**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Институт Цифры**

**ОТЧЕТ**

**О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №2**

по дисциплине «Разработка современных приложений на Python»

студента 3 курса

**Лободы Максима Алексеевича**

Направление 02.03.02– «Фундаментальная информатика и информационные технологии»

Преподаватель:

ст. пр. Зимин А. И.

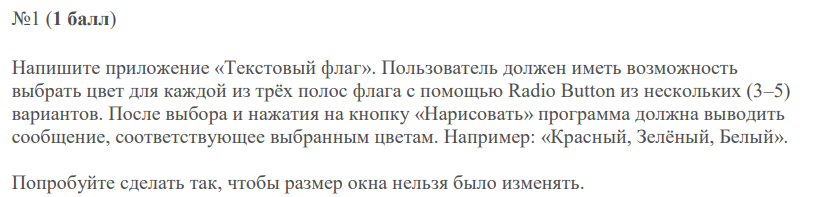
Работа защищена

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

Кемерово 2023

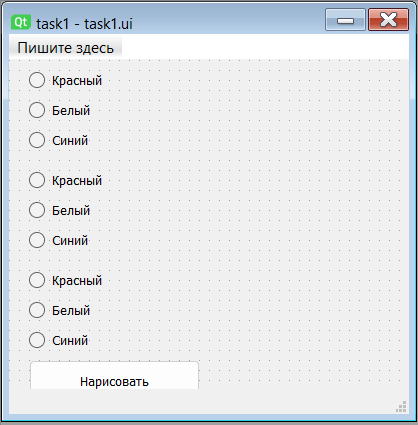
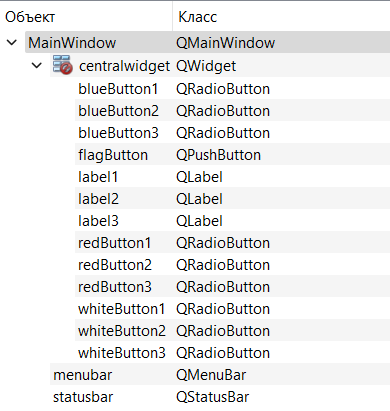
**Основная часть**



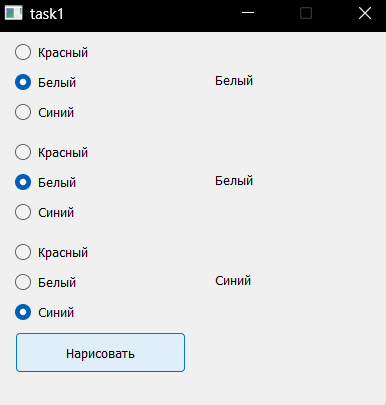
Код приложения

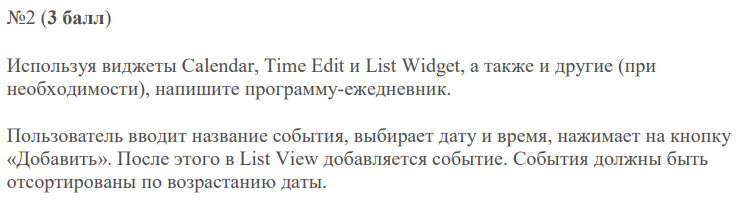
import sys  
  
from PyQt5 import uic  
from PyQt5.QtWidgets import QApplication, QMainWindow, QButtonGroup  
  
  
class MyWidget(QMainWindow):  
 def \_\_init\_\_(self):  
 super().\_\_init\_\_()  
 uic.loadUi('task1.ui', self)  
 self.setFixedSize(400, 380)  
  
 self.group1 = QButtonGroup(self)  
 self.group2 = QButtonGroup(self)  
 self.group3 = QButtonGroup(self)  
  
 self.group1.addButton(self.redButton1)  
 self.group1.addButton(self.whiteButton1)  
 self.group1.addButton(self.blueButton1)  
  
 self.group2.addButton(self.redButton2)  
 self.group2.addButton(self.whiteButton2)  
 self.group2.addButton(self.blueButton2)  
  
 self.group3.addButton(self.redButton3)  
 self.group3.addButton(self.whiteButton3)  
 self.group3.addButton(self.blueButton3)  
  
