app4. 실습에 사용된 코드

이 책의 4장부터 8장까지 실습에 사용된 전체 코드를 정리한 자료입니다.

- 실습 4.2.1 : main.py

```
# ch 4.2.1 main.py
import sys
from PyQt5.QtWidgets import QApplication, QWidget


class Calculator(QWidget):

    def __init__(self):
        super().__init__()
        self.initUI()

    def initUI(self):
        self.setWindowTitle('Calculator')
        self.resize(256,256)
        self.show()


if __name__=='__main__':
    app = QApplication(sys.argv)
    view = Calculator()
    sys.exit(app.exec_())
```

- 실습 4.2.3 : main.py

```
# ch 4.2.3 main.py
import sys
from PyQt5.QtWidgets import (QApplication,QWidget,QPushButton,QVBoxLayout,
                             QMessageBox)
from PyQt5.QtGui import QIcon


class Calculator(QWidget):

    def __init__(self):
        super().__init__()
        self.initUI()

    def initUI(self):
        self.btn1=QPushButton('Message',self)
        self.btn1.clicked.connect(self.activateMessage)
```

```
        vbox=QVBoxLayout()
        vbox.addStretch(1)
        vbox.addWidget(self.btn1)
        vbox.addStretch(1)

        self.setLayout(vbox)

        self.setWindowTitle('Calculator')
        self.setWindowIcon(QIcon('icon.png'))
        self.resize(256,256)
        self.show()

    def activateMessage(self):
        QMessageBox.information(self,"information","Button clicked!")


if __name__ =='__main__':
    app=QApplication(sys.argv)
    view=Calculator()
    sys.exit(app.exec_())
```

- 실습 4.2.4 : main.py

```
# ch 4.2.4 main.py
import sys
from PyQt5.QtWidgets import (QApplication,QWidget,QPushButton,QVBoxLayout,
                             QMessageBox, QPlainTextEdit)
from PyQt5.QtGui import QIcon


class Calculator(QWidget):

    def __init__(self):
        super().__init__()
        self.initUI()

    def initUI(self):
        self.te1 = QPlainTextEdit()
        self.te1.setReadOnly(True)

        self.btn1=QPushButton('Message',self)
        self.btn1.clicked.connect(self.activateMessage)
```

```
        vbox=QVBoxLayout()
        vbox.addWidget(self.te1)
        vbox.addWidget(self.btn1)
        vbox.addStretch(1)

        self.setLayout(vbox)

        self.setWindowTitle('Calculator')
        self.setWindowIcon(QIcon('icon.png'))
        self.resize(256,256)
        self.show()

    def activateMessage(self):
        # QMessageBox.information(self, "information", "Button clicked!")
        self.te1.appendPlainText("Button clicked!")


if __name__ =='__main__':
    app=QApplication(sys.argv)
    view=Calculator()
    sys.exit(app.exec_())
```

- 실습 4.4.1 : main.py

```
# ch 4.4.1 main.py
import sys
from PyQt5.QtWidgets import (QApplication,QWidget,QPushButton,QVBoxLayout,
                             QMessageBox, QPlainTextEdit, QHBoxLayout)
from PyQt5.QtGui import QIcon


class Calculator(QWidget):

    def __init__(self):
        super().__init__()
        self.initUI()

    def initUI(self):
        self.te1 = QPlainTextEdit()
        self.te1.setReadOnly(True)

        self.btn1=QPushButton('Message',self)
        self.btn1.clicked.connect(self.activateMessage)
        self.btn2=QPushButton('Clear',self)
```

```
        self.btn2.clicked.connect(self.clearMessage)

        hbox = QHBoxLayout()
        hbox.addStretch(1)
        hbox.addWidget(self.btn1)
        hbox.addWidget(self.btn2)

        vbox=QVBoxLayout()
        vbox.addWidget(self.te1)
        #vbox.addWidget(self.btn1)
        vbox.addLayout(hbox)
        vbox.addStretch(1)

        self.setLayout(vbox)

        self.setWindowTitle('Calculator')
        self.setWindowIcon(QIcon('icon.png'))
        self.resize(256,256)
        self.show()

    def activateMessage(self):
        self.te1.appendPlainText("Button clicked!")

    def clearMessage(self):
        self.te1.clear()


if __name__ =='__main__':
    app=QApplication(sys.argv)
    view=Calculator()
    sys.exit(app.exec_())
```

