

# Sprawozdanie

## Architektura Systemów Komputerowych

Skład:

Paweł Sternicki 154775

Bartłomiej Nowakowski 150863

Data oddania: 3.04.2017

### Zadanie 3: Aplikacja wykorzystująca technikę tworzenia graficznego interfejsu użytkownika.

#### 1. Cele projektu.

Wykorzystując dowolny język programowania dla komputerów w standardzie PC napisać aplikację wykorzystującą technikę tworzenia graficznego interfejsu użytkownika. Interfejs powinien w sposób interaktywny wykorzystywać do komunikacji z operatorem monitor ekranowy, klawiaturę oraz myszkę.

#### 2. Realizacja.

Aplikacja została zrealizowana za pomocą języka programowania Python 2.7. Zgodnie z sugestią w dokumencie opisującym zadania projektowe napisaliśmy kalkulator z możliwością zmiany skórek, oraz zegarem. Okienko z programem zostało zaprojektowane z wykorzystaniem biblioteki PyQt4.

a. Obsługa myszki oraz klawiatury.

Kalkulator może być obsługiwany za pomocą myszki poprzez klikanie przycisków umieszczonych w okienku programu, oraz za pomocą klawiatury zgodnie z wymaganiami projektowymi. Liczba jest wyświetlana na ekranie kalkulatora, następnie po wciśnięciu jednego z przycisków operatorów (plus, minus, razy, podzielić) liczba z ekranu jest zapisywana do bufora a sam ekran jest czyszczony. Możliwe są operacje wyłącznie na dwóch liczbach. Poza podstawowymi operacjami na liczbach, możliwe jest także pierwiastkowanie liczb i podnoszenie do kwadratu. Umożliwiliśmy także operacje na liczbach zmiennoprzecinkowych.

```
def keyPressEvent(self, event):
    global num
    global newNum
    global opVar
    global operator
    global sumIt
    global sumAll
    char = event.text()
    if char == '0' or char == '1' or char == '2' or char == '3' or char == '4'
or char == '5' \
        or char == '6' or char == '7' or char == '8' or char == '9':
        newNum = int(char)
        setNum = str(newNum)
        if opVar == False:
            self.line.setText(self.line.text() + setNum)
        else:
            self.line.setText(setNum)
            opVar = False
    if char == '+' or char == '-' or char == '*' or char == '/':
        sumIt += 1
```

```

        if sumIt > 1:
            self.Equal()
            num = self.line.text()
            operator = char
            opVar = True
    if char == '=':
        self.Equal()
    if char == '.':
        self.pointClicked()
    if event.key() == QtCore.Qt.Key_Escape:
        self.close()

```

Rys 1: Obsługa klawiatury.

```

zero = QtGui.QPushButton("0", self)
zero.move(50, 205)
zero.resize(35, 30)
one = QtGui.QPushButton("1", self)
one.move(10, 170)
one.resize(35, 30)
two = QtGui.QPushButton("2", self)
two.move(50, 170)
two.resize(35, 30)
three = QtGui.QPushButton("3", self)
three.move(90, 170)
three.resize(35, 30)
four = QtGui.QPushButton("4", self)
four.move(10, 135)
four.resize(35, 30)
five = QtGui.QPushButton("5", self)
five.move(50, 135)
five.resize(35, 30)
six = QtGui.QPushButton("6", self)
six.move(90, 135)
six.resize(35, 30)
seven = QtGui.QPushButton("7", self)
seven.move(10, 100)
seven.resize(35, 30)
eight = QtGui.QPushButton("8", self)
eight.move(50, 100)
eight.resize(35, 30)
nine = QtGui.QPushButton("9", self)
nine.move(90, 100)
nine.resize(35, 30)
point = QtGui.QPushButton(".", self)
point.move(90, 205)
point.resize(35, 30)
point.clicked.connect(self.pointClicked)
div = QtGui.QPushButton("/ ", self)
div.move(130, 100)
div.resize(35, 30)
mult = QtGui.QPushButton("*", self)
mult.move(130, 135)
mult.resize(35, 30)
minus = QtGui.QPushButton("-", self)
minus.move(130, 170)
minus.resize(35, 30)
plus = QtGui.QPushButton("+", self)
plus.move(130, 205)
plus.resize(35, 30)
sqrt = QtGui.QPushButton("sqrt", self)
sqrt.move(170, 100)
sqrt.resize(35, 30)

