

# Laboratorium ASK

## Zadanie 1

**Michał Kopczyński 184817 KSD ACR**

**Maksymilian Terebus 181595 KSD ACR**

**Aplikacja została napisana w języku Python używając biblioteki PyQt5.**

### **1) Sformułowanie problemu:**

Jako nasze zadanie postawiliśmy sobie stworzenie kalkulatora z zegarem. Jest to kalkulator prosty, który pozwala zarówno na wpisywanie do “text boxa” równania oraz po kliknięciu przycisku Enter uzyskujemy wynik, co wskazuje na to, że komunikacja programu z klawiaturą oraz ekranem działa poprawnie. Również możliwe jest operowanie na przyciskach, które pozwalają na uzyskanie takiego samego efektu.

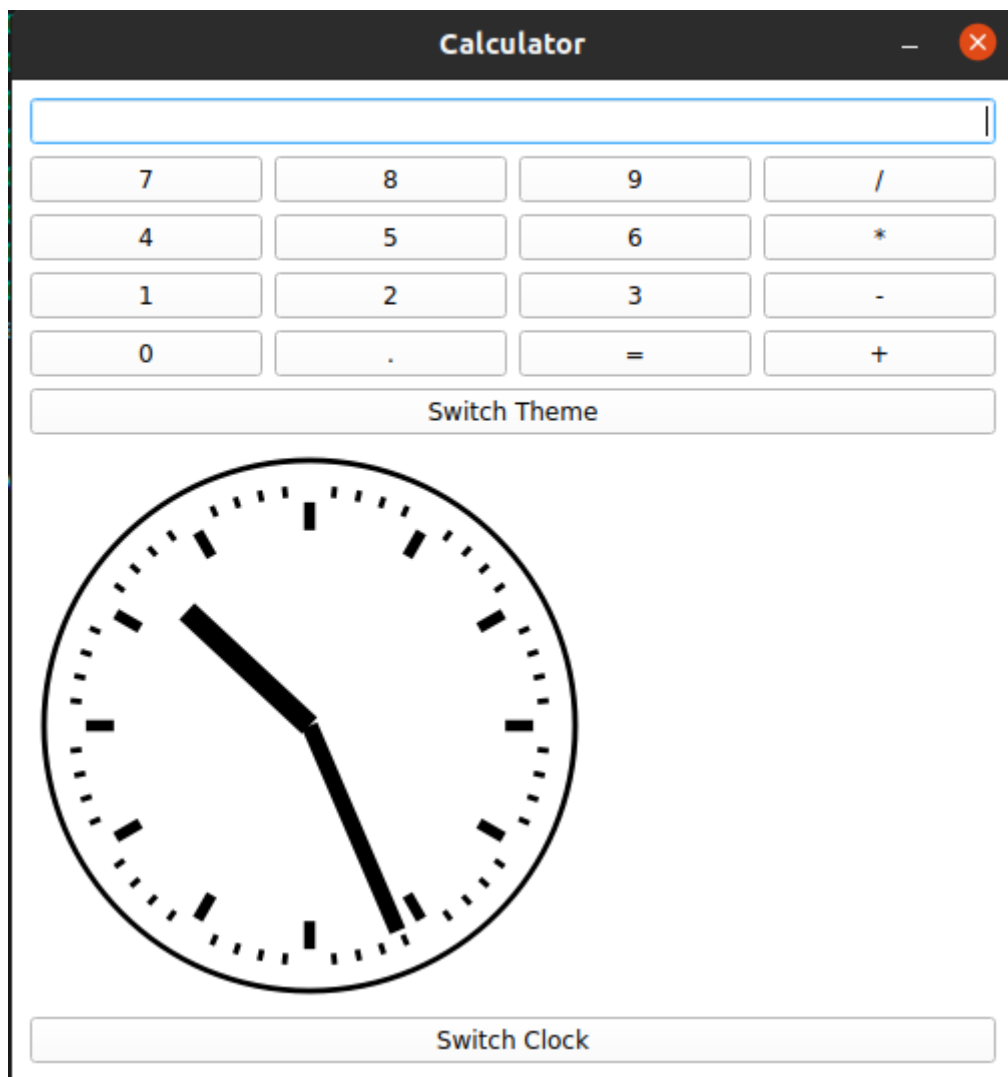
W programie jest dostępny również zegar, który wyświetla aktualną godzinę w naszej strefie czasowej. Możemy go zmieniać z analogowego na cyfrowy poprzez odpowiedni guzik – tj. “Switch clock”.

