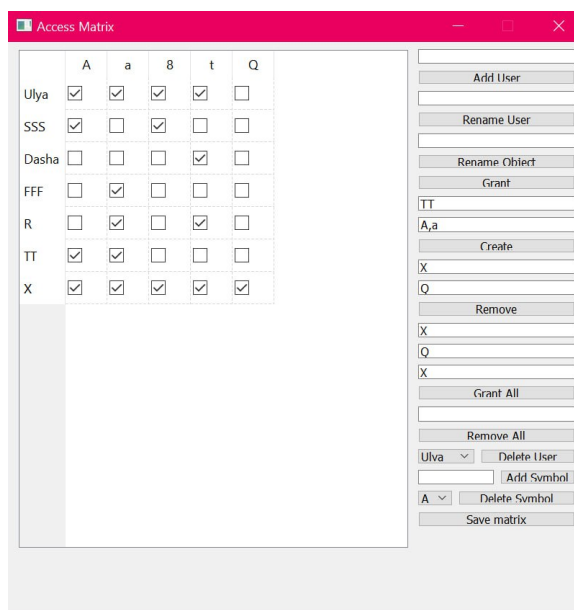


Рисунок 5 – Результат работы функции *Grant All*

1.5. Функция *remove all access right* позволяет забрать у заданного набора субъектов права доступа на все существующие объекты. Для этого достаточно ввести нужные нам субъекты в поле разделив их запятой и нажать кнопку *Remove All*. Результат представлен на рисунке 6.

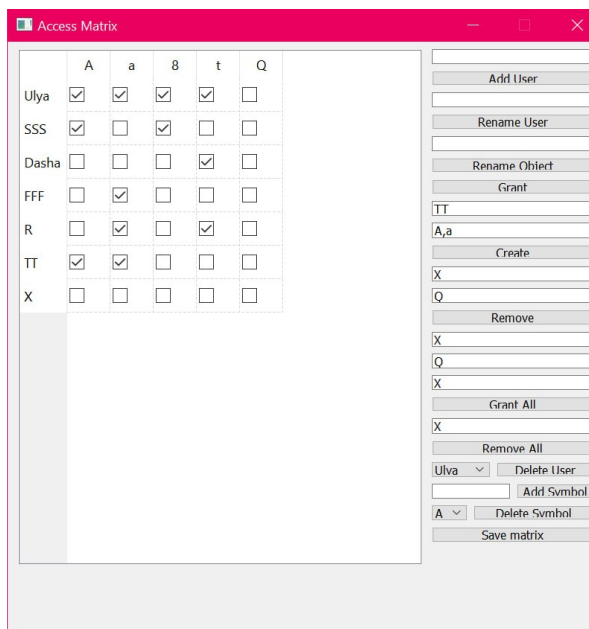


Рисунок 6 – Результат работы функции *Grant All*

2. Была реализована программа пользователя с графическим интерфейсом. Интерфейс представлен на рисунке 7.

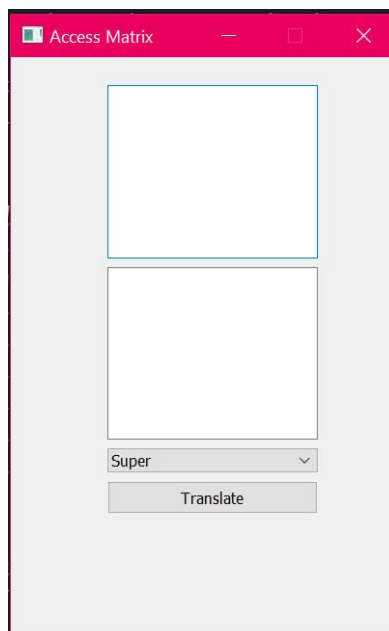


Рисунок 7 – Интерфейс программы

2.1. Теперь введем имя пользователя и строку, права доступа в которой надо выделить для пользователя – рисунок 8.

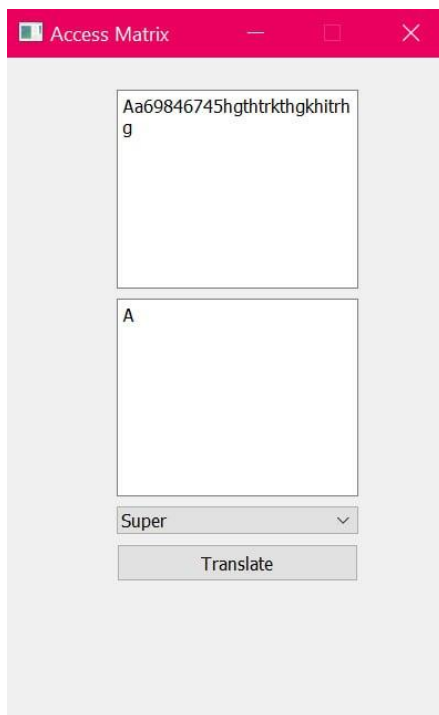


Рисунок 8 – Проверка прав доступа User

Как видно, программа пользователя выводит только те объекты введенной строки, которые совпадают с объектами, на которые имеют права данные субъекты.

Так же в ходе тестирования были выявлены и решены следующие проблемы: создание и выдача прав доступа субъектам и объектам, длина названия которых более одного символа, в программе User вывод доступных объектов не совпадает с введенной строкой.

ВЫВОД

В ходе данной лабораторной работы были реализованы две программы. Первая программа создает, изменяет и удаляет объекты, субъекты, а также выдает, удаляет и создает доступ субъектов к отдельным или всем объектам. Вторая программа проверяет наличие прав доступа у введенного субъекта на введенные в виде строки объекты. Также в ходе лабораторной работы была рассмотрена модель матрицы доступа.

ПРИЛОЖЕНИЕ 1. ИСХОДНЫЙ КОД ПРОГРАММЫ

1. Программа пользователя

1.1. Программа “matrix”

```
from PyQt5 import QtWidgets, QtCore
from PyQt5.QtWidgets import QApplication, QMainWindow
import sys
import pickle
from ui_t import Ui_MainWindow

class MainWindow(QMainWindow, Ui_MainWindow):
    def __init__(self):
        super().__init__()
        self.setupUi(self)
        self.objects = []
        self.users = []
        self.user_states = []
        self.loadMatrix()
        self.translateButton.clicked.connect(self.translate)

    def loadMatrix(self):
        try:
            with open('../admin/matrix_state.pkl', 'rb') as f:
                data = pickle.load(f)
                if len(data) == 3:
                    self.objects, self.users, self.user_states =
data

                    self.updateUserList()
                else:
                    print("Ошибка: неверное количество данных в
файле состояния матрицы.")
            except FileNotFoundError:
                print("Файл состояния матрицы не найден.")

    def updateUserList(self):
        self.userList_translate.clear()
```

```

        self.userList_translate.addItem(self.users)

def translate(self):
    self.outputText.setPlainText("")
    if not self.objects or not self.users:
        return

    stringToRead = self.inputText.toPlainText()
    allowedSymbols = []

    currentUser = self.userList_translate.currentText()
    if not currentUser:
        return

    userIndex = self.users.index(currentUser)
    for col, obj in enumerate(self.objects):
        if self.user_states[userIndex][col] == 1:
            allowedSymbols.append(obj)

    stringToWrite = "".join(ch for ch in stringToRead if ch in
allowedSymbols)
    self.outputText.setPlainText(stringToWrite)

def main():
    app = QApplication(sys.argv)
    window = MainWindow()
    window.show()
    app.exec()

if __name__ == '__main__':
    main()

