Міністерство освіти і науки України

**Прикарпатський національний університет**

**імені В.Стефаника**

*Факультет математики та інформатики*

*Кафедра інформаційних технологій*

*Людино-машинна взаємодія*

Лабораторна робота № 9

Тема: РРобота с контейнерами в середовищі Qt Creator: QMap, QSet

Варіант: 14

Виконав: ***Пикало М.І***

Група ІПЗ-23

Дата: 23 листопада2024р.

Викладач: Пікуляк М.В.

Івано-Франківськ - 2024

**Завдання для виконання**

**Варіант 2**

Завдання 1. Розробіть графічний додаток mapStudent. Заздалегідь заповніть

словник 10 студентами свого потоку та назвою їх групи. При запуску програми

їх список повинен бути відображений у віджеті QTextEdit. Додайте 2 віджети

QRadioButton, які будуть здійснювати перемикання між пошуком по ключу і

пошуком за значенням. Також слід додати 2 текстових поля: для пошуку і для

виведення результату.

Також передбачте видалення і додавання записів в довідник, якщо введений

ключ вже існує при додаванні запису, виведіть інформаційне повідомлення.

Завдання 2. Заданий деякий набір товарів. Визначити для кожного товару:

а) які з них є в кожному з n магазинів;

б) які товари є хоча б в одному магазині;

в) яких товарів немає ні в одному магазині.

Завдання 3. Створіть два порожніх списки контактів для зберігання записів.

Заповніть телефонні книги прізвищами абонентів і телефонними номерами.

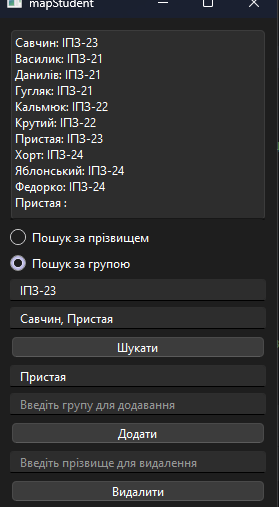
Виведіть в textEdit вміст телефонних книг. Виконайте обмін словників і виведіть

їх на екран. Очистіть словники. Доповніть телефонну книгу новими записами і

змініть один з номері

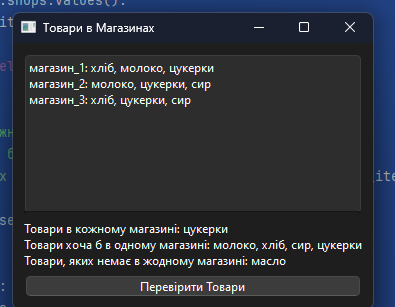
**Завдання 1**

from PySide6.QtWidgets import (  
 QApplication,  
 QWidget,  
 QVBoxLayout,  
 QTextEdit,  
 QRadioButton,  
 QLineEdit,  
 QPushButton,  
 QMessageBox,  
)  
import sys  
  
  
class MapStudentApp(QWidget):  
 def \_\_init\_\_(self) -> None:  
 super().\_\_init\_\_()  
 self.setWindowTitle("mapStudent")  
  
 self.students = {  
 "Савчин": "ІПЗ-23",  
 "Василик": "ІПЗ-21",  
 "Данилів": "ІПЗ-21",  
 "Гугляк": "ІПЗ-21",  
 "Кальмюк": "ІПЗ-22",  
 "Крутий": "ІПЗ-22",  
 "Пристая": "ІПЗ-23",  
 "Хорт": "ІПЗ-24",  
 "Яблонський": "ІПЗ-24",  
 "Федорко": "ІПЗ-24",  
 }  
  
 self.textEdit = QTextEdit(self)  
 self.textEdit.setReadOnly(True)  
  
 self.radioKey = QRadioButton("Пошук за прізвищем")  
 self.radioValue = QRadioButton("Пошук за групою")  
 self.radioKey.setChecked(True)  
  
 self.searchField = QLineEdit(self)  
 self.searchField.setPlaceholderText("Введіть значення для пошуку")  
  
 self.resultField = QLineEdit(self)  
 self.resultField.setReadOnly(True)  
  
 self.searchButton = QPushButton("Шукати", self)  
 self.searchButton.clicked.connect(self.search)  
  