Na sam koniec, dodaliśmy możliwość zmiany motywu aplikacji. Są dostępne dwie opcje:

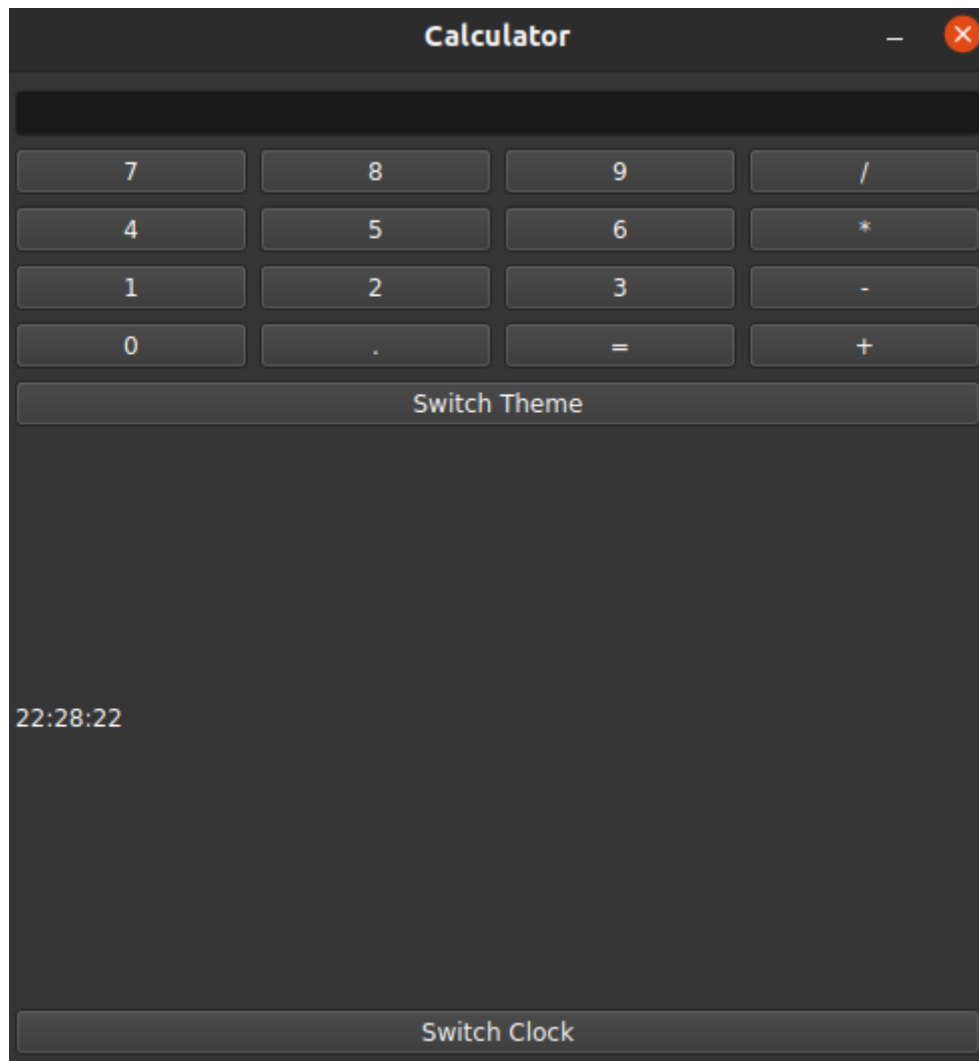
- dark
- light

Możemy go zmieniać za pomocą przycisku “Switch theme”.

