

Міністерство освіти і науки України
**Прикарпатський національний університет
імені В.Стефаника**

*Факультет математики та інформатики
Кафедра інформаційних технологій*

Людинно-машинна взаємодія

Лабораторна робота № 8

Тема: Робота з контейнерами в середовищі Qt Creator: QList, QLinkedList

Варіант 2

Виконав: Гук Д.П.
Група ПІЗ-31
Дата: 12 листопада 2023 р.
Викладач: Пікуляк М.В.

Івано-Франківськ – 2023

Мета роботи:

Отримати навички роботи з файлами в ІСР “Qt-Creator”.

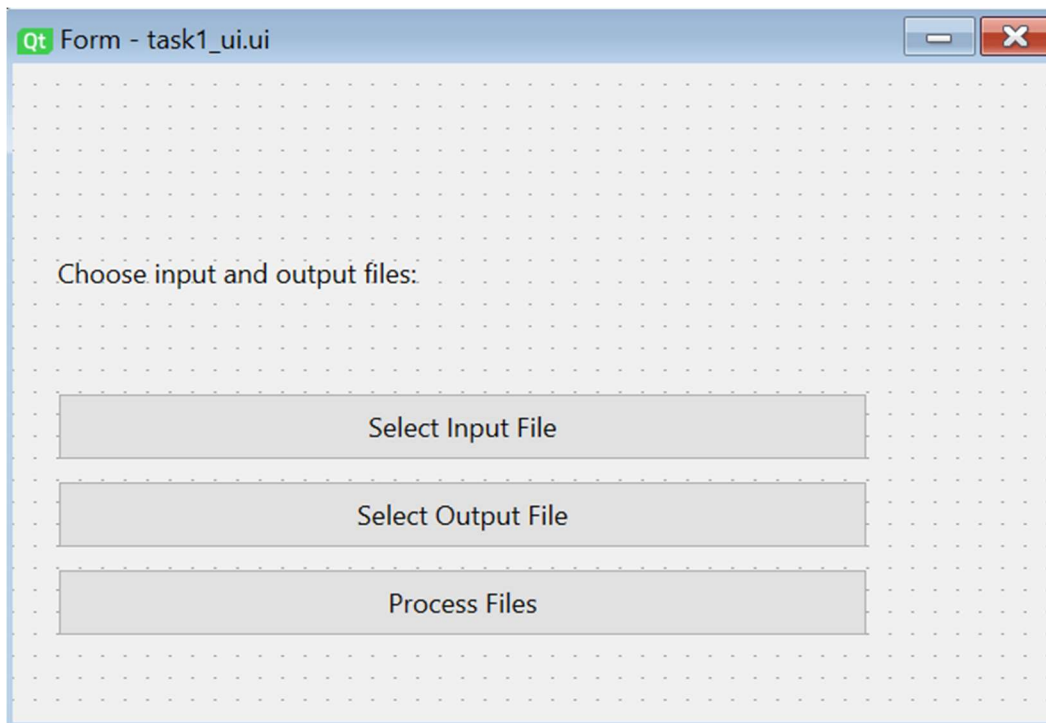
Завдання для виконання :

Завдання 1. Дано файл *f*, компоненти якого є цілими числами. Запишіть в файл *g* найбільше значення перших п’яти компонентів файлу *f*, потім – таких п’яти компонентів і т.д. Якщо в останній групі виявиться менше п’яти компонентів, то останній компонент файлу *g* має дорівнювати найбільшому з компонентів файлу *f*, що утворюють останню (неповну) групу.

Завдання 2. Дано символічні файли *f1* і *f2*. Перепишіть зі збереженням порядку проходження компоненти файлу *f1* в файл *f2*, а компоненти файлу *f2* – в файл *f1*. Використовуйте допоміжний файл *h*.