```

1.2. Программа “prog_table”

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>MainWindow</class>

```

```

<widget class="QMainWindow" name="MainWindow">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>0</y>
      <width>800</width>
      <height>800</height>
    </rect>
  </property>
  <property name="sizePolicy">
    <sizepolicy hsizetype="Fixed" vsizetype="Fixed">
      <horstretch>0</horstretch>
      <verstretch>0</verstretch>
    </sizepolicy>
  </property>
  <property name="minimumSize">
    <size>
      <width>800</width>
      <height>800</height>
    </size>
  </property>
  <property name="maximumSize">
    <size>
      <width>800</width>
      <height>800</height>
    </size>
  </property>
  <property name="windowTitle">
    <string>Access Matrix</string>
  </property>
  <property name="styleSheet">
    <string notr="true"/>
  </property>
  <property name="toolButtonStyle">
    <enum>Qt::ToolButtonIconOnly</enum>

```

```

</property>
<widget class="QWidget" name="centralwidget">
  <widget class="QWidget" name="verticalLayoutWidget">
    <property name="geometry">
      <rect>
        <x>570</x>
        <y>10</y>
        <width>218</width>
        <height>691</height>
      </rect>
    </property>
  <layout class="QVBoxLayout" name="editLayout">
    <item>
      <spacer name="verticalSpacer_2">
        <property name="orientation">
          <enum>Qt::Vertical</enum>
        </property>
        <property name="sizeHint" stdset="0">
          <size>
            <width>20</width>
            <height>40</height>
          </size>
        </property>
      </spacer>
    </item>
    <item>
      <widget class="QPushButton" name="addUserButton">
        <property name="styleSheet">
          <string notr="true"/>
        </property>
        <property name="text">
          <string>Add User</string>
        </property>
      </widget>
    </item>
  </layout>
</widget>

```

```

<item>
<layout class="QHBoxLayout" name="delUserLayout">
  <item>
    <widget class="QComboBox" name="userList_delete">
      <property name="sizePolicy">
        <sizepolicy hsize="Fixed" vsize="Fixed">
          <horstretch>0</horstretch>
          <verstretch>0</verstretch>
        </sizepolicy>
      </property>
      <item>
        <property name="text">
          <string>s_1</string>
        </property>
      </item>
      <item>
        <property name="text">
          <string>s_2</string>
        </property>
      </item>
      <item>
        <property name="text">
          <string>s_3</string>
        </property>
      </item>
    </widget>
  </item>
  <item>
    <widget class="QPushButton" name="delUserButton">
      <property name="text">
        <string>Delete User</string>
      </property>
    </widget>
  </item>
</layout>

```

```

</item>
<item>
<layout class="QHBoxLayout" name="addSymbolLayout">
  <item>
    <widget class="QLineEdit" name="inputSymbol">
      <property name="sizePolicy">
        <sizepolicy hsizeType="Minimum" vsizeType="Fixed">
          <horstretch>0</horstretch>
          <verstretch>0</verstretch>
        </sizepolicy>
      </property>
      <property name="maxLength">
        <number>1</number>
      </property>
    </widget>
  </item>
  <item>
    <widget class="QPushButton" name="addButton">
      <property name="text">
        <string>Add Symbol</string>
      </property>
    </widget>
  </item>
</layout>
</item>
<item>
<layout class="QHBoxLayout" name="delSymbolLayout">
  <item>
    <widget class="QComboBox" name="symbolList">
      <property name="sizePolicy">
        <sizepolicy hsizeType="Fixed" vsizeType="Fixed">
          <horstretch>0</horstretch>
          <verstretch>0</verstretch>
        </sizepolicy>
      </property>
    </widget>
  </item>
</layout>
</item>

```



```

<item>
  <property name="text">
    <string>A</string>
  </property>
</item>
<item>
  <property name="text">
    <string>B</string>
  </property>
</item>
<item>
  <property name="text">
    <string>C</string>
  </property>
</item>
<item>
  <property name="text">
    <string>D</string>
  </property>
</item>
</widget>
</item>
<item>
  <widget class="QPushButton" name="deleteButton">
    <property name="text">
      <string>Delete Symbol</string>
    </property>
  </widget>
</item>
</layout>
</item>
<item>
  <spacer name="verticalSpacer">
    <property name="orientation">
      <enum>Qt::Vertical</enum>

```

```

</property>
<property name="sizeHint" stdset="0">
  <size>
    <width>20</width>
    <height>40</height>
  </size>
</property>
</spacer>
</item>
<item>
<layout class="QHBoxLayout" name="translateLayout">
  <item>
    <widget class="QComboBox" name="userList_translate">
      <item>
        <property name="text">
          <string>s_1</string>
        </property>
      </item>
      <item>
        <property name="text">
          <string>s_2</string>
        </property>
      </item>
      <item>
        <property name="text">
          <string>s_3</string>
        </property>
      </item>
    </widget>
  </item>
  <item>
    <widget class="QPushButton" name="translateButton">
      <property name="text">
        <string>Translate</string>
      </property>
    </widget>
  </item>
</layout>
</item>

```

```

        </widget>
    </item>
</layout>
</item>
</layout>
</widget>
<widget class="QWidget" name="horizontalLayoutWidget_5">
    <property name="geometry">
        <rect>
            <x>10</x>
            <y>710</y>
            <width>781</width>
            <height>80</height>
        </rect>
    </property>
    <layout class="QHBoxLayout" name="textTranslateLayout">
        <item>
            <widget class="QPlainTextEdit" name="inputText">
                <property name="sizePolicy">
                    <sizepolicy hsizetype="Expanding" vsizetype="Preferred">
                        <horstretch>0</horstretch>
                        <verstretch>0</verstretch>
                    </sizepolicy>
                </property>
            </widget>
        </item>
        <item>
            <widget class="QTextBrowser" name="outputText">
                <property name="sizePolicy">
                    <sizepolicy hsizetype="Expanding" vsizetype="Preferred">
                        <horstretch>0</horstretch>
                        <verstretch>0</verstretch>
                    </sizepolicy>
                </property>
                <property name="frameShadow">

```

