

```

Asig_7
"""
Author: Coral S. Schmidt Montilla
Student Number: 148830
This module provides a class called TemperatureConversion for converting
temperatures between Celsius and Kelvin.
It also handles displaying the results and updating the graphical user
interface accordingly.
"""

from PyQt5 import uic
from PyQt5.QtCore import Qt
from PyQt5.QtGui import QPixmap
from PyQt5.QtWidgets import QMainWindow, QApplication, QStyleFactory

QApplication.setAttribute(Qt.AA_EnableHighDpiScaling, True)

class MainWindow(QMainWindow):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        uic.loadUi('Asig_7.ui', self)

        # Set initial index for ComboBox
        self.current_index = 0

        # Connect button click to conversion function
        self.pushButton.clicked.connect(self.show_result)

        # Connect combo box index change to image and color update function
        self.comboBox.currentIndexChanged.connect(self.update_image_and_color)

        # Set default color
        self.imageChanged.setStyleSheet('background-color: red')

        # Set default image
        self.imageChanged.setPixmap(QPixmap("cat.gif"))

    def update_image_and_color(self, index):
        colors = ['Red', 'Green', 'Blue', 'Yellow', 'Pink']
        images = ["cat.gif", "bird.gif", "dog.gif", "pig.gif", "rabbit.gif"]

        if self.checkBox.isChecked():
            image = QPixmap(images[self.current_index])
        else:
            self.current_index = index
            image = QPixmap(images[index])

        self.imageChanged.setStyleSheet(f"background-color: {colors[index]};")
        self.imageChanged.setPixmap(image)

    def show_result(self):
        try:
            temperature = float(self.lineEdit.text())
            if self.radioButton.isChecked():

```

```
        # Convert to Kelvin
        result = temperature + 273.15
        unit = '°K'
    elif self.radioButton_2.isChecked():
        # Convert to Celsius
        result = temperature - 273.15
        unit = '°C'
    else:
        self.label.setText("Please select a conversion method.")
        return

    # Update the result label
    self.label.setText(f"Result:\t\t{result:.2f}{unit}")

except Exception as e:
    self.label.setText("Error: " + str(e))

if __name__ == '__main__':
    app = QApplication([])
    QApplication.setStyle(QStyleFactory.create("Fusion"))
    window = MainWindow()
    window.show()
    app.exec()
```