Тексти скриптів і зображення діалогових вікон QtCreator з виконаними завданнями :

Завдання №1:



```
import sys
from PyQt6 import QtCore, QtGui, QtWidgets
```

```
from PyQt6.QtWidgets import QFileDialog

class FileProcessor:
    def __init__(self, ui):
        self.input_filename = None
        self.output_filename = None

        self.ui = ui
        self.setup_connections()

    def setup_connections(self):
        self.ui.pushButton.clicked.connect(self.get_input_file)
        self.ui.pushButton_2.clicked.connect(self.get_output_file)
        self.ui.pushButton_3.clicked.connect(self.process_files)

    def get_input_file(self):
        self.input_filename, _ = QFileDialog.getOpenFileName(None, 'Select Input File')
        if self.input_filename:
            self.ui.label.setText(f'Input File: {self.input_filename}')

    def get_output_file(self):
        self.output_filename, _ = QFileDialog.getSaveFileName(None, 'Select Output File')
        if self.output_filename:
            self.ui.label.setText(f'Output File: {self.output_filename}')

    def process_files(self):
```

```

        if not self.input_filename or not
self.output_filename:
            return

        try:
            with open(self.input_filename, 'r') as
file:
                numbers = [int(line) for line in
file.readlines()]
                chunk_size = 5
                with open(self.output_filename, 'w') as
output_file:
                    while numbers:
                        chunk =
sorted(numbers[:chunk_size], reverse=True)
                        for num in chunk:
                            output_file.write(str(num)
+ '\n')
                            numbers = numbers[chunk_size:]
            except FileNotFoundError:
                self.ui.label.setText('File not found.
Please select valid files.')
                return

                self.ui.label.setText('Files processed
successfully.')

class Ui_Form(object):
    def setupUi(self, Form):
        Form.setObjectName("Form")
        Form.resize(400, 300)
        self.label = QtWidgets.QLabel(parent=Form)
        self.label.setGeometry(QtCore.QRect(20, 70,
171, 31))
        self.label.setObjectName("label")