- 실습 5.2.1 : ui.py

```
# ch 5.2.1 ui.py
from PyQt5.QtWidgets import (QApplication,QWidget,QPushButton,QVBoxLayout,
                             QMessageBox, QPlainTextEdit, QHBoxLayout)
from PyQt5.QtGui import QIcon


class View(QWidget):

    def __init__(self):
        super().__init__()
```

```
        self.initUI()

    def initUI(self):
        self.te1 = QPlainTextEdit()
        self.te1.setReadOnly(True)

        self.btn1=QPushButton('Message',self)
        self.btn2=QPushButton('Clear',self)

        hbox = QHBoxLayout()
        hbox.addStretch(1)
        hbox.addWidget(self.btn1)
        hbox.addWidget(self.btn2)

        vbox=QVBoxLayout()
        vbox.addWidget(self.te1)
        vbox.addLayout(hbox)
        vbox.addStretch(1)

        self.setLayout(vbox)

        self.setWindowTitle('Calculator')
        self.setWindowIcon(QIcon('icon.png'))
        self.resize(256,256)
        self.show()

    def activateMessage(self):
        self.te1.appendPlainText("Button clicked!")

    def clearMessage(self):
        self.te1.clear()
```

- 실습 5.2.1 : ctrl.py

```
# ch 5.2.1 ctrl.py
class Control:

    def __init__(self, view):
        self.view = view
        self.connectSignals()

    def connectSignals(self):
        self.view.btn1.clicked.connect(self.view.activateMessage)
        self.view.btn2.clicked.connect(self.view.clearMessage)
```

- 실습 5.2.1 : main.py

```
# ch 5.2.1 main.py
import sys
from ui import View
from ctrl import Control
from PyQt5.QtWidgets import QApplication

def main():
    calc = QApplication(sys.argv)
    app=QApplication(sys.argv)
    view=View()
    Control(view=view)
    sys.exit(app.exec_())

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

- 실습 5.4.1 : ui.py

```
# ch 5.4.1 ui.py
from PyQt5.QtWidgets import (QApplication,QWidget,QPushButton,QVBoxLayout,
                             QMessageBox, QPlainTextEdit, QHBoxLayout, QLabel)
from PyQt5.QtGui import QIcon
from PyQt5.QtCore import QDate, Qt


class View(QWidget):

    def __init__(self):
        super().__init__()
        self.date = QDate.currentDate()
        self.initUI()

    def initUI(self):
        self.lbl1 = QLabel(self.date.toString(Qt.DefaultLocaleLongDate), self)
        self.te1 = QPlainTextEdit()
        self.te1.setReadOnly(True)

        self.btn1=QPushButton('Message',self)
        self.btn2=QPushButton('Clear',self)

        hbox = QHBoxLayout()
        hbox.addStretch(1)
```

```
        hbox.addWidget(self.btn1)
        hbox.addWidget(self.btn2)

        vbox=QVBoxLayout()
        vbox.addWidget(self.te1)
        vbox.addLayout(hbox)
        vbox.addWidget(self.lbl1)

        self.setLayout(vbox)

        self.setWindowTitle('Calculator')
        self.setWindowIcon(QIcon('icon.png'))
        self.resize(256,256)
        self.show()

    def activateMessage(self):
        self.te1.appendPlainText("Button clicked!")