 self.flagButton.clicked.connect(self.updateLabels)  
  
 def updateLabels(self):  
 color1 = self.getSelectedColour(self.redButton1, self.whiteButton1, self.blueButton1)  
 color2 = self.getSelectedColour(self.redButton2, self.whiteButton2, self.blueButton2)  
 color3 = self.getSelectedColour(self.redButton3, self.whiteButton3, self.blueButton3)  
  
 self.label1.setText(color1)  
 self.label2.setText(color2)  
 self.label3.setText(color3)  
  
 def getSelectedColour(self, red, white, blue):  
 if red.isChecked():  
 return 'Красный'  
 elif white.isChecked():  
 return 'Белый'  
 elif blue.isChecked():  
 return 'Синий'  
 else:  
 return 'Без цвета'  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app = QApplication(sys.argv)  
 ex = MyWidget()  
 ex.show()  
 sys.exit(app.exec\_())

Интерфейс

Демонстрация работы

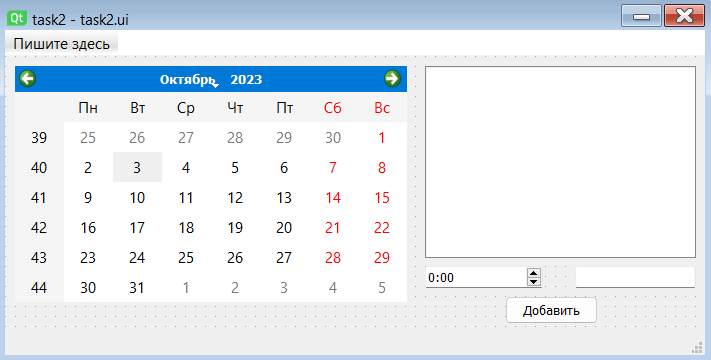


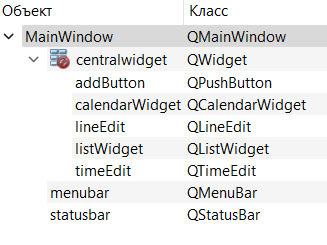


Код приложения

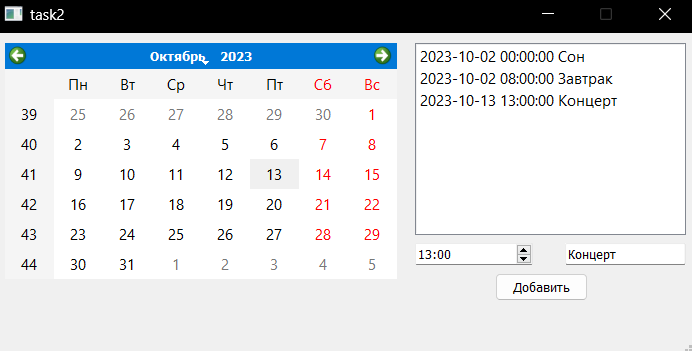
import sys  
  
from PyQt5 import uic  
from PyQt5.QtWidgets import QApplication, QMainWindow  
  
  
class MyWidget(QMainWindow):  
 def \_\_init\_\_(self):  
 super().\_\_init\_\_()  
 uic.loadUi('task2.ui', self)  
 self.setFixedSize(700, 325)  
  
 self.addButton.clicked.connect(self.addRecord)  
  
 def addRecord(self):  
 text = self.lineEdit.text()  
 date = self.calendarWidget.selectedDate().toPyDate()  
 time = self.timeEdit.time().toPyTime()  
  
 self.listWidget.addItem((f"{date} {time} {text}"))  
  
  
def except\_hook(cls, exception, traceback):  
 sys.\_\_excepthook\_\_(cls, exception, traceback)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app = QApplication(sys.argv)  
 ex = MyWidget()  
 ex.show()  
 sys.excepthook = except\_hook  
 sys.exit(app.exec\_())