```

```

        self.pushButton =
QtWidgets.QPushButton(parent=Form)
        self.pushButton.setGeometry(QtCore.QRect(20,
150, 360, 31))
        self.pushButton.setObjectName("pushButton")
        self.pushButton_2 =
QtWidgets.QPushButton(parent=Form)
        self.pushButton_2.setGeometry(QtCore.QRect(20,
190, 360, 31))
        self.pushButton_2.setObjectName("pushButton_2")
        self.pushButton_3 =
QtWidgets.QPushButton(parent=Form)
        self.pushButton_3.setGeometry(QtCore.QRect(20,
230, 360, 31))
        self.pushButton_3.setObjectName("pushButton_3")

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

    def retranslateUi(self, Form):
        _translate = QtCore.QCoreApplication.translate
        Form.setWindowTitle(_translate("Form", "Task
1"))
        self.label.setText(_translate("Form", "Choose
input and output files: "))
        self.pushButton.setText(_translate("Form",
"Select Input File"))
        self.pushButton_2.setText(_translate("Form",
"Select Output File"))
        self.pushButton_3.setText(_translate("Form",
"Process Files"))

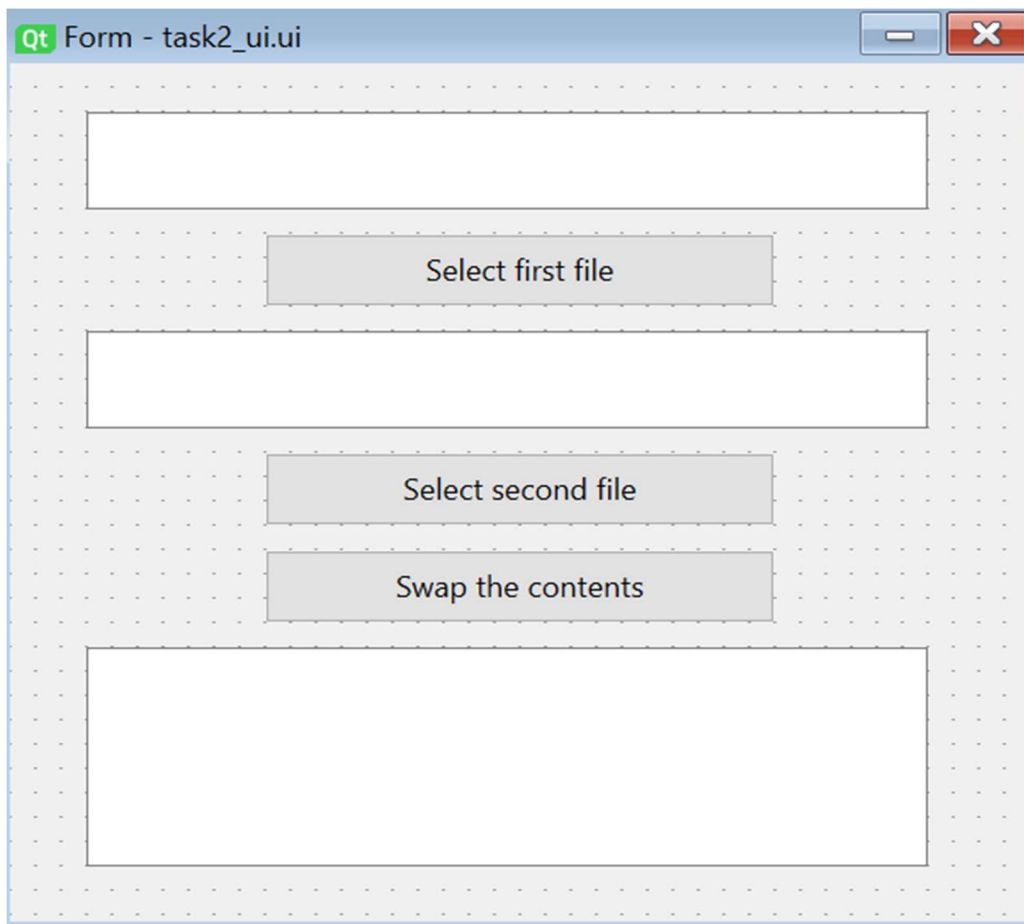
if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    Form = QtWidgets.QWidget()

```

```
ui = Ui_Form()
ui.setupUi(Form)
Form.show()

processor = FileProcessor(ui)
sys.exit(app.exec())
```

Завдання №2:



```
from PyQt6 import QtCore, QtGui, QtWidgets
from PyQt6.QtWidgets import QFileDialog
import shutil

class FileSwapper:
```

```

def __init__(self, ui):
    self.ui = ui
    self.setup_connections()

def setup_connections(self):

self.ui.pushButton.clicked.connect(self.select_first_file)

self.ui.pushButton_2.clicked.connect(self.select_second_file)

self.ui.pushButton_3.clicked.connect(self.swap_contents)

    def select_first_file(self):
        file_path, _ =
QFileDialog.getOpenFileName(None, "Select first file",
"", "Text Files (*.txt);;All Files (*)")
        self.ui.lineEdit.setText(file_path)

    def select_second_file(self):
        file_path, _ =
QFileDialog.getOpenFileName(None, "Select second file",
"", "Text Files (*.txt);;All Files (*)")
        self.ui.lineEdit_2.setText(file_path)

    def swap_contents(self):
        first_file_path = self.ui.lineEdit.text()
        second_file_path = self.ui.lineEdit_2.text()

        if first_file_path == "" or second_file_path ==
"":
            self.ui.lineEdit_3.setText("Please select
both files.")