```

        <enum>QFrame::Sunken</enum>
    </property>
</widget>
</item>
</layout>
</widget>
<widget class="QTableWidget" name="tableWidget">
    <property name="geometry">
        <rect>
            <x>15</x>
            <y>11</y>
            <width>541</width>
            <height>691</height>
        </rect>
    </property>
    <property name="sizePolicy">
        <sizepolicy hstyp="Ignored" vstyp="Ignored">
            <horstretch>0</horstretch>
            <verstretch>0</verstretch>
        </sizepolicy>
    </property>
    <property name="contextMenuPolicy">
        <enum>Qt::DefaultContextMenu</enum>
    </property>
    <property name="frameShadow">
        <enum>QFrame::Sunken</enum>
    </property>
    <property name="verticalScrollBarPolicy">
        <enum>Qt::ScrollBarAlwaysOff</enum>
    </property>
    <property name="horizontalScrollBarPolicy">
        <enum>Qt::ScrollBarAlwaysOff</enum>
    </property>
    <property name="sizeAdjustPolicy">
        <enum>QAbstractScrollArea::AdjustToContents</enum>

```

```

</property>
<property name="autoScroll">
  <bool>true</bool>
</property>
<property name="selectionMode">
  <enum>QAbstractItemView::ExtendedSelection</enum>
</property>
<property name="showGrid">
  <bool>true</bool>
</property>
<property name="gridStyle">
  <enum>Qt::DashLine</enum>
</property>
<attribute name="horizontalHeaderVisible">
  <bool>true</bool>
</attribute>
<attribute name="horizontalHeaderStretchLastSection">
  <bool>false</bool>
</attribute>
<attribute name="verticalHeaderCascadingSectionResizes">
  <bool>false</bool>
</attribute>
<row>
  <property name="text">
    <string>s_1</string>
  </property>
</row>
<row>
  <property name="text">
    <string>s_2</string>
  </property>
</row>
<row>
  <property name="text">
    <string>s_3</string>

```

```

    </property>
</row>
<column>
  <property name="text">
    <string>A</string>
  </property>
</column>
<column>
  <property name="text">
    <string>B</string>
  </property>
</column>
<column>
  <property name="text">
    <string>C</string>
  </property>
</column>
<column>
  <property name="text">
    <string>D</string>
  </property>
</column>
<item row="0" column="0">
  <property name="text">
    <string/>
  </property>
  <property name="checkState">
    <enum>Checked</enum>
  </property>
</item>
<item row="0" column="1">
  <property name="text">
    <string/>
  </property>
  <property name="checkState">

```

```

    <enum>Checked</enum>
  </property>
</item>
<item row="0" column="2">
  <property name="text">
    <string/>
  </property>
  <property name="checkState">
    <enum>Checked</enum>
  </property>
</item>
<item row="0" column="3">
  <property name="text">
    <string/>
  </property>
  <property name="checkState">
    <enum>Checked</enum>
  </property>
</item>
<item row="1" column="0">
  <property name="text">
    <string/>
  </property>
  <property name="checkState">
    <enum>Checked</enum>
  </property>
</item>
<item row="1" column="1">
  <property name="text">
    <string/>
  </property>
  <property name="checkState">
    <enum>Checked</enum>
  </property>
</item>

```

```

<item row="1" column="2">
  <property name="text">
    <string/>
  </property>
  <property name="checkState">
    <enum>Checked</enum>
  </property>
</item>
<item row="1" column="3">
  <property name="text">
    <string/>
  </property>
  <property name="checkState">
    <enum>Checked</enum>
  </property>
</item>
<item row="2" column="0">
  <property name="text">
    <string/>
  </property>
  <property name="checkState">
    <enum>Checked</enum>
  </property>
</item>
<item row="2" column="1">
  <property name="text">
    <string/>
  </property>
  <property name="checkState">
    <enum>Checked</enum>
  </property>
</item>
<item row="2" column="2">
  <property name="text">
    <string/>

```



```

        </property>
        <property name="checkState">
            <enum>Checked</enum>
        </property>
    </item>
    <item row="2" column="3">
        <property name="text">
            <string/>
        </property>
        <property name="checkState">
            <enum>Checked</enum>
        </property>
    </item>
</widget>
<widget class="QCheckBox" name="checkBox">
    <property name="geometry">
        <rect>
            <x>600</x>
            <y>260</y>
            <width>16</width>
            <height>16</height>
        </rect>
    </property>
    <property name="text">
        <string/>
    </property>
</widget>
</widget>
</widget>
<resources/>
<connections/>
</ui>

```

1.3. Программа “ui_t”

```
# -*- coding: utf-8 -*-
```

```

# Form implementation generated from reading ui file
'prog_table.ui'
#
# Created by: PyQt5 UI code generator 5.15.6
#
# WARNING: Any manual changes made to this file will be lost when
pyuic5 is
# run again. Do not edit this file unless you know what you are
doing.

from PyQt5 import QtCore, QtGui, QtWidgets

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(400, 600)
        sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Fixed,
QtWidgets.QSizePolicy.Fixed)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(MainWindow.sizePolicy().hasHeightFo
rWidth())
        MainWindow.setSizePolicy(sizePolicy)
        MainWindow.setMinimumSize(QtCore.QSize(400, 600))
        MainWindow.setMaximumSize(QtCore.QSize(400, 600))
        MainWindow.setStyleSheet("")

MainWindow.setToolButtonStyle(QtCore.Qt.ToolButtonIconOnly)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.verticalLayoutWidget =
QtWidgets.QWidget(self.centralwidget)

```

```

        self.verticalLayoutWidget.setGeometry(QRect(100,
2, 218, 500))

self.verticalLayoutWidget.setObjectName("verticalLayoutWidget")
    self.editLayout =
QtWidgets.QVBoxLayout(self.verticalLayoutWidget)
        self.editLayout.setContentsMargins(0, 0, 0, 0)
        self.editLayout.setObjectName("editLayout")
        spacerItem = QtWidgets.QSpacerItem(20, 40,
QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)
        self.editLayout.addItem(spacerItem)

        # Поле ввода для перевода
        self.inputText =
QtWidgets.QPlainTextEdit(self.verticalLayoutWidget)
        sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Expanding,
QtWidgets.QSizePolicy.Preferred)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.inputText.sizePolicy().hasHeightForWidth())
        self.inputText.setSizePolicy(sizePolicy)
        self.inputText.setObjectName("inputText")
        self.editLayout.addWidget(self.inputText)

        # Поле вывода для перевода
        self.outputText =
QtWidgets.QTextBrowser(self.verticalLayoutWidget)
        sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Expanding,
QtWidgets.QSizePolicy.Preferred)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)

```

```

sizePolicy.setHeightForWidth(self.outputText.sizePolicy().hasHeightForWidth())

        self.outputText.setSizePolicy(sizePolicy)
        self.outputText.setFrameShadow(QtWidgets.QFrame.Sunken)
        self.outputText.setObjectName("outputText")
        self.editLayout.addWidget(self.outputText)


        # Выпадающий список пользователей для перевода
        self.userList_translate =
QtWidgets.QComboBox(self.verticalLayoutWidget)


self.userList_translate.setObjectName("userList_translate")
        self.editLayout.addWidget(self.userList_translate)