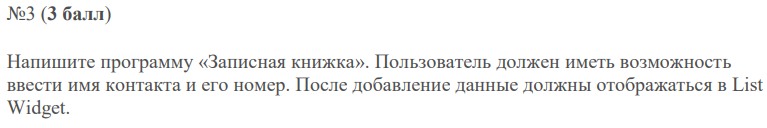
Интерфейс





Демонстрация работы

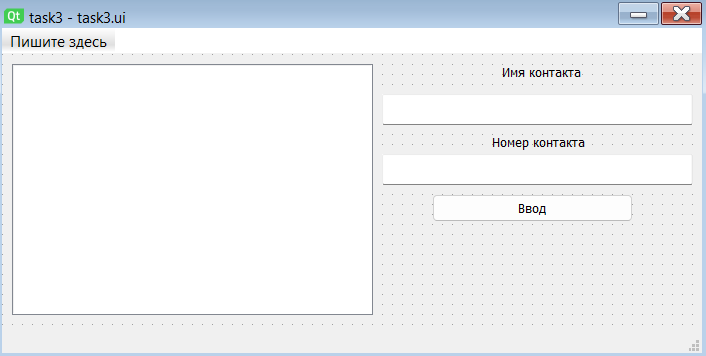


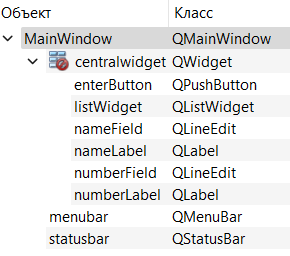


Код приложения

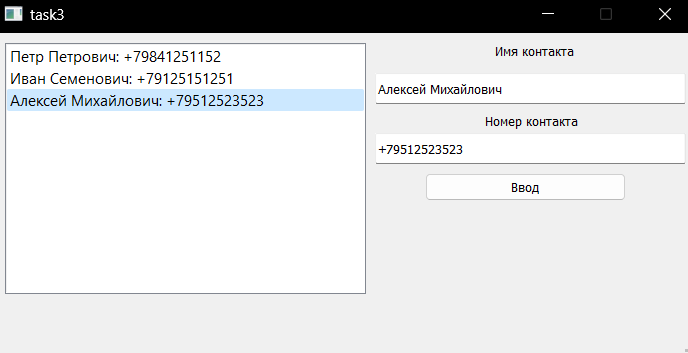
import sys  
  
from PyQt5 import uic  
from PyQt5.QtWidgets import QApplication, QMainWindow  
  
  
class MyWidget(QMainWindow):  
 def \_\_init\_\_(self):  
 super().\_\_init\_\_()  
 uic.loadUi('task3.ui', self)  
 self.setFixedSize(700, 325)  
 self.enterButton.clicked.connect(self.addRecord)  
  
 def addRecord(self):  
 name = self.nameField.text()  
 number = self.numberField.text()  
  
 self.listWidget.addItem((f"{name}: {number}"))  
  
def except\_hook(cls, exception, traceback):  
 sys.\_\_excepthook\_\_(cls, exception, traceback)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app = QApplication(sys.argv)  
 ex = MyWidget()  
 ex.show()  
 sys.excepthook = except\_hook  
 sys.exit(app.exec\_())

