Міністерство освіти і науки України

**Прикарпатський національний університет**

**імені В.Стефаника**

*Факультет математики та інформатики*

*Кафедра інформаційних технологій*

*Людино-машинна взаємодія*

Лабораторна робота № 10

Тема: Створення проекту «Калькулятор»

Варіант: 14

Виконав: ***Пикало М.І***

Група ІПЗ-23

Дата: 28 листопада 2024р.

Викладач: Пікуляк М.В.

Івано-Франківськ - 2024

**Завдання для виконання**

**Варіант 2**

Створення проекту «Калькулятор»

import sys  
import math  
from PySide6.QtWidgets import (  
 QApplication,  
 QMainWindow,  
 QLineEdit,  
 QPushButton,  
 QVBoxLayout,  
 QWidget,  
 QGridLayout,  
 QMessageBox,  
)  
from PySide6.QtCore import Qt  
from PySide6.QtGui import QFont  
from collections import deque  
  
  
class Calculator(QMainWindow):  
 def \_\_init\_\_(self) -> None:  
 super().\_\_init\_\_()  
 self.setWindowTitle("Калькулятор")  
 self.setGeometry(500, 500, 300, 350)  
 self.memory = 0  
 self.stack = deque()  
  
 self.lineEdit = QLineEdit(self)  
 self.lineEdit\_2 = QLineEdit(self)  
  
 layout = QVBoxLayout()  
 grid\_layout = QGridLayout()  
  
 self.lineEdit.setMaxLength(10)  
 self.lineEdit.setAlignment(Qt.AlignRight)  
 self.lineEdit.setFont(QFont("", 16))  
 self.lineEdit\_2.setAlignment(Qt.AlignRight)  
 self.lineEdit\_2.setFont(QFont("", 16))  
 self.lineEdit\_2.setEnabled(False)  
  
 buttons = [  
 ("7", 0, 0),  
 ("8", 0, 1),  
 ("9", 0, 2),  
 ("/", 0, 3),  
 ("4", 1, 0),  
 ("5", 1, 1),  
 ("6", 1, 2),  
 ("\*", 1, 3),  
 ("1", 2, 0),  
 ("2", 2, 1),  
 ("3", 2, 2),  
 ("-", 2, 3),  
 ("0", 3, 0),  
 (".", 3, 1),  
 ("+", 3, 2),  
 ("=", 3, 3),  
 ("C", 4, 0),  
 ("±", 4, 1),  
 ("M+", 4, 2),  
 ("M-", 4, 3),  
 ("MC", 5, 0),  
 ("MR", 5, 1),  
 ("MS", 5, 2),  
 ("⌫", 5, 3),  
 ("cos", 6, 0),  
 ("sin", 6, 1),  
 ("tg", 6, 2),  
 ("x²", 6, 3),  
 ("√", 7, 0),  
 ("1/x", 7, 1),  
 ]  
  
 for text, row, col in buttons:  
 button = QPushButton(text)  
 button.clicked.connect(lambda checked, t=text: self.on\_button\_click(t))  
 grid\_layout.addWidget(button, row, col)  
  
 layout.addWidget(self.lineEdit)  
 layout.addWidget(self.lineEdit\_2)  
 layout.addLayout(grid\_layout)  
  
 central\_widget = QWidget()  
 central\_widget.setLayout(layout)  
 self.setCentralWidget(central\_widget)  
  
 def on\_button\_click(self, char) -> None:  
 try:  
 if char.isdigit():  
 self.lineEdit.insert(char)  
 elif char in ["+", "-", "\*", "/"]:  
 self.calculate(char)  
 elif char == "C":  
 self.clear\_all()  
 elif char == "=":  
 self.equals()  
 elif char == ".":  
 self.add\_decimal()  
 elif char == "±":  
 self.change\_sign()  
 elif char == "M+":  
 self.memory += float(self.lineEdit.text() or "0")  
 elif char == "M-":  
 self.memory -= float(self.lineEdit.text() or "0")  
 elif char == "MR":  
 self.lineEdit.setText(str(self.memory))  
 elif char == "MC":  
 self.memory = 0  
 elif char == "MS":  
 self.memory = float(  
 self.lineEdit.text() or "0"  
 )  
 elif char == "⌫":  
 self.lineEdit.backspace()  
 elif char == "cos":  
 self.lineEdit.setText(  
 str(  
 round(  
 math.cos(math.radians(float(self.lineEdit.text() or "0"))),  
 6,  
 )  
 )  
 )  
 elif char == "sin":  
 self.lineEdit.setText(  
 str(  
 round(  
 math.sin(math.radians(float(self.lineEdit.text() or "0"))),  
 6,  
 )  
 )  
 )  
 elif char == "tg":  
 self.lineEdit.setText(  
 str(  
 round(  
 math.tan(math.radians(float(self.lineEdit.text() or "0"))),  
 6,  
 )  
 )  
 )  
 elif char == "x²":  
 self.lineEdit.setText(str(float(self.lineEdit.text() or "0") \*\* 2))  
 elif char == "√":  
 self.lineEdit.setText(  
 str(math.sqrt(float(self.lineEdit.text() or "0")))  
 )  
 elif char == "1/x":  
 self.lineEdit.setText(str(1 / float(self.lineEdit.text() or "0")))  
 except Exception as e:  
 QMessageBox.critical(self, "Error", str(e))  
  
 def calculate(self, operator) -> None:  
 if self.lineEdit.text():  
 self.stack.append(self.lineEdit.text())  
 self.stack.append(operator)  
 self.lineEdit.clear()  
 self.lineEdit\_2.setText(" ".join(self.stack))  
  
 def equals(self) -> None:  
 try:  
 if len(self.stack) == 2 and self.lineEdit.text():  
 self.stack.append(self.lineEdit.text())  
 val2 = float(self.stack.pop())  
 operator = self.stack.pop()  
 val1 = float(self.stack.pop())  
 result = 0  
 if operator == "+":  
 result = val1 + val2  
 elif operator == "-":  
 result = val1 - val2  
 elif operator == "\*":  
 result = val1 \* val2  
 elif operator == "/":  
 result = val1 / val2 if val2 != 0 else val1  
 expression = f"{val1} {operator} {val2} = {result}"  
 self.lineEdit.setText(str(result))  
 self.lineEdit\_2.setText(expression)  
 self.stack.clear()  
 except Exception as e:  
 QMessageBox.critical(self, "Error", str(e))  
  
 def clear\_all(self) -> None:  
 self.lineEdit.clear()  
 self.lineEdit\_2.clear()  
 self.stack.clear()  
  
 def add\_decimal(self) -> None:  
 if "." not in self.lineEdit.text():  
 self.lineEdit.insert(".")  
  
 def change\_sign(self) -> None:  
 text = self.lineEdit.text()  
 if text.startswith("-"):  
 self.lineEdit.setText(text[1:])  
 else:  
 self.lineEdit.setText("-" + text)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app = QApplication(sys.argv)  
 window = Calculator()  
 window.show()  
 sys.exit(app.exec())