        # Кнопка перевода
        self.translateButton =
QtWidgets.QPushButton(self.verticalLayoutWidget)
        self.translateButton.setObjectName("translateButton")
        self.editLayout.addWidget(self.translateButton)


        spacerItem1 = QtWidgets.QSpacerItem(20, 40,
QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)
        self.editLayout.addItem(spacerItem1)


        MainWindow.setCentralWidget(self.centralwidget)


        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow",
"Access Matrix"))

```

```
        self.translateButton.setText(_translate("MainWindow",
"Translate"))
```

```
if __name__ == "__main__":
```

```
    import sys
```

```
    app = QtWidgets.QApplication(sys.argv)
```

```
    MainWindow = QtWidgets.QMainWindow()
```

```
    ui = Ui_MainWindow()
```

```
    ui.setupUi(MainWindow)
```

```
    MainWindow.show()
```

```
    sys.exit(app.exec_())
```

2. Программа администратора

2.1. Программа “matrix”

```
from PyQt5 import QtWidgets, QtCore
```

```
from PyQt5.QtWidgets import QApplication, QMainWindow
```

```
import sys
```

```
import pickle
```

```
from ui_t import Ui_MainWindow
```

```
class MainWindow(QMainWindow, Ui_MainWindow):
```

```
    def __init__(self):
```

```
        super().__init__()
```

```
        self.setupUi(self)    # Инициализация пользовательского  
интерфейса
```

```
        self.objects = [] # Список объектов
```

```
        self.users = [] # Список пользователей
```

```
        self.user_states = [] # Список состояний пользователей
```

```
        # Подключение сигналов к слотам
```

```
        self.addUserButton.clicked.connect(self.addUser)
```

```
        self.addButton.clicked.connect(self.addObject)
```

```
        self.delButton.clicked.connect(self.removeUser)
```

```
        self.deleteButton.clicked.connect(self.removeObject)
```

```
        self.saveMatrixButton.clicked.connect(self.saveMatrix)
```

```

        self.renameUserButton.clicked.connect(self.renameUser)    #
Подключение кнопки переименования пользователя

self.renameObjectButton.clicked.connect(self.renameObject)      #
Подключение кнопки переименования объекта

        self.grantButton.clicked.connect(self.grantAction)       #
Подключение кнопки grant

        self.createButton.clicked.connect(self.create)           #
Подключение кнопки create

        self.removeButton.clicked.connect(self.remove)           #
Подключение кнопки remove

        self.grantAllButton.clicked.connect(self.grant_all)      #
Подключение кнопки grant_all

        self.removeAllButton.clicked.connect(self.remove_all)    #
Подключение кнопки remove_all

        self.loadMatrix() # Загрузка состояния матрицы

self.tableWidget.itemChanged.connect(self.updateUserStates)    #
Обновление состояния пользователя при изменении элемента


def loadMatrix(self):
    try:
        with open('matrix_state.pkl', 'rb') as f:
            data = pickle.load(f)
            if len(data) == 3:
                self.objects, self.users, self.user_states =
data
                self.updateMatrixUI() # Обновление
пользовательского интерфейса
            else:
                print("Ошибка: неверное количество данных в
файле состояния матрицы.")
    except FileNotFoundError:
        pass

```

```

def saveMatrix(self):
    with open('matrix_state.pkl', 'wb') as f:
        pickle.dump((self.objects, self.users,
self.user_states), f)
    print("Состояние матрицы успешно сохранено.")

def updateMatrixUI(self):
    self.tableWidget.clear()
    self.userList_delete.clear()
    self.symbolList.clear()

    for user in self.users:
        self.userList_delete.addItem(user)

    for symbol in self.objects:
        self.symbolList.addItem(symbol)

    self.tableWidget.setRowCount(len(self.users))
    self.tableWidget.setColumnCount(len(self.objects))
    self.tableWidget.setHorizontalHeaderLabels(self.objects)
# Установка заголовков столбцов
    self.tableWidget.setVerticalHeaderLabels(self.users) #
Установка заголовков строк
    for row in range(len(self.users)):
        for col in range(len(self.objects)):
            item = QtWidgets.QTableWidgetItem()
            try:
                item.setCheckState(QtCore.Qt.Unchecked if
self.user_states[row][col] == 0 else QtCore.Qt.Checked)
            except IndexError:
                item.setCheckState(QtCore.Qt.Unchecked)
            self.tableWidget.setItem(row, col, item)

def updateUserStates(self, item):

```

```

        row = item.row()
        col = item.column()
        if row < len(self.user_states) and col <
len(self.user_states[row]):
            self.user_states[row][col] = 1 if item.checkState() ==
QtCore.Qt.Checked else 0
            print(
                f"Обновлено состояние галочки для пользователя
{self.users[row]} и объекта {self.objects[col]}:
{item.checkState()}")

    def addUser(self):
        new_user_name = self.inputUser.text()
        if new_user_name not in self.users and new_user_name !=
"":
            self.users.append(new_user_name)

            rowPosition = self.tableWidget.rowCount()
            self.tableWidget.insertRow(rowPosition)

            self.tableWidget.setVerticalHeaderLabels(self.users)
# Установка заголовков строк

            self.userList_delete.clear()
            self.userList_delete.addItem(self.users)

            self.user_states.append([0] * len(self.objects))

            for i in range(len(self.objects)):
                item = QtWidgets.QTableWidgetItem()
                item.setCheckState(QtCore.Qt.Unchecked)
                self.tableWidget.setItem(rowPosition, i, item)
            else:
                print("Пользователь с таким именем уже существует или
имя пустое.")

```



```

def addObject(self):
    symbol_text = self.inputSymbol.text()
    if symbol_text not in self.objects and symbol_text != "":
        self.objects.append(str(symbol_text))    # Добавляем
текст символа как строку
        self.tableWidget.setColumnCount(len(self.objects))

self.tableWidget.setHorizontalHeaderLabels(self.objects)    #
Обновляем заголовки столбцов
        self.symbolList.clear()
        self.symbolList.addItem(self.objects)
    else:
        return
    for i in range(len(self.users)):
        item = QtWidgets.QTableWidgetItem()
        item.setCheckState(QtCore.Qt.Unchecked)
        self.tableWidget.setItem(i,
self.tableWidget.columnCount() - 1, item)
        for user_state in self.user_states:
            user_state.append(0)

def removeUser(self):
    if len(self.users) == 0:
        return
    count = self.userList_delete.currentIndex()
    if count >= 0:
        self.users.pop(count)
        self.user_states.pop(count)
        self.tableWidget.removeRow(count)
        self.userList_delete.clear()
        self.userList_delete.addItem(self.users)

def removeObject(self):
    if len(self.objects) == 0:

```

```

        return
count = self.symbolList.currentIndex()
if count >= 0:
    self.objects.pop(count)
    for user_state in self.user_states:
        user_state.pop(count)
    self.tableWidget.removeColumn(count)
    self.symbolList.clear()
    self.symbolList.addItem(self.objects)

def renameUser(self):
    new_user_name = self.renameUserInput.text()
    selected_user_index = self.userList_delete.currentIndex()
    if new_user_name not in self.users and new_user_name != ""
and selected_user_index >= 0:
        self.users[selected_user_index] = new_user_name

self.userList_delete.setItemText(selected_user_index,
new_user_name)
        self.updateMatrixUI() # Обновление пользовательского
интерфейса
    else:
        print("Неверное имя пользователя или пользователь с
таким именем уже существует.")