```

```

sqrt.clicked.connect(self.Sqrt)
squared = QtGui.QPushButton("^2", self)
squared.move(170, 135)
squared.resize(35, 30)
squared.clicked.connect(self.Squared)
equal = QtGui.QPushButton("=", self)
equal.move(170, 170)
equal.resize(35, 65)
equal.clicked.connect(self.Equal)
c = QtGui.QPushButton("C", self)
c.move(145, 60)
c.resize(60, 30)
c.clicked.connect(self.C)
ce = QtGui.QPushButton("CE", self)
ce.move(77, 60)
ce.resize(60, 30)
ce.clicked.connect(self.CE)
back = QtGui.QPushButton("Back", self)
back.move(10, 60)
back.resize(60, 30)
back.clicked.connect(self.Back)

```

Rys 2: Tworzenie przycisków do obsługi myszki.

## b. Konfigurowanie elementów grafiki.

Program umożliwia użytkownikowi w pewnym stopniu konfigurowanie elementów grafiki poprzez zmianę skórki (kolor tła programu). Skórki możliwe do ustawienia to: niebieski, biały, zielony, czerwony oraz szary. Zmiana skórki następuje poprzez wybór odpowiedniego elementu w menu Skins.

```

def setBlue(self):
    palette = QtGui.QPalette()
    palette.setColor(QtGui.QPalette.Background, QtCore.Qt.darkBlue)
    self.setPalette(palette)
def setWhite(self):
    palette = QtGui.QPalette()
    palette.setColor(QtGui.QPalette.Background, QtCore.Qt.white)
    self.setPalette(palette)
def setGreen(self):
    palette = QtGui.QPalette()
    palette.setColor(QtGui.QPalette.Background, QtCore.Qt.darkGreen)
    self.setPalette(palette)
def setRed(self):
    palette = QtGui.QPalette()
    palette.setColor(QtGui.QPalette.Background, QtCore.Qt.darkRed)
    self.setPalette(palette)
def setGray(self):
    palette = QtGui.QPalette()
    palette.setColor(QtGui.QPalette.Background, QtCore.Qt.darkGray)
    self.setPalette(palette)

```

Rys 3: Zdefiniowanie skórki w kodzie.

### c. Wyświetlanie czasu.

Kalkulator został także zaopatrzony w zegar cyfrowy umieszczony na dole okienka z programem. Zegar został zrealizowany poprzez zastosowanie odpowiedniego widgetu z biblioteki PyQt.

```
def Time(self):  
    self.lcd.display(strftime("%H" + ":" + "%M"))  
def digitClock(self):  
    self.timer = QtCore.QTimer(self)  
    self.timer.timeout.connect(self.Time)  
    self.timer.start(1000)  
    self.lcd = QtGui.QLCDNumber(self)  
    self.lcd.move(50, 240)
```

Rys 4: Kod odpowiadający za wyświetlanie zegara.

## 3. Ocena pracy.

W naszej ocenie zrealizowaliśmy wszystkie założenia projektowe. Kalkulator obsługiwany być może zarówno za pomocą myszki jak i klawiatury. Użytkownik ma także możliwość konfigurowania elementów graficznych programu poprzez zmianę skórek. Pewną wadą może być nieuporządkowany kod programu, oraz to, że nie ma możliwości zmiany sposobu wyświetlania czasu na zegar analogowy.