    def clearMessage(self):
        self.te1.clear()
```

- 실습 6.2.1 : ui.py

```
# ch 6.2.1 ui.py
from PyQt5.QtWidgets import (QApplication,QWidget,QPushButton,QVBoxLayout,
                             QMessageBox, QPlainTextEdit, QHBoxLayout,
                             QLineEdit, QComboBox)
from PyQt5.QtGui import QIcon
from PyQt5 import QtCore


class View(QWidget):

    def __init__(self):
        super().__init__()
        self.initUI()

    def initUI(self):
        self.te1 = QPlainTextEdit()
        self.te1.setReadOnly(True)

        self.btn1=QPushButton('Message',self)
        self.btn2=QPushButton('Clear',self)
```

```
        self.le1=QLineEdit('0',self)
        self.le1.setAlignment(QtCore.Qt.AlignRight)

        self.le2=QLineEdit('0',self)
        self.le2.setAlignment(QtCore.Qt.AlignRight)

        self.cb = QComboBox(self)
        self.cb.addItems(['+', '-', '*', '/'])

        hbox_formular = QHBoxLayout()
        hbox_formular.addWidget(self.le1)
        hbox_formular.addWidget(self.cb)
        hbox_formular.addWidget(self.le2)

        hbox = QHBoxLayout()
        hbox.addStretch(1)
        hbox.addWidget(self.btn1)
        hbox.addWidget(self.btn2)

        vbox=QVBoxLayout()
        vbox.addWidget(self.te1)
        vbox.addLayout(hbox_formular)
        vbox.addLayout(hbox)
        vbox.addStretch(1)

        self.setLayout(vbox)

        self.setWindowTitle('Calculator')
        self.setWindowIcon(QIcon('icon.png'))
        self.resize(256,256)
        self.show()

    def activateMessage(self):
        self.te1.appendPlainText("Button clicked!")

    def clearMessage(self):
        self.te1.clear()
```

- 실습 6.2.2 : ctrl.py

```
# ch 6.2.2 ctrl.py
class Control:
```

```
    def __init__(self, view):
        self.view = view
        self.connectSignals()

    def calculate(self):
        pass

    def connectSignals(self):
        self.view.btn1.clicked.connect(self.calculate)
        self.view.btn2.clicked.connect(self.view.clearMessage)
```

- 실습 6.3.1 : ui.py

```
# ch 6.3.1 ui.py
from PyQt5.QtWidgets import (QApplication,QWidget,QPushButton,QVBoxLayout,
                             QMessageBox, QPlainTextEdit, QHBoxLayout,
                             QLineEdit, QComboBox)
from PyQt5.QtGui import QIcon
from PyQt5 import QtCore


class View(QWidget):

    def __init__(self):
        super().__init__()
        self.initUI()

    def initUI(self):
        self.te1 = QPlainTextEdit()
        self.te1.setReadOnly(True)

        self.btn1=QPushButton('Calc',self)
        self.btn2=QPushButton('Clear',self)

        self.le1=QLineEdit('0',self)
        self.le1.setAlignment(QtCore.Qt.AlignRight)
        self.le1.setFocus(True)
        self.le1.selectAll()

        self.le2=QLineEdit('0',self)
        self.le2.setAlignment(QtCore.Qt.AlignRight)

        self.cb = QComboBox(self)
        self.cb.addItems(['+', '-', '*', '/'])
```

```
        hbox_formular = QHBoxLayout()
        hbox_formular.addWidget(self.le1)
        hbox_formular.addWidget(self.cb)
        hbox_formular.addWidget(self.le2)

        hbox = QHBoxLayout()
        hbox.addStretch(1)
        hbox.addWidget(self.btn1)
        hbox.addWidget(self.btn2)

        vbox=QVBoxLayout()
        vbox.addWidget(self.te1)
        vbox.addLayout(hbox_formular)
        vbox.addLayout(hbox)
        vbox.addStretch(1)

        self.setLayout(vbox)

        self.setWindowTitle('Calculator')
        self.setWindowIcon(QIcon('icon.png'))
        self.resize(256,256)
        self.show()

    def setDisplay(self):
        self.te1.appendPlainText("Button clicked!")