def renameObject(self):
    new_object_name = self.renameObjectInput.text()
    selected_object_index = self.symbolList.currentIndex()
    if new_object_name not in self.objects and new_object_name
!= "" and selected_object_index >= 0:
        self.objects[selected_object_index] = new_object_name
        self.symbolList.setItemText(selected_object_index,
new_object_name)
        self.updateMatrixUI() # Обновление пользовательского
интерфейса

```

```

        else:
            print("Неверное имя объекта или объект с таким именем
уже существует.")

def grantAction(self):
    users_text = self.grantUsersInput.text()
    objects_text = self.grantObjectsInput.text()
    users = users_text.split(',') if users_text else []
    objects = objects_text.split(',') if objects_text else []
    self.user_states = grant(users, objects,
self.user_states, self.users, self.objects)
    self.updateMatrixUI()

def create(self):
    new_user_name = self.createUserInput.text()
    objects_text = self.createObjectsInput.text()
    objects = objects_text.split(',') if objects_text else []
    self.users, self.objects, self.user_states =
create(new_user_name, objects, self.users, self.user_states,
self.objects)
    self.updateMatrixUI()

def remove(self):
    users_text = self.removeUsersInput.text()
    objects_text = self.removeObjectsInput.text()
    users = users_text.split(',') if users_text else []
    objects = objects_text.split(',') if objects_text else []
    self.user_states = remove(users, objects,
self.user_states, self.users, self.objects)
    self.updateMatrixUI()

def grant_all(self):
    users_text = self.grantAllUsersInput.text()
    users = users_text.split(',') if users_text else []

```

```

        self.user_states = grant_all(users, self.user_states,
self.users, self.objects)
        self.updateMatrixUI()

def remove_all(self):
    users_text = self.removeAllUsersInput.text()
    users = users_text.split(',') if users_text else []
    self.user_states = remove_all(users, self.user_states,
self.users, self.objects)
    self.updateMatrixUI()

def grant(users, objects, user_states, users_list, objects_list):
    for user in users:
        if user in users_list:
            user_index = users_list.index(user)
            for obj in objects:
                if obj in objects_list:
                    obj_index = objects_list.index(obj)
                    user_states[user_index][obj_index] = 1
    return user_states

def create(new_user_name, objects, users, user_states,
objects_list):
    # Проверяем, не существует ли пользователь с таким именем и не
пустое ли имя
    if new_user_name not in users and new_user_name != "":
        users.append(new_user_name)
        user_states.append([0] * len(objects_list))
    for obj in objects:
        if obj not in objects_list and obj != "" and len(obj) <=
1:
            objects_list.append(obj)
            for user_state in user_states:
                user_state.append(0)
    else:

```

```
        print("Объект с таким именем уже существует, имя  
пустое или имя превышает один символ.")
```

```
user_index = users.index(new_user_name)  
for obj in objects:  
    if obj in objects_list:  
        obj_index = objects_list.index(obj)  
        user_states[user_index][obj_index] = 1
```

```
# Возвращаем обновленные списки пользователей, объектов и  
состояний прав доступа
```

```
return users, objects_list, user_states
```

```
def remove(users, objects, user_states, users_list, objects_list):
```

```
    for user in users:  
        if user in users_list:  
            user_index = users_list.index(user)  
            for obj in objects:  
                if obj in objects_list:  
                    obj_index = objects_list.index(obj)  
                    user_states[user_index][obj_index] = 0  
    return user_states
```

```
def grant_all(users, user_states, users_list, objects_list):
```

```
    for user in users:  
        if user in users_list:  
            user_index = users_list.index(user)  
            for obj_index in range(len(objects_list)):  
                user_states[user_index][obj_index] = 1  
    return user_states
```

```
def remove_all(users, user_states, users_list, objects_list):
```

```
    for user in users:  
        if user in users_list:
```

```

        user_index = users_list.index(user)
        for obj_index in range(len(objects_list)):
            user_states[user_index][obj_index] = 0
    return user_states

def main():
    app = QApplication(sys.argv)
    window = MainWindow()
    window.show()
    app.exec()

if __name__ == '__main__':
    main()

```

2.2. Программа “prog_table”

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>MainWindow</class>
    <widget class="QMainWindow" name="MainWindow">
        <property name="geometry">
            <rect>
                <x>0</x>
                <y>0</y>
                <width>800</width>
                <height>800</height>
            </rect>
        </property>
        <property name="sizePolicy">
            <sizepolicy hsizeType="Fixed" vsizetype="Fixed">
                <horstretch>0</horstretch>
                <verstretch>0</verstretch>
            </sizepolicy>
        </property>
        <property name="minimumSize">
            <size>

```

```

        <width>800</width>
        <height>800</height>
    </size>
</property>
<property name="maximumSize">
    <size>
        <width>800</width>
        <height>800</height>
    </size>
</property>
<property name="windowTitle">
    <string>Access Matrix</string>
</property>
<property name="styleSheet">
    <string notr="true"/>
</property>
<property name="toolButtonStyle">
    <enum>Qt::ToolButtonIconOnly</enum>
</property>
<widget class="QWidget" name="centralwidget">
    <widget class="QWidget" name="verticalLayoutWidget">
        <property name="geometry">
            <rect>
                <x>570</x>
                <y>10</y>
                <width>218</width>
                <height>691</height>
            </rect>
        </property>
    <layout class="QVBoxLayout" name="editLayout">
        <item>
            <spacer name="verticalSpacer_2">
                <property name="orientation">
                    <enum>Qt::Vertical</enum>
                </property>
            </spacer>
        </item>
    </layout>
</widget>

```

```

    <property name="sizeHint" stdset="0">
      <size>
        <width>20</width>
        <height>40</height>
      </size>
    </property>
  </spacer>
</item>
<item>
  <widget class="QPushButton" name="addUserButton">
    <property name="styleSheet">
      <string notr="true"/>
    </property>
    <property name="text">
      <string>Add User</string>
    </property>
  </widget>
</item>
<item>
  <layout class="QHBoxLayout" name="delUserLayout">
    <item>
      <widget class="QComboBox" name="userList_delete">
        <property name="sizePolicy">
          <sizepolicy hsizetype="Fixed" vsizetype="Fixed">
            <horstretch>0</horstretch>
            <verstretch>0</verstretch>
          </sizepolicy>
        </property>
      </item>
      <property name="text">
        <string>s_1</string>
      </property>
    </item>
    <item>
      <property name="text">