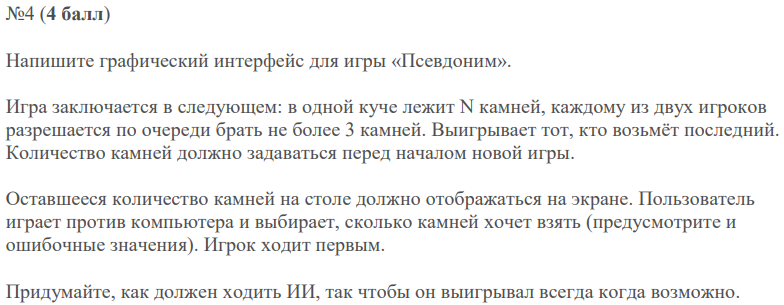
Интерфейс





Демонстрация работы

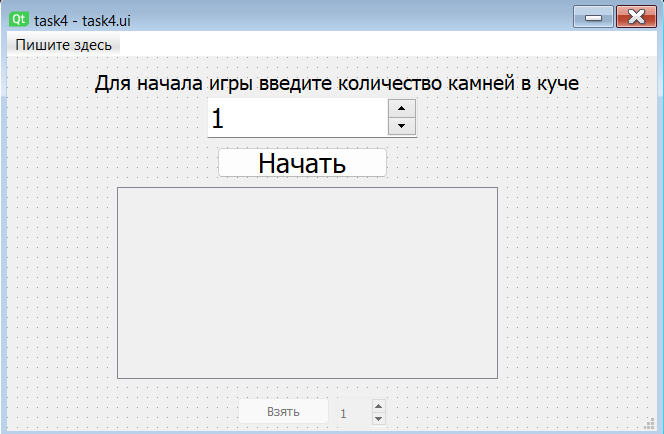


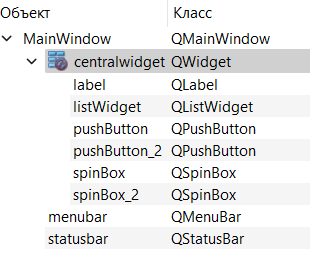


Код приложения

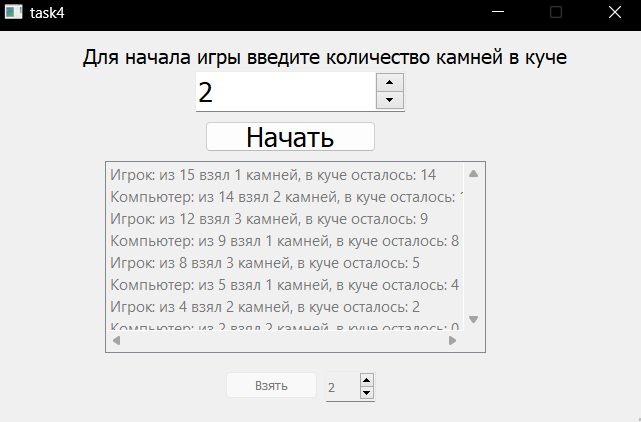
import sys  
  
from PyQt5 import uic  
from PyQt5.QtWidgets import QApplication, QMainWindow  
  
  
class MyWidget(QMainWindow):  
 def \_\_init\_\_(self):  
 super().\_\_init\_\_()  
 uic.loadUi('task4.ui', self)  
 self.setFixedSize(650, 400)  
 self.pushButton.clicked.connect(self.start)  
 self.pushButton\_2.clicked.connect(self.user)  
  
 def start(self):  
 self.pushButton.setEnabled(False)  
 self.spinBox.setEnabled(False)  
 self.pushButton\_2.setEnabled(True)  
 self.spinBox\_2.setEnabled(True)  
 self.listWidget.setEnabled(True)  
 self.listWidget.clear()  
  
 def take\_a\_rock(self, sender, num):  
 prev = self.spinBox.value()  
 if self.spinBox.value() - num <= 0:  
 self.listWidget.addItem(f"{sender}: из {prev} взял {num} камней, в куче осталось: 0")  
 self.listWidget.addItem(f"{sender} побеждает!")  
 self.pushButton.setEnabled(True)  
 self.spinBox.setEnabled(True)  
 self.pushButton\_2.setEnabled(False)  
 self.spinBox\_2.setEnabled(False)  
 self.listWidget.setEnabled(False)  
 else:  
 self.spinBox.setValue(self.spinBox.value() - num)  
 self.listWidget.addItem(f"{sender}: из {prev} взял {num} камней, в куче осталось: {self.spinBox.value()}")  
  
 def user(self):  
 print(self.spinBox\_2.value())  
 self.take\_a\_rock(str("Игрок"), self.spinBox\_2.value())  
 self.computer()  
  
 def computer(self):  
 if not self.spinBox.isEnabled():  
 com\_num = self.spinBox.value() % 4  
 self.take\_a\_rock(str("Компьютер"), com\_num)  
  
  
def except\_hook(cls, exception, traceback):  
 sys.\_\_excepthook\_\_(cls, exception, traceback)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app = QApplication(sys.argv)  
 ex = MyWidget()  
 ex.show()  
 sys.excepthook = except\_hook  
 sys.exit(app.exec\_())