```

```

        return

    try:
        temp_file_path = "temp.txt"
        shutil.copy(first_file_path,
temp_file_path)
        shutil.copy(second_file_path,
first_file_path)
        shutil.copy(temp_file_path,
second_file_path)
        self.ui.lineEdit_3.setText("Files swapped
successfully.")

    except Exception as e:
        self.ui.lineEdit_3.setText(f"Error: {e}")

class Ui_Form(object):
    def setupUi(self, Form):
        Form.setObjectName("Form")
        Form.resize(400, 353)
        self.lineEdit =
QtWidgets.QLineEdit(parent=Form)
        self.lineEdit.setGeometry(QtCore.QRect(30, 20,
330, 40))
        self.lineEdit.setObjectName("lineEdit")
        self.pushButton =
QtWidgets.QPushButton(parent=Form)
        self.pushButton.setGeometry(QtCore.QRect(100,
70, 200, 30))
        self.pushButton.setObjectName("pushButton")
        self.pushButton_2 =
QtWidgets.QPushButton(parent=Form)
        self.pushButton_2.setGeometry(QtCore.QRect(100,
160, 200, 30))
        self.pushButton_2.setObjectName("pushButton_2")

```



```

        self.lineEdit_2 =
QtWidgets.QLineEdit(parent=Form)
        self.lineEdit_2.setGeometry(QtCore.QRect(30,
110, 330, 40))
        self.lineEdit_2.setObjectName("lineEdit_2")
        self.pushButton_3 =
QtWidgets.QPushButton(parent=Form)
        self.pushButton_3.setGeometry(QtCore.QRect(100,
200, 200, 30))
        self.pushButton_3.setObjectName("pushButton_3")
        self.lineEdit_3 =
QtWidgets.QLineEdit(parent=Form)
        self.lineEdit_3.setGeometry(QtCore.QRect(30,
240, 330, 90))
        self.lineEdit_3.setObjectName("lineEdit_3")

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

    def retranslateUi(self, Form):
        _translate = QtCore.QCoreApplication.translate
        Form.setWindowTitle(_translate("Form", "Task
2"))
        self.pushButton.setText(_translate("Form",
"Select first file"))
        self.pushButton_2.setText(_translate("Form",
"Select second file"))
        self.pushButton_3.setText(_translate("Form",
"Swap the contents"))

if __name__ == "__main__":
    app = QtWidgets.QApplication([])
    Form = QtWidgets.QWidget()
    ui = Ui_Form()
    ui.setupUi(Form)

```

```
Form.show()

file_swapper = FileSwapper(ui) # Initialize the
FileSwapper passing the Ui_Form instance
app.exec()
```

Скрін-шоти виконання завдань лабораторної роботи :

Завдання №1 :

1	10
2	20
3	30
4	40
5	50

Task 1

Choose input and output files:

Select Input File

Select Output File

Process Files

Task 1



Input File: C:/Users/dmytr/PycharmProjects/pythonProject/file1.txt

Select Input File

Select Output File

Process Files

Task 1



Output File: C:/Users/dmytr/PycharmProjects/pythonProject/file2.txt

Select Input File

Select Output File

Process Files

Task 1

Files processed successfully.

Select Input File

Select Output File

Process Files

1	50
2	40
3	30
4	20
5	10
6	

Завдання №2:

1	Sample Text 1 Line 1
2	Sample Text 1 Line 1
3	Sample Text 1 Line 1
4	Sample Text 1 Line 1
5	Sample Text 1 Line 1

```
1 Sample Text 2 Line 2
2 Sample Text 2 Line 2
3 Sample Text 2 Line 2
4 Sample Text 2 Line 2
5 Sample Text 2 Line 2
```

Task 2

C:/Users/dmytr/PycharmProjects/pythonProject/newfile1.txt

Select first file

C:/Users/dmytr/PycharmProjects/pythonProject/newfile2.txt

Select second file

Swap the contents

Files swapped successfully.

```
1 Sample Text 2 Line 2
2 Sample Text 2 Line 2
3 Sample Text 2 Line 2
4 Sample Text 2 Line 2
5 Sample Text 2 Line 2
```

1	Sample Text 1 Line 1
2	Sample Text 1 Line 1
3	Sample Text 1 Line 1
4	Sample Text 1 Line 1
5	Sample Text 1 Line 1