```



```

        <string>s_2</string>
    </property>
</item>
<item>
    <property name="text">
        <string>s_3</string>
    </property>
</item>
</widget>
</item>
<item>
    <widget class="QPushButton" name="delButton">
        <property name="text">
            <string>Delete User</string>
        </property>
    </widget>
</item>
</layout>
</item>
<item>
    <layout class="QHBoxLayout" name="addSymbolLayout">
        <item>
            <widget class="QLineEdit" name="inputSymbol">
                <property name="sizePolicy">
                    <sizepolicy hsizetype="Minimum" vsizetype="Fixed">
                        <horstretch>0</horstretch>
                        <verstretch>0</verstretch>
                    </sizepolicy>
                </property>
                <property name="maxLength">
                    <number>1</number>
                </property>
            </widget>
        </item>
        <item>

```

```

    <widget class="QPushButton" name="addButton">
        <property name="text">
            <string>Add Symbol</string>
        </property>
    </widget>
</item>
</layout>
</item>
<item>
    <layout class="QHBoxLayout" name="delSymbolLayout">
        <item>
            <widget class="QComboBox" name="symbolList">
                <property name="sizePolicy">
                    <sizepolicy hsietype="Fixed" vsizetype="Fixed">
                        <horstretch>0</horstretch>
                        <verstretch>0</verstretch>
                    </sizepolicy>
                </property>
            <item>
                <property name="text">
                    <string>A</string>
                </property>
            </item>
            <item>
                <property name="text">
                    <string>B</string>
                </property>
            </item>
            <item>
                <property name="text">
                    <string>C</string>
                </property>
            </item>
            <item>
                <property name="text">

```

```

        <string>D</string>
    </property>
</item>
</widget>
</item>
<item>
    <widget class="QPushButton" name="deleteButton">
        <property name="text">
            <string>Delete Symbol</string>
        </property>
    </widget>
</item>
</layout>
</item>
<item>
    <spacer name="verticalSpacer">
        <property name="orientation">
            <enum>Qt::Vertical</enum>
        </property>
        <property name="sizeHint" stdset="0">
            <size>
                <width>20</width>
                <height>40</height>
            </size>
        </property>
    </spacer>
</item>
<item>
    <layout class="QHBoxLayout" name="translateLayout">
        <item>
            <widget class="QComboBox" name="userList_translate">
                <item>
                    <property name="text">
                        <string>s_1</string>
                    </property>

```

```

    </item>
    <item>
        <property name="text">
            <string>s_2</string>
        </property>
    </item>
    <item>
        <property name="text">
            <string>s_3</string>
        </property>
    </item>
</widget>
</item>
<item>
    <widget class="QPushButton" name="translateButton">
        <property name="text">
            <string>Translate</string>
        </property>
    </widget>
</item>
</layout>
</item>
</layout>
</widget>
<widget class="QWidget" name="horizontalLayoutWidget_5">
    <property name="geometry">
        <rect>
            <x>10</x>
            <y>710</y>
            <width>781</width>
            <height>80</height>
        </rect>
    </property>
    <layout class="QHBoxLayout" name="textTranslateLayout">
        <item>

```

```

<widget class="QPlainTextEdit" name="inputText">
  <property name="sizePolicy">
    <sizepolicy hsize="Expanding" vsize="Preferred">
      <horstretch>0</horstretch>
      <verstretch>0</verstretch>
    </sizepolicy>
  </property>
</widget>
</item>
<item>
  <widget class="QTextBrowser" name="outputText">
    <property name="sizePolicy">
      <sizepolicy hsize="Expanding" vsize="Preferred">
        <horstretch>0</horstretch>
        <verstretch>0</verstretch>
      </sizepolicy>
    </property>
    <property name="frameShadow">
      <enum>QFrame::Sunken</enum>
    </property>
  </widget>
</item>
</layout>
</widget>
<widget class="QTableWidget" name="tableWidget">
  <property name="geometry">
    <rect>
      <x>15</x>
      <y>11</y>
      <width>541</width>
      <height>691</height>
    </rect>
  </property>
  <property name="sizePolicy">
    <sizepolicy hsize="Ignored" vsize="Ignored">

```

```

    <horstretch>0</horstretch>
    <verstretch>0</verstretch>
  </sizepolicy>
</property>
<property name="contextMenuPolicy">
  <enum>Qt::DefaultContextMenu</enum>
</property>
<property name="frameShadow">
  <enum>QFrame::Sunken</enum>
</property>
<property name="verticalScrollBarPolicy">
  <enum>Qt::ScrollBarAlwaysOff</enum>
</property>
<property name="horizontalScrollBarPolicy">
  <enum>Qt::ScrollBarAlwaysOff</enum>
</property>
<property name="sizeAdjustPolicy">
  <enum>QAbstractScrollArea::AdjustToContents</enum>
</property>
<property name="autoScroll">
  <bool>true</bool>
</property>
<property name="selectionMode">
  <enum>QAbstractItemView::ExtendedSelection</enum>
</property>
<property name="showGrid">
  <bool>true</bool>
</property>
<property name="gridStyle">
  <enum>Qt::DashLine</enum>
</property>
<attribute name="horizontalHeaderVisible">
  <bool>true</bool>
</attribute>
<attribute name="horizontalHeaderStretchLastSection">

```

```

    <bool>false</bool>
</attribute>
<attribute name="verticalHeaderCascadingSectionResizes">
    <bool>false</bool>
</attribute>
<row>
    <property name="text">
        <string>s_1</string>
    </property>
</row>
<row>
    <property name="text">
        <string>s_2</string>
    </property>
</row>
<row>
    <property name="text">
        <string>s_3</string>
    </property>
</row>
<column>
    <property name="text">
        <string>A</string>
    </property>
</column>
<column>
    <property name="text">
        <string>B</string>
    </property>
</column>
<column>
    <property name="text">
        <string>C</string>
    </property>
</column>

```

```

<column>
  <property name="text">
    <string>D</string>
  </property>
</column>
<item row="0" column="0">
  <property name="text">
    <string/>
  </property>
  <property name="checkState">
    <enum>Checked</enum>
  </property>
</item>
<item row="0" column="1">
  <property name="text">
    <string/>
  </property>
  <property name="checkState">
    <enum>Checked</enum>
  </property>
</item>
<item row="0" column="2">
  <property name="text">
    <string/>
  </property>
  <property name="checkState">
    <enum>Checked</enum>
  </property>
</item>
<item row="0" column="3">
  <property name="text">
    <string/>
  </property>
  <property name="checkState">
    <enum>Checked</enum>

```



```

    </property>
</item>
<item row="1" column="0">
  <property name="text">
    <string/>
  </property>
  <property name="checkState">
    <enum>Checked</enum>
  </property>
</item>
<item row="1" column="1">
  <property name="text">
    <string/>
  </property>
  <property name="checkState">
    <enum>Checked</enum>
  </property>
</item>
<item row="1" column="2">
  <property name="text">
    <string/>
  </property>
  <property name="checkState">
    <enum>Checked</enum>
  </property>
</item>
<item row="1" column="3">
  <property name="text">
    <string/>
  </property>
  <property name="checkState">
    <enum>Checked</enum>
  </property>
</item>
<item row="2" column="0">