Tak się prezentuje główny interfejs programu (wersja light):



Główny interfejs w motywie dark oraz z zegarem cyfrowym:



## 2) Opis przyjętych rozwiązań programowych

Aplikacja została napisana z użyciem programowania obiektowego. Składa się ona z klas:

- DigitalClock (odpowiedzialna za tworzenie cyfrowego zegara)
- AnalogClock (odpowiedzialna za tworzenie analogowego zegara)
- Calculator (odpowiedzialna za tworzenie kalkulatora, jego funkcjonalności oraz motywu)

W każdej klasie zostały zaimplementowane odpowiednie funkcje odpowiedzialne za zadane funkcjonalności

Jest również funkcja main, która odpala całą aplikację po uruchomieniu skryptu:

```
def main():
    """main function"""
    app = QApplication(sys.argv)
    calculator = Calculator()
    calculator.show()
    sys.exit(app.exec_())

if __name__ == '__main__':
    main()
```

### 3) Dyskusja osiągniętych wyników z wskazaniem wad oraz zalet napisanej aplikacji

Na pewno największą wadą napisanego przez nas programu jest dosyć ubogi design aplikacji, ponieważ jest bardzo prosty.

Skupialiśmy się raczej na funkcjonalnościach, aby połączyć wszystko w logiczną oraz dobrze działającą całość, co nam się udało.

Komunikacja programu z ekranem, myszką oraz klawiaturą działa jak najbardziej poprawnie.

Kalkulator zwraca poprawne wyniki, oraz jest uodporniony na pułapki typu dzielenie przez zero:

```
"""Calculation of the formula"""
try:
    result = eval(self.result_box.text())
    self.result_box.setText(str(result))
except ZeroDivisionError:
    self.result_box.setText('Error!')
except Exception as e:
    self.result_box.setText('Error')
```

Jako całość uważamy, że program od strony funkcjonalności jest napisany bardzo dobrze, jego jedyną wadą jest design, który oczywiście można poprawić według uznania estetycznych.

Najtrudniejszym zadaniem, ale nie jakoś bardzo wymagającym było dla nas napisanie analogowego zegara.

Tutaj funkcja, która go rysuje oraz wylicza czas:

```

def drawClock(self, painter):
    """Drawing the clock"""
    side = min(self.width(), self.height())
    painter.scale(side / 200.0, side / 200.0)
    painter.translate(100, 100)
    painter.setPen(Qt.NoPen)
    painter.setBrush(QBrush(Qt.black))

    for i in range(12):
        painter.save()
        painter.rotate(30 * i)
        painter.drawRect(70, -2, 10, 4)
        painter.restore()

    for i in range(60):
        if i % 5 != 0:
            painter.save()
            painter.rotate(6 * i)
            painter.drawRect(82, -1, 4, 2)
            painter.restore()

    time = QTime.currentTime()

    painter.setPen(Qt.NoPen)
    painter.setBrush(QBrush(Qt.black))

    painter.save()
    painter.rotate(30 * (time.hour() + time.minute() / 60.0))
    painter.drawRect(-4, -60, 8, 60)
    painter.restore()

    painter.save()
    painter.rotate(6 * (time.minute() + time.second() / 60.0))
    painter.drawRect(-3, -80, 6, 80)
    painter.restore()

    painter.setPen(QPen(Qt.black, 2))
    painter.setBrush(Qt.NoBrush)
    painter.drawEllipse(QPoint(0, 0), 95, 95)

```

Poza tym, nie mieliśmy żadnych problemów z napisaniem tej aplikacji. Chociaż podsumowując, analogowy zegar nie był dużym wyzwaniem.