Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №9**

**«Работа с типизированными файлами»**

**ПО МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ**

Выполнил: студент учебной группы

ИСПк-203-52-00

Поскребышев Роман Алексеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. **Цель домашней контрольной работы**: получение базовых навыков реализации приложений с графическим интерфейсом пользователя на основе событийно-ориентированного программирования.
2. **Скриншот задания с вариантом**:

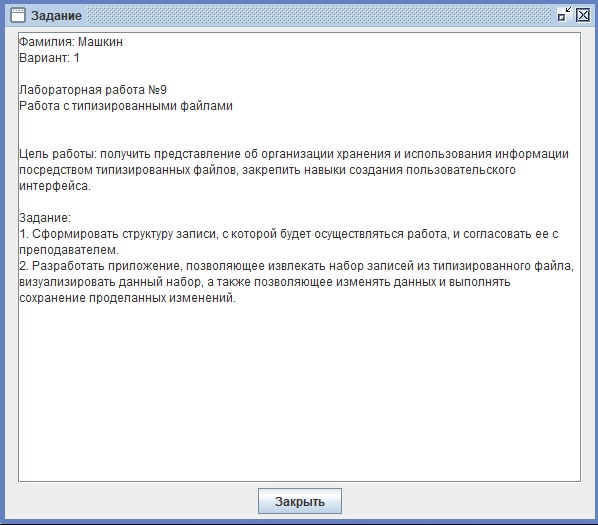


Рисунок 1 – Скриншот задания

1. **Код программы:**

**Код файла start**

**from PyQt5 import QtWidgets**

**from DCR9 import Ui\_MainWindow # импорт нашего сгенерированного файла**

**import sys**

**import mysql.connector**

**y=""**

**class mywindow(QtWidgets.QMainWindow):**

**global y**

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

**super(mywindow, self).\_\_init\_\_()**

**self.ui = Ui\_MainWindow()**

**self.ui.setupUi(self)**

**self.ui.stackedWidget.setCurrentIndex(0)**

**self.ui.pushButton\_add.clicked.connect(self.widget\_Add)**

**self.ui.pushButton\_red.clicked.connect(self.widget\_Red)**

**self.ui.pushButton\_undo\_1.clicked.connect(self.widget\_Undo\_1)**

**self.ui.pushButton\_undo\_2.clicked.connect(self.widget\_Undo\_2)**

**self.ui.tableWidget.cellClicked.connect(self.info)**

**self.ui.pushButton\_save\_1.clicked.connect(self.insert)**

**self.ui.pushButton\_del.clicked.connect(self.delete)**

**self.ui.pushButton\_Save\_2.clicked.connect(self.redact)**

**myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "12345",database = "kiborgs")**

**cur = myconn.cursor()**

**cur.execute("SELECT \* FROM kiborgs")**

**result = cur.fetchall()**

**for row in result:**

**rowPosition = self.ui.tableWidget.rowCount() # +++**

**self.ui.tableWidget.insertRow(rowPosition)**

**self.ui.tableWidget.setItem(rowPosition,0,QtWidgets.QTableWidgetItem(row[1]))**

**myconn.close()**

**def redact(self):**

**if self.ui.spinBox.text!='0' and self.ui.lineName\_2.text!='' and self.ui.textInfo\_2.toPlainText()!='':**

**myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "12345",database = "kiborgs")**

**cur = myconn.cursor()**

**x=str(int(self.ui.spinBox.text()))**

**cur.execute("update kiborgs set kiborgs="+str(self.ui.lineName\_2.text())+" where id="+x)**

**cur.execute("update kiborgs set info="+str(self.ui.textInfo\_2.toPlainText())+" where id="+x)**

**myconn.commit()**

**myconn.close()**

**self.ui.stackedWidget.setCurrentIndex(0)**

**self.ui.textInfo\_2.clear()**

**self.ui.lineName\_2.clear()**

**else:**

**from PyQt5.QtWidgets import QMessageBox**

**msg = QMessageBox()**

**msg.setIcon(QMessageBox.Critical)**

**msg.setText("Error")**

**msg.setInformativeText('Для удаления есть отдельная кнопка')**

**msg.setWindowTitle("Error")**

**msg.exec\_()**

**def info(self):**

**myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "12345",database = "kiborgs")**

**cur = myconn.cursor()**

**x=str(int(self.ui.spinBox.text()))**

**cur.execute("SELECT \* FROM kiborgs where id="+x)**

**result = cur.fetchall()**

**for row in result:**

**self.ui.textInfo.setText(row[2])**

**myconn.close()**

**def delete(self):**

**if self.ui.spinBox.text!='':**

**myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "12345",database = "kiborgs")**

**cur=myconn.cursor()**

**x=str(int(self.ui.spinBox.text()))**

**cur.execute("delete FROM kiborgs where id="+str(x))**

**self.ui.spinBox.setValue(0)**

**self.ui.textInfo.clear()**

**myconn.commit()**

**myconn.close()**

**self.ui.tableWidget.setRowCount(0)**

**myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "12345",database = "kiborgs")**

**cur = myconn.cursor()**

**cur.execute("SELECT \* FROM kiborgs")**

**result = cur.fetchall()**

**for row in result:**

**rowPosition = self.ui.tableWidget.rowCount() # +++**

**self.ui.tableWidget.insertRow(rowPosition)**

**self.ui.tableWidget.setItem(rowPosition,0,QtWidgets.QTableWidgetItem(row[1]))**

**myconn.close()**

**def widget\_Add(self):**

**self.ui.stackedWidget.setCurrentIndex(1)**

**def widget\_Undo\_1(self):**

**self.ui.stackedWidget.setCurrentIndex(0)**

**self.ui.textInfo\_1.clear()**

**self.ui.lineName\_1.clear()**

**def insert(self):**

**if self.ui.lineName\_1.text!='' and self.ui.textInfo\_1.toPlainText()!='':**

**myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "12345",database = "kiborgs")**

**cur = myconn.cursor()**

**x=str(int(self.ui.spinBox.text()))**

**cur.execute("INSERT INTO kiborgs (id,kiborgs,info) values "+str((int(self.ui.tableWidget.rowCount())+1,self.ui.lineName\_1.text(),self.ui.textInfo\_1.toPlainText())))**