```

```

    <property name="text">
      <string/>
    </property>
    <property name="checkState">
      <enum>Checked</enum>
    </property>
  </item>
  <item row="2" column="1">
    <property name="text">
      <string/>
    </property>
    <property name="checkState">
      <enum>Checked</enum>
    </property>
  </item>
  <item row="2" column="2">
    <property name="text">
      <string/>
    </property>
    <property name="checkState">
      <enum>Checked</enum>
    </property>
  </item>
  <item row="2" column="3">
    <property name="text">
      <string/>
    </property>
    <property name="checkState">
      <enum>Checked</enum>
    </property>
  </item>
</widget>
<widget class="QCheckBox" name="checkBox">
  <property name="geometry">
    <rect>

```

```

        <x>600</x>
        <y>260</y>
        <width>16</width>
        <height>16</height>
    </rect>
</property>
<property name="text">
    <string/>
</property>
</widget>
</widget>
</widget>
<resources/>
<connections/>
</ui>

```

2.3. Программа “ui_t”

```
from PyQt5 import QtCore, QtGui, QtWidgets
```

```

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(800, 800)
        sizePolicy
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Fixed,
QtWidgets.QSizePolicy.Fixed)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(MainWindow.sizePolicy().hasHeightFo
rWidth())
        MainWindow.setSizePolicy(sizePolicy)
        MainWindow.setMinimumSize(QtCore.QSize(800, 800))
        MainWindow.setMaximumSize(QtCore.QSize(800, 800))
        MainWindow.setStyleSheet("")

```

```

MainWindow.setToolButtonStyle(QtCore.Qt.ToolButtonIconOnly)
    self.centralwidget = QtWidgets.QWidget(MainWindow)
    self.centralwidget.setObjectName("centralwidget")
    self.verticalLayoutWidget =
QtWidgets.QWidget(self.centralwidget)
    self.verticalLayoutWidget.setGeometry(QtCore.QRect(570,
10, 218, 691))

self.verticalLayoutWidget.setObjectName("verticalLayoutWidget")
    self.editLayout =
QtWidgets.QVBoxLayout(self.verticalLayoutWidget)
    self.editLayout.setContentsMargins(0, 0, 0, 0)
    self.editLayout.setObjectName("editLayout")
    spacerItem = QtWidgets.QSpacerItem(20, 40,
QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)
    self.editLayout.addItem(spacerItem)

    # Создание поля ввода для имени пользователя
    self.inputUser =
QtWidgets.QLineEdit(self.verticalLayoutWidget)
    self.inputUser.setObjectName("inputUser")
    self.editLayout.addWidget(self.inputUser)

    self.addUserButton =
QtWidgets.QPushButton(self.verticalLayoutWidget)
    self.addUserButton.setStyleSheet("")
    self.editLayout.addWidget(self.addUserButton)

    # Создание поля для ввода нового имени пользователя
    self.renameUserInput =
QtWidgets.QLineEdit(self.verticalLayoutWidget)
    self.renameUserInput.setObjectName("renameUserInput")
    self.editLayout.addWidget(self.renameUserInput)

```

```

        # Создание кнопки для переименования пользователя
        self.renameUserButton                                     =
QtWidgets.QPushButton(self.verticalLayoutWidget)
        self.renameUserButton.setObjectName("renameUserButton")
        self.renameUserButton.setText("Rename User")
        self.editLayout.addWidget(self.renameUserButton)

        # Добавление поля ввода для нового имени объекта
        self.renameObjectInput                                   =
QtWidgets.QLineEdit(self.verticalLayoutWidget)

        self.renameObjectInput.setObjectName("renameObjectInput")
        self.editLayout.addWidget(self.renameObjectInput)

        # Добавление кнопки для переименования объекта
        self.renameObjectButton                                  =
QtWidgets.QPushButton(self.verticalLayoutWidget)

        self.renameObjectButton.setObjectName("renameObjectButton")
        self.renameObjectButton.setText("Rename Object")
        self.editLayout.addWidget(self.renameObjectButton)

        # Добавление кнопки и поля ввода для grant
        self.grantButton                                        =
QtWidgets.QPushButton(self.verticalLayoutWidget)
        self.grantButton.setObjectName("grantButton")
        self.grantButton.setText("Grant")
        self.editLayout.addWidget(self.grantButton)

        self.grantUsersInput                                    =
QtWidgets.QLineEdit(self.verticalLayoutWidget)
        self.grantUsersInput.setObjectName("grantUsersInput")
        self.editLayout.addWidget(self.grantUsersInput)

```

```

        self.grantObjectsInput                                     =
QtWidgets.QLineEdit(self.verticalLayoutWidget)

self.grantObjectsInput.setObjectName("grantObjectsInput")
    self.editLayout.addWidget(self.grantObjectsInput)

    # Добавление кнопки и поля ввода для create
    self.createButton                                           =
QtWidgets.QPushButton(self.verticalLayoutWidget)
    self.createButton.setObjectName("createButton")
    self.createButton.setText("Create")
    self.editLayout.addWidget(self.createButton)

    self.createUserInput                                         =
QtWidgets.QLineEdit(self.verticalLayoutWidget)
    self.createUserInput.setObjectName("createUserInput")
    self.editLayout.addWidget(self.createUserInput)

    self.createObjectsInput                                       =
QtWidgets.QLineEdit(self.verticalLayoutWidget)

self.createObjectsInput.setObjectName("createObjectsInput")
    self.editLayout.addWidget(self.createObjectsInput)

    # Добавление кнопки и поля ввода для remove
    self.removeButton                                           =
QtWidgets.QPushButton(self.verticalLayoutWidget)
    self.removeButton.setObjectName("removeButton")
    self.removeButton.setText("Remove")
    self.editLayout.addWidget(self.removeButton)

    self.removeUsersInput                                         =
QtWidgets.QLineEdit(self.verticalLayoutWidget)
    self.removeUsersInput.setObjectName("removeUsersInput")

```

```

        self.editLayout.addWidget(self.removeUsersInput)

        self.removeObjectsInput =
QtWidgets.QLineEdit(self.verticalLayoutWidget)

self.removeObjectsInput.setObjectName("removeObjectsInput")
        self.editLayout.addWidget(self.removeObjectsInput)

        # Добавление кнопок для grant_all и remove_all
        # Добавление строки для ввода субъектов для grant_all

        self.grantAllUsersInput =
QtWidgets.QLineEdit(self.verticalLayoutWidget)

self.grantAllUsersInput.setObjectName("grantAllUsersInput")
        self.editLayout.addWidget(self.grantAllUsersInput)

        self.grantAllButton =
QtWidgets.QPushButton(self.verticalLayoutWidget)
        self.grantAllButton.setObjectName("grantAllButton")
        self.grantAllButton.setText("Grant All")
        self.editLayout.addWidget(self.grantAllButton)

        self.removeAllUsersInput =
QtWidgets.QLineEdit(self.verticalLayoutWidget)

self.removeAllUsersInput.setObjectName("removeAllUsersInput")
        self.editLayout.addWidget(self.removeAllUsersInput)

        self.removeAllButton =
QtWidgets.QPushButton(self.verticalLayoutWidget)
        self.removeAllButton.setObjectName("removeAllButton")