    def clearMessage(self):
        self.te1.clear()
```

- 실습 6.3.2 : ui.py

```
# ch 6.3.2 ui.py
from PyQt5.QtWidgets import (QApplication,QWidget,QPushButton,QVBoxLayout,
                             QMessageBox, QPlainTextEdit, QHBoxLayout,
                             QLineEdit, QComboBox)
from PyQt5.QtGui import QIcon
from PyQt5 import QtCore


class View(QWidget):

    def __init__(self):
        super().__init__()
```

```
        self.initUI()

    def initUI(self):
        self.te1 = QPlainTextEdit()
        self.te1.setReadOnly(True)

        self.btn1=QPushButton('Message',self)
        self.btn2=QPushButton('Clear',self)

        self.le1=QLineEdit('0',self)
        self.le1.setAlignment(QtCore.Qt.AlignRight)

        self.le2=QLineEdit('0',self)
        self.le2.setAlignment(QtCore.Qt.AlignRight)

        self.cb = QComboBox(self)
        self.cb.addItems(['+', '-', '*', '/'])

        hbox_formular = QHBoxLayout()
        hbox_formular.addWidget(self.le1)
        hbox_formular.addWidget(self.cb)
        hbox_formular.addWidget(self.le2)

        hbox = QHBoxLayout()
        hbox.addStretch(1)
        hbox.addWidget(self.btn1)
        hbox.addWidget(self.btn2)

        vbox=QVBoxLayout()
        vbox.addWidget(self.te1)
        vbox.addLayout(hbox_formular)
        vbox.addLayout(hbox)
        vbox.addStretch(1)

        self.setLayout(vbox)

        self.setWindowTitle('Calculator')
        self.setWindowIcon(QIcon('icon.png'))
        self.resize(256,256)
        self.show()

    def activateMessage(self, text):
        self.te1.appendPlainText(text)
```

```
    def clearMessage(self):
        self.te1.clear()
```

- 실습 6.3.3 : ui.py

```
# ch 6.3.3 ui.py
from PyQt5.QtWidgets import (QApplication,QWidget,QPushButton,QVBoxLayout,
                             QMessageBox, QPlainTextEdit, QHBoxLayout,
                             QLineEdit, QComboBox)
from PyQt5.QtGui import QIcon
from PyQt5 import QtCore


class View(QWidget):

    def __init__(self):
        super().__init__()
        self.initUI()

    def initUI(self):
        self.te1 = QPlainTextEdit()
        self.te1.setReadOnly(True)

        self.btn1=QPushButton('Calc',self)
        self.btn2=QPushButton('Clear',self)

        self.le1=QLineEdit('0',self)
        self.le1.setAlignment(QtCore.Qt.AlignRight)
        self.le1.setFocus(True)
        self.le1.selectAll()

        self.le2=QLineEdit('0',self)
        self.le2.setAlignment(QtCore.Qt.AlignRight)

        self.cb = QComboBox(self)
        self.cb.addItems(['+', '-', '*', '/'])

        hbox_formular = QHBoxLayout()
        hbox_formular.addWidget(self.le1)
        hbox_formular.addWidget(self.cb)
        hbox_formular.addWidget(self.le2)

        hbox = QHBoxLayout()
```

```
        hbox.addStretch(1)
        hbox.addWidget(self.btn1)
        hbox.addWidget(self.btn2)

        vbox=QVBoxLayout()
        vbox.addWidget(self.te1)
        vbox.addLayout(hbox_formular)
        vbox.addLayout(hbox)
        vbox.addStretch(1)

        self.setLayout(vbox)

        self.setWindowTitle('Calculator')
        self.setWindowIcon(QIcon('icon.png'))
        self.resize(256,256)
        self.show()

    def setDisplay(self, text):
        self.te1.appendPlainText(text)

    def clearMessage(self):
        self.te1.clear()
```

- 실습 6.4.1 : ctrl.py

```
# ch 6.4.1 ctrl.py
class Control:

    def __init__(self, view):
        self.view = view
        self.connectSignals()