**myconn.commit()**

**myconn.close()**

**myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "12345",database = "kiborgs")**

**cur = myconn.cursor()**

**cur.execute("SELECT \* FROM kiborgs")**

**result = cur.fetchall()**

**self.ui.tableWidget.setRowCount(0)**

**for row in result:**

**rowPosition = self.ui.tableWidget.rowCount() # +++**

**self.ui.tableWidget.insertRow(rowPosition)**

**self.ui.tableWidget.setItem(rowPosition,0,QtWidgets.QTableWidgetItem(row[1]))**

**myconn.close()**

**self.ui.stackedWidget.setCurrentIndex(0)**

**self.ui.textInfo\_1.clear()**

**self.ui.lineName\_1.clear()**

**else:**

**from PyQt5.QtWidgets import QMessageBox**

**msg = QMessageBox()**

**msg.setIcon(QMessageBox.Critical)**

**msg.setText("Error")**

**msg.setInformativeText('заполните оба поля')**

**msg.setWindowTitle("Error")**

**msg.exec\_()**

**def widget\_Undo\_2(self):**

**self.ui.stackedWidget.setCurrentIndex(0)**

**self.ui.textInfo\_2.clear()**

**self.ui.lineName\_2.clear()**

**def widget\_Red(self):**

**global y**

**myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "12345",database = "kiborgs")**

**cur = myconn.cursor()**

**x=str(int(self.ui.spinBox.text()))**

**cur.execute("SELECT \* FROM kiborgs where id="+x)**

**result = cur.fetchall()**

**for row in result:**

**self.ui.textInfo\_2.setText(row[2])**

**self.ui.lineName\_2.setText(row[1])**

**myconn.close()**

**self.ui.stackedWidget.setCurrentIndex(2)**

**app = QtWidgets.QApplication([])**

**application = mywindow()**

**application.show()**

**sys.exit(app.exec())**

**Код файла DCR9**

**from PyQt5 import QtWidgets**

**from DCR9 import Ui\_MainWindow # импорт нашего сгенерированного файла**

**import sys**

**import mysql.connector**

**y=""**

**class mywindow(QtWidgets.QMainWindow):**

**global y**

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

**super(mywindow, self).\_\_init\_\_()**

**self.ui = Ui\_MainWindow()**

**self.ui.setupUi(self)**

**self.ui.stackedWidget.setCurrentIndex(0)**

**self.ui.pushButton\_add.clicked.connect(self.widget\_Add)**

**self.ui.pushButton\_red.clicked.connect(self.widget\_Red)**

**self.ui.pushButton\_undo\_1.clicked.connect(self.widget\_Undo\_1)**

**self.ui.pushButton\_undo\_2.clicked.connect(self.widget\_Undo\_2)**

**self.ui.tableWidget.cellClicked.connect(self.info)**

**self.ui.pushButton\_save\_1.clicked.connect(self.insert)**

**self.ui.pushButton\_del.clicked.connect(self.delete)**

**self.ui.pushButton\_Save\_2.clicked.connect(self.redact)**

**myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "12345",database = "kiborgs")**

**cur = myconn.cursor()**

**cur.execute("SELECT \* FROM kiborgs")**

**result = cur.fetchall()**

**for row in result:**

**rowPosition = self.ui.tableWidget.rowCount() # +++**

**self.ui.tableWidget.insertRow(rowPosition)**

**self.ui.tableWidget.setItem(rowPosition,0,QtWidgets.QTableWidgetItem(row[1]))**

**myconn.close()**

**def redact(self):**

**if self.ui.spinBox.text!