```

```

self.removeAllButton.setText("Remove All")
self.editLayout.addWidget(self.removeAllButton)


self.delUserLayout = QtWidgets.QHBoxLayout()
self.delUserLayout.setObjectName("delUserLayout")
self.userList_delete =
QtWidgets.QComboBox(self.verticalLayoutWidget)
sizePolicy =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Fixed,
QtWidgets.QSizePolicy.Fixed)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.userList_delete.sizePolicy().h
asHeightForWidth())
self.userList_delete.setSizePolicy(sizePolicy)
self.userList_delete.setObjectName("userList_delete")
self.delUserLayout.addWidget(self.userList_delete)
self.delButton =
QtWidgets.QPushButton(self.verticalLayoutWidget)
self.delButton.setObjectName("delButton")
self.delUserLayout.addWidget(self.delButton)
self.editLayout.addLayout(self.delUserLayout)
self.addSymbolLayout = QtWidgets.QHBoxLayout()
self.addSymbolLayout.setObjectName("addSymbolLayout")

```



```

        self.inputSymbol
QtWidgets.QLineEdit(self.verticalLayoutWidget)
        sizePolicy
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Minimum,
QtWidgets.QSizePolicy.Fixed)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.inputSymbol.sizePolicy().hasHeightForWidth())
        self.inputSymbol.setSizePolicy(sizePolicy)
        self.inputSymbol.setMaxLength(1)
        self.inputSymbol.setObjectName("inputSymbol")
        self.addSymbolLayout.addWidget(self.inputSymbol)
        self.addButton
QtWidgets.QPushButton(self.verticalLayoutWidget)
        self.addButton.setObjectName("addButton")
        self.addSymbolLayout.addWidget(self.addButton)
        self.editLayout.addLayout(self.addSymbolLayout)
        self.delSymbolLayout = QtWidgets.QHBoxLayout()
        self.delSymbolLayout.setObjectName("delSymbolLayout")
        self.symbolList
QtWidgets.QComboBox(self.verticalLayoutWidget)
        sizePolicy
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Fixed,
QtWidgets.QSizePolicy.Fixed)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.symbolList.sizePolicy().hasHeightForWidth())
        self.symbolList.setSizePolicy(sizePolicy)
        self.symbolList.setObjectName("symbolList")
        self.delSymbolLayout.addWidget(self.symbolList)

```

```

        self.deleteButton                                     =
QtWidgets.QPushButton(self.verticalLayoutWidget)
        self.deleteButton.setObjectName("deleteButton")
        self.delSymbolLayout.addWidget(self.deleteButton)
        self.editLayout.addLayout(self.delSymbolLayout)
        self.saveMatrixButton                               =
QtWidgets.QPushButton(self.verticalLayoutWidget)
        self.saveMatrixButton.setObjectName("saveMatrixButton")
        self.saveMatrixButton.setText("Save matrix")
        self.editLayout.addWidget(self.saveMatrixButton)

        # Добавление QLabel для отображения сообщений
        self.saveStatusLabel                                 =
QtWidgets.QLabel(self.verticalLayoutWidget)
        self.saveStatusLabel.setObjectName("saveStatusLabel")
        self.saveStatusLabel.setText("")
        self.editLayout.addWidget(self.saveStatusLabel)

        spacerItem1 = QtWidgets.QSpacerItem(20, 40,
QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Expanding)
        self.editLayout.addItem(spacerItem1)

        self.tableWidget                                     =
QtWidgets.QTableWidget(self.centralwidget)
        self.tableWidget.setGeometry(QtCore.QRect(15, 11, 541,
691))
        sizePolicy                                           =
QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Ignored,
QtWidgets.QSizePolicy.Ignored)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)

        sizePolicy.setHeightForWidth(self.tableWidget.sizePolicy().hasHeightForWidth())
        self.tableWidget.setSizePolicy(sizePolicy)

```

```

self.tableWidget.setContextMenuPolicy(QtCore.Qt.DefaultContextMe
nu)

        self.tableWidget.setFrameShadow(QtWidgets.QFrame.Sunken)

self.tableWidget.setVerticalScrollBarPolicy(QtCore.Qt.ScrollBarA
lwaysOff)

self.tableWidget.setHorizontalScrollBarPolicy(QtCore.Qt.ScrollBa
rAlwaysOff)

self.tableWidget.setSizeAdjustPolicy(QtWidgets.QAbstractScrollAr
ea.AdjustToContents)

        self.tableWidget.setAutoScroll(True)

self.tableWidget.setSelectionMode(QtWidgets.QAbstractItemView.Ex
tendedSelection)

        self.tableWidget.setShowGrid(True)
        self.tableWidget.setGridStyle(QtCore.Qt.DashLine)
        self.tableWidget.setObjectName("tableWidget")
        self.tableWidget.setColumnCount(0)
        self.tableWidget.setRowCount(0)

        self.tableWidget.horizontalHeader().setVisible(True)

self.tableWidget.horizontalHeader().setStretchLastSection(False)

self.tableWidget.verticalHeader().setCascadingSectionResizes(Fal
se)

        MainWindow.setCentralWidget(self.centralwidget)

        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

```

```

def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow",
"Access Matrix"))
    self.addUserButton.setText(_translate("MainWindow", "Add
User"))
    self.userList_delete.setItemText(0,
_translate("MainWindow", "s_1"))
    self.userList_delete.setItemText(1,
_translate("MainWindow", "s_2"))
    self.userList_delete.setItemText(2,
_translate("MainWindow", "s_3"))
    self.delButton.setText(_translate("MainWindow", "Delete
User"))
    self.addButton.setText(_translate("MainWindow", "Add
Symbol"))
    self.symbolList.setItemText(0, _translate("MainWindow",
"A"))
    self.symbolList.setItemText(1, _translate("MainWindow",
"B"))
    self.symbolList.setItemText(2, _translate("MainWindow",
"C"))
    self.symbolList.setItemText(3, _translate("MainWindow",
"D"))
    self.deleteButton.setText(_translate("MainWindow",
"Delete Symbol"))
    self.saveMatrixButton.setText(_translate("MainWindow",
"Save matrix"))

```

```

        self.renameUserButton.setText(_translate("MainWindow",
"Rename User"))
        self.renameObjectButton.setText(_translate("MainWindow",
"Rename Object"))
        __sortingEnabled = self.tableWidget.isSortingEnabled()
        self.tableWidget.setSortingEnabled(False)
        self.tableWidget.setSortingEnabled(__sortingEnabled)

if __name__ == "__main__":
    import sys

    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())

```