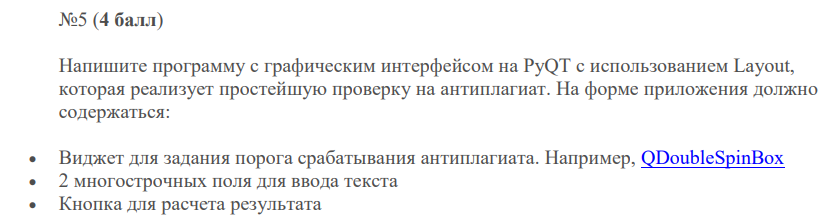
Интерфейс

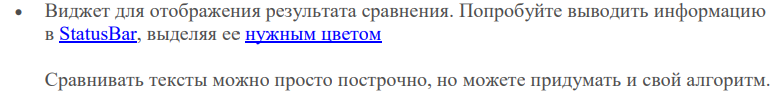




Демонстрация работы



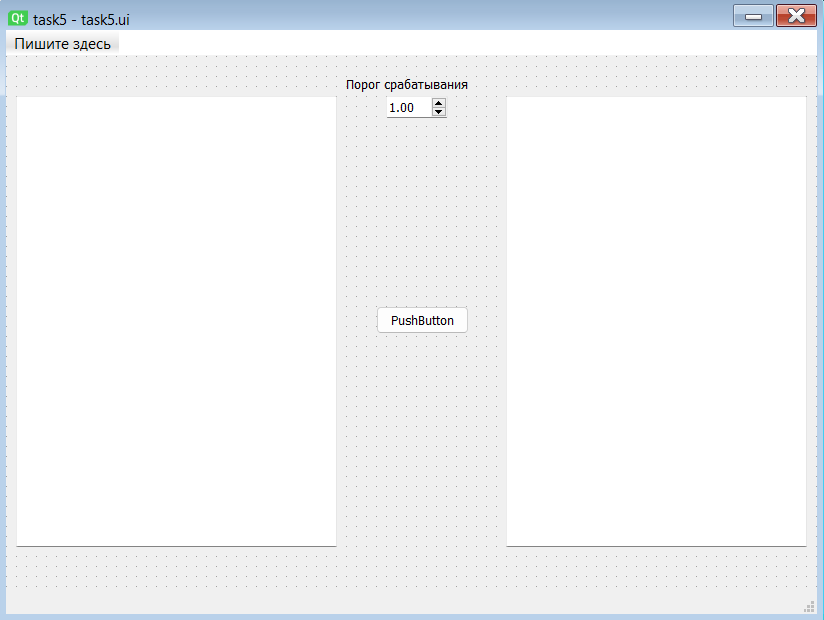


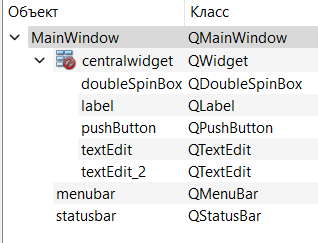


Код приложения

import difflib as df  
import sys  
from PyQt5 import uic  
from PyQt5.QtWidgets import QApplication, QMainWindow, QStatusBar  
  
  
class MyWidget(QMainWindow):  
 def \_\_init\_\_(self):  
 super().\_\_init\_\_()  
 uic.loadUi('task5.ui', self)  
 self.setFixedSize(811, 533)  
 self.statusBar = QStatusBar()  
 self.setStatusBar(self.statusBar)  
 self.pushButton.clicked.connect(self.compare)  
  
 def compare(self):  
 str1 = self.textEdit.toPlainText()  
 str2 = self.textEdit\_2.toPlainText()  
  
 y = df.SequenceMatcher(None, str1.lower(), str2.lower()).ratio() \* 100  
 if float(self.doubleSpinBox.text()) >= round(y, 2):  
 self.statusBar.showMessage(f'Тексты совпадают на {round(y, 2)}%')  
 self.statusBar.setStyleSheet('background-color: green')  
 else:  
 self.statusBar.showMessage(f'Тексты совпадают на {round(y, 2)}%')  
 self.statusBar.setStyleSheet('background-color: red')  
  
  
def except\_hook(cls, exception, traceback):  
 sys.\_\_excepthook\_\_(cls, exception, traceback)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app = QApplication(sys.argv)  
 ex = MyWidget()  
 ex.show()  
 sys.excepthook = except\_hook  
 sys.exit(app.exec\_())

Интерфейс





Демонстрация работы