='0' and self.ui.lineName\_2.text!='' and self.ui.textInfo\_2.toPlainText()!='':**

**myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "12345",database = "kiborgs")**

**cur = myconn.cursor()**

**x=str(int(self.ui.spinBox.text()))**

**cur.execute("update kiborgs set kiborgs="+str(self.ui.lineName\_2.text())+" where id="+x)**

**cur.execute("update kiborgs set info="+str(self.ui.textInfo\_2.toPlainText())+" where id="+x)**

**myconn.commit()**

**myconn.close()**

**self.ui.stackedWidget.setCurrentIndex(0)**

**self.ui.textInfo\_2.clear()**

**self.ui.lineName\_2.clear()**

**else:**

**from PyQt5.QtWidgets import QMessageBox**

**msg = QMessageBox()**

**msg.setIcon(QMessageBox.Critical)**

**msg.setText("Error")**

**msg.setInformativeText('Для удаления есть отдельная кнопка')**

**msg.setWindowTitle("Error")**

**msg.exec\_()**

**def info(self):**

**myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "12345",database = "kiborgs")**

**cur = myconn.cursor()**

**x=str(int(self.ui.spinBox.text()))**

**cur.execute("SELECT \* FROM kiborgs where id="+x)**

**result = cur.fetchall()**

**for row in result:**

**self.ui.textInfo.setText(row[2])**

**myconn.close()**

**def delete(self):**

**if self.ui.spinBox.text!='':**

**myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "12345",database = "kiborgs")**

**cur=myconn.cursor()**

**x=str(int(self.ui.spinBox.text()))**

**cur.execute("delete FROM kiborgs where id="+str(x))**

**self.ui.spinBox.setValue(0)**

**self.ui.textInfo.clear()**

**myconn.commit()**

**myconn.close()**

**self.ui.tableWidget.setRowCount(0)**

**myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "12345",database = "kiborgs")**

**cur = myconn.cursor()**

**cur.execute("SELECT \* FROM kiborgs")**

**result = cur.fetchall()**

**for row in result:**

**rowPosition = self.ui.tableWidget.rowCount() # +++**

**self.ui.tableWidget.insertRow(rowPosition)**

**self.ui.tableWidget.setItem(rowPosition,0,QtWidgets.QTableWidgetItem(row[1]))**

**myconn.close()**

**def widget\_Add(self):**

**self.ui.stackedWidget.setCurrentIndex(1)**

**def widget\_Undo\_1(self):**

**self.ui.stackedWidget.setCurrentIndex(0)**

**self.ui.textInfo\_1.clear()**

**self.ui.lineName\_1.clear()**

**def insert(self):**

**if self.ui.lineName\_1.text!='' and self.ui.textInfo\_1.toPlainText()!='':**

**myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "12345",database = "kiborgs")**

**cur = myconn.cursor()**

**x=str(int(self.ui.spinBox.text()))**

**cur.execute("INSERT INTO kiborgs (id,kiborgs,info) values "+str((int(self.ui.tableWidget.rowCount())+1,self.ui.lineName\_1.text(),self.ui.textInfo\_1.toPlainText())))**