 self.addNameField = QLineEdit(self)  
 self.addNameField.setPlaceholderText("Введіть прізвище для додавання")  
  
 self.addGroupField = QLineEdit(self)  
 self.addGroupField.setPlaceholderText("Введіть групу для додавання")  
  
 self.addButton = QPushButton("Додати", self)  
 self.addButton.clicked.connect(self.add\_student)  
  
 self.deleteNameField = QLineEdit(self)  
 self.deleteNameField.setPlaceholderText("Введіть прізвище для видалення")  
  
 self.deleteButton = QPushButton("Видалити", self)  
 self.deleteButton.clicked.connect(self.delete\_student)  
  
 self.layout = QVBoxLayout()  
 self.layout.addWidget(self.textEdit)  
 self.layout.addWidget(self.radioKey)  
 self.layout.addWidget(self.radioValue)  
 self.layout.addWidget(self.searchField)  
 self.layout.addWidget(self.resultField)  
 self.layout.addWidget(self.searchButton)  
 self.layout.addWidget(self.addNameField)  
 self.layout.addWidget(self.addGroupField)  
 self.layout.addWidget(self.addButton)  
 self.layout.addWidget(self.deleteNameField)  
 self.layout.addWidget(self.deleteButton)  
  
 self.display\_students()  
 self.setLayout(self.layout)  
  
 def display\_students(self) -> None:  
 self.textEdit.clear()  
 for student, group in self.students.items():  
 self.textEdit.append(f"{student}: {group}")  
  
 def search(self) -> None:  
 query = self.searchField.text()  
 if self.radioKey.isChecked():  
 if query in self.students:  
 self.resultField.setText(self.students[query])  
 else:  
 QMessageBox.warning(  
 self, "Увага", f"Студента з прізвищем '{query}' не знайдено."  
 )  
 else:  
 result = [  
 student for student, group in self.students.items() if group == query  
 ]  
 if result:  
 self.resultField.setText(", ".join(result))  
 else:  
 QMessageBox.warning(  
 self, "Увага", f"Студентів з групи '{query}' не знайдено."  
 )  
  
 def add\_student(self) -> None:  
 name = self.addNameField.text()  
 group = self.addGroupField.text()  
 if name in self.students:  
 QMessageBox.warning(  
 self, "Увага", f"Студент з прізвищем '{name}' вже існує."  
 )  
 else:  
 self.students[name] = group  
 self.display\_students()  
 QMessageBox.information(  
 self, "Успіх", f"Студента '{name}' додано до групи '{group}'."  
 )  
  
 def delete\_student(self) -> None:  
 name = self.deleteNameField.text()  
 if name in self.students:  
 group = self.students[name]  
 del self.students[name]  
 self.display\_students()  
 QMessageBox.information(  
 self, "Успіх", f"Студента '{name}' з групи '{group}' видалено."  
 )  
 else:  
 QMessageBox.warning(  
 self, "Увага", f"Студента з прізвищем '{name}' не знайдено."  
 )  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app = QApplication(sys.argv)  
 window = MapStudentApp()  
 window.show()  
 sys.exit(app.exec())



**Завдання 2**

from PySide6.QtWidgets import (  
 QApplication,  
 QWidget,  
 QVBoxLayout,  
 QTextEdit,  
 QPushButton,  
 QLabel,  
)  
import sys  
  
  
class ShopItemsApp(QWidget):  
 def \_\_init\_\_(self) -> None:  
 super().\_\_init\_\_()  
 self.setWindowTitle("Товари в Магазинах")  
  
 self.shops = {  
 "магазин\_1": {"хліб", "молоко", "цукерки"},  
 "магазин\_2": {"молоко", "цукерки", "сир"},  
 "магазин\_3": {"хліб", "цукерки", "сир"},  
 }  
  
 self.full\_item\_set = {"хліб", "молоко", "цукерки", "сир", "масло"}  
  
 self.textEdit = QTextEdit(self)  
 self.textEdit.setReadOnly(True)  
  
 self.resultLabel = QLabel(self)  
  
 self.checkButton = QPushButton("Перевірити Товари", self)  
 self.checkButton.clicked.connect(self.check\_items)  
  
 self.layout = QVBoxLayout()  
 self.layout.addWidget(self.textEdit)  
 self.layout.addWidget(self.resultLabel)  
 self.layout.addWidget(self.checkButton)  
  
 self.display\_shops()  
 self.setLayout(self.layout)  
  
 def display\_shops(self) -> None:  
 self.textEdit.clear()  
 for shop, items in self.shops.items():  
 self.textEdit.append(f"{shop}: {', '.join(items)}")  
  
 def check\_items(self) -> None:  
 common\_items = self.full\_item\_set.copy()  
 for items in self.shops.values():  
 common\_items &= items  
  
 all\_items = set()  
 for items in self.shops.values():  
 all\_items |= items  
  
 missing\_items = self.full\_item\_set - all\_items  
  
 result\_text = (  
 f"Товари в кожному магазині: {', '.join(common\_items)}\n"  
 f"Товари хоча б в одному магазині: {', '.join(all\_items)}\n"  
 f"Товари, яких немає в жодному магазині: {', '.join(missing\_items)}"  
 )  
 self.resultLabel.setText(result\_text)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app = QApplication(sys.argv)  
 window = ShopItemsApp()  
 window.show()  
 sys.exit(app.exec())



**Завдання 3**

from PySide6.QtWidgets import QApplication, QWidget, QVBoxLayout, QTextEdit, QPushButton  
import sys  
  
  
class ContactBookApp(QWidget):  
 def \_\_init\_\_(self) -> None:  
 super().\_\_init\_\_()  
 self.setWindowTitle("Телефонні книги")  
  
 self.contacts1 = {}  
 self.contacts2 = {}  
  
 self.textEdit = QTextEdit(self)  
 self.textEdit.setReadOnly(True)  
  
 self.populateButton = QPushButton("Заповнити", self)  
 self.populateButton.clicked.connect(self.populate\_contacts)  
  
 self.swapButton = QPushButton("Обміняти", self)  
 self.swapButton.clicked.connect(self.swap\_contacts)  
  
 self.clearButton = QPushButton("Очистити", self)  
 self.clearButton.clicked.connect(self.clear\_contacts)  
  
 self.updateButton = QPushButton("Оновити", self)  
 self.updateButton.clicked.connect(self.update\_contacts)  
  
 self.layout = QVBoxLayout()  
 self.layout.addWidget(self.textEdit)  
 self.layout.addWidget(self.populateButton)  
 self.layout.addWidget(self.swapButton)  
 self.layout.addWidget(self.clearButton)  
 self.layout.addWidget(self.updateButton)  
  
 self.setLayout(self.layout)  
  
 def populate\_contacts(self) -> None:  
 self.contacts1 = {  
 "Іванов": "123-456-789",  
 "Петров": "987-654-321",  
 "Сидоров": "555-555-555",  
 }  
 self.contacts2 = {  
 "Кузнєцов": "111-222-333",  
 "Смирнов": "444-555-666",  
 "Попов": "777-888-999",  
 }  
 self.display\_contacts()  
  
 def swap\_contacts(self) -> None:  
 self.contacts1, self.contacts2 = self.contacts2, self.contacts1  
 self.display\_contacts()  
  
 def clear\_contacts(self) -> None:  
 self.contacts1.clear()  
 self.contacts2.clear()  
 self.display\_contacts()  
  
 def update\_contacts(self) -> None:  
 self.contacts1.update({"Новий": "000-111-222", "Старий": "333-444-555"})  
 self.contacts2.update({"Інший": "666-777-888"})  
  
 if "Іванов" in self.contacts1:  
 self.contacts1["Іванов"] = "999-999-999"  
 self.display\_contacts()  
  
 def display\_contacts(self) -> None:  
 self.textEdit.clear()  
 self.textEdit.append("Телефонна книга 1:")  
 for name, number in self.contacts1.items():  
 self.textEdit.append(f"{name}: {number}")  
 self.textEdit.append("\nТелефонна книга 2:")  
 for name, number in self.contacts2.items():  
 self.textEdit.append(f"{name}: {number}")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app = QApplication(sys.argv)  
 window = ContactBookApp()  
 window.show()  
 sys.exit(app.exec())