**myconn.commit()**

**myconn.close()**

**myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "12345",database = "kiborgs")**

**cur = myconn.cursor()**

**cur.execute("SELECT \* FROM kiborgs")**

**result = cur.fetchall()**

**self.ui.tableWidget.setRowCount(0)**

**for row in result:**

**rowPosition = self.ui.tableWidget.rowCount() # +++**

**self.ui.tableWidget.insertRow(rowPosition)**

**self.ui.tableWidget.setItem(rowPosition,0,QtWidgets.QTableWidgetItem(row[1]))**

**myconn.close()**

**self.ui.stackedWidget.setCurrentIndex(0)**

**self.ui.textInfo\_1.clear()**

**self.ui.lineName\_1.clear()**

**else:**

**from PyQt5.QtWidgets import QMessageBox**

**msg = QMessageBox()**

**msg.setIcon(QMessageBox.Critical)**

**msg.setText("Error")**

**msg.setInformativeText('заполните оба поля')**

**msg.setWindowTitle("Error")**

**msg.exec\_()**

**def widget\_Undo\_2(self):**

**self.ui.stackedWidget.setCurrentIndex(0)**

**self.ui.textInfo\_2.clear()**

**self.ui.lineName\_2.clear()**

**def widget\_Red(self):**

**global y**

**myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "12345",database = "kiborgs")**

**cur = myconn.cursor()**

**x=str(int(self.ui.spinBox.text()))**

**cur.execute("SELECT \* FROM kiborgs where id="+x)**

**result = cur.fetchall()**

**for row in result:**

**self.ui.textInfo\_2.setText(row[2])**

**self.ui.lineName\_2.setText(row[1])**

**myconn.close()**

**self.ui.stackedWidget.setCurrentIndex(2)**

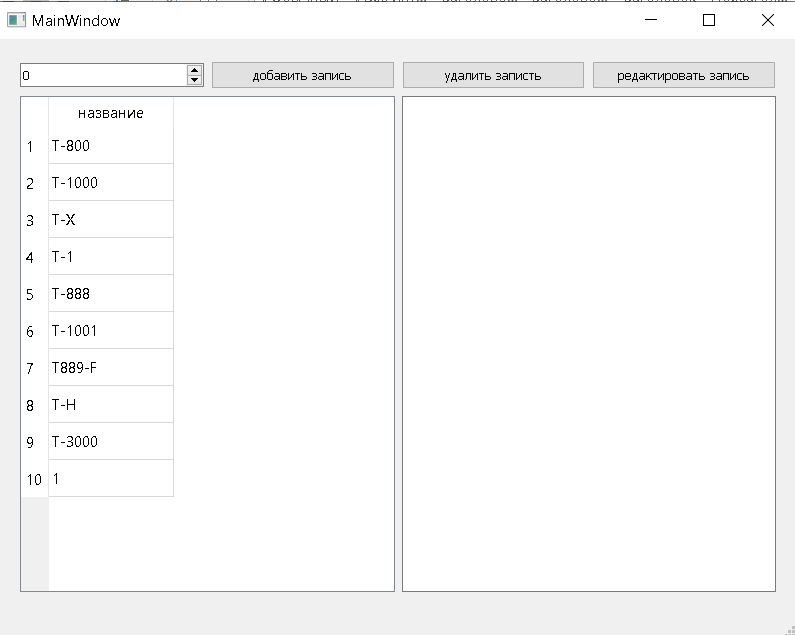
**app = QtWidgets.QApplication([])**

**application = mywindow()**

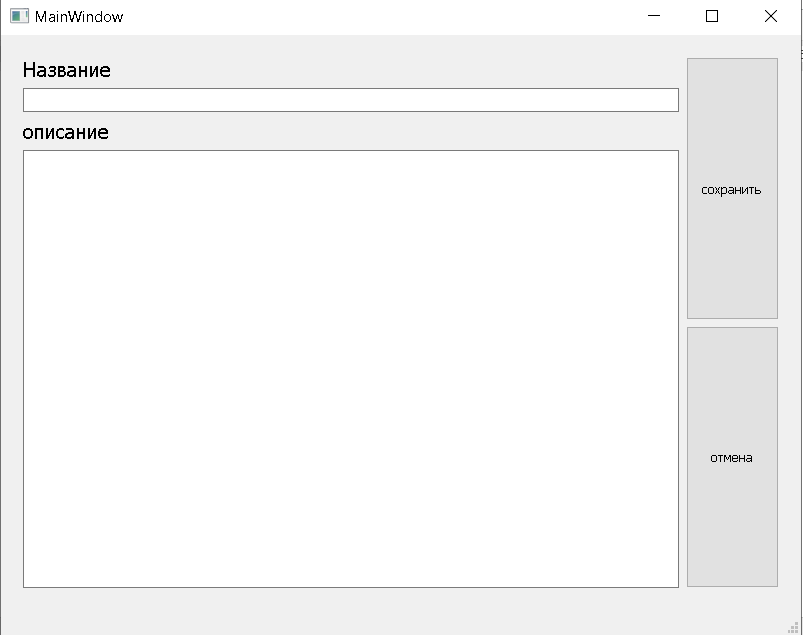
**application.show()**

**sys.exit(app.exec())**

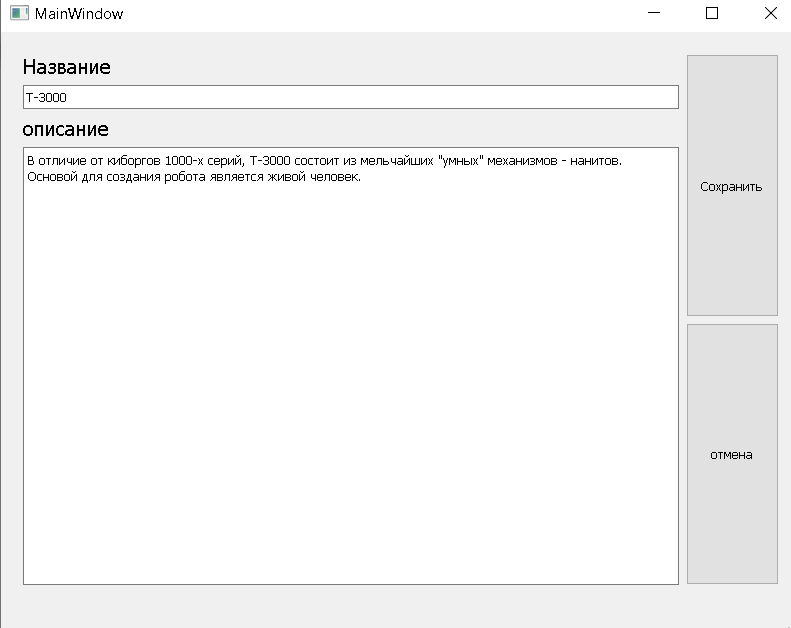
1. **Результат выполнения программы:**

****

**Рисунок 1-Начало программы**

****

**Рисунок 2-окно добавления**

****

**Рисунок 3-окно редактирования**

1. **Вывод:** за время выполнения домашней контрольной работы были закреплены теоретические знания, а также приобретены навыки и умения в соответствии с установленными компетенциями, а именно получены базовые навыки работы с серверами MySQL.