    def calculate(self):
        pass

    def connectSignals(self):
        self.view.btn1.clicked.connect(self.calculate)
        self.view.btn2.clicked.connect(self.view.clearMessage)

    def sum(self, a, b):
        return a+b
```

- 실습 6.4.4 : ctrl.py

```
# ch 6.4.4 ctrl.py
```

```
class Control:

    def __init__(self, view):
        self.view = view
        self.connectSignals()

    def calculate(self):
        pass

    def connectSignals(self):
        self.view.btn1.clicked.connect(self.calculate)
        self.view.btn2.clicked.connect(self.view.clearMessage)

    def sum(self, a, b):
        try:
            return a+b
        except:
            return"Calculation Error"
```

- 실습 6.6.1 : ui.py

```
# ch 6.6.1 ui.py
from PyQt5.QtWidgets import (QApplication,QWidget,QPushButton,QVBoxLayout,
                             QMessageBox, QPlainTextEdit, QHBoxLayout,
                             QLineEdit, QComboBox)
from PyQt5.QtGui import QIcon
from PyQt5 import QtCore


class View(QWidget):

    def __init__(self):
        super().__init__()
        self.initUI()

    def initUI(self):
        self.te1 = QPlainTextEdit()
        self.te1.setReadOnly(True)

        self.btn1=QPushButton('Calc',self)
        self.btn2=QPushButton('Clear',self)

        self.le1=QLineEdit('0',self)
        self.le1.setAlignment(QtCore.Qt.AlignRight)
```

```
        self.le1.setFocus(True)
        self.le1.selectAll()

        self.le2=QLineEdit('0',self)
        self.le2.setAlignment(QtCore.Qt.AlignRight)

        self.cb = QComboBox(self)
        self.cb.addItems(['+', '-', '*', '/', '^'])

        hbox_formular = QHBoxLayout()
        hbox_formular.addWidget(self.le1)
        hbox_formular.addWidget(self.cb)
        hbox_formular.addWidget(self.le2)

        hbox = QHBoxLayout()
        hbox.addStretch(1)
        hbox.addWidget(self.btn1)
        hbox.addWidget(self.btn2)

        vbox=QVBoxLayout()
        vbox.addWidget(self.te1)
        vbox.addLayout(hbox_formular)
        vbox.addLayout(hbox)
        vbox.addStretch(1)

        self.setLayout(vbox)

        self.setWindowTitle('Calculator')
        self.setWindowIcon(QIcon('icon.png'))
        self.resize(256,256)
        self.show()

    def setDisplay(self, text):
        self.te1.appendPlainText(text)

    def clearMessage(self):
        self.te1.clear()
```

- 실습 6.6.2 : ctrl.py

```
# ch 6.6.2 ctrl.py
class Control:

    def __init__(self, view):
```

```
        self.view = view
        self.connectSignals()

    def calculate(self):
        num1 = float(self.view.le1.text())
        num2 = float(self.view.le2.text())
        operator =self.view.cb.currentText()

        if operator =='+':
            return f'{num1} + {num2} = {self.sum(num1, num2)}'

        else:
            return"Calculation Error"

    def connectSignals(self):
        self.view.btn1.clicked.connect(self.calculate)
        self.view.btn2.clicked.connect(self.view.clearMessage)

    def sum(self, a, b):
        try:
            return a+b
        except:
            return"Calculation Error"
```

- 실습 6.6.7 : ctrl.py

```
# ch 6.6.7 ctrl.py
class Control:

    def __init__(self, view):
        self.view = view
        self.connectSignals()

    def calculate(self):
        num1 = float(self.view.le1.text())
        num2 = float(self.view.le2.text())
        operator =self.view.cb.currentText()

        if operator =='+':
            return f'{num1} + {num2} = {self.sum(num1, num2)}'

        else:
            return"Calculation Error"
```

```
    def connectSignals(self):
        self.view.btn1.clicked.connect(lambda:\
                                       self.view.setDisplay(self.calculate()))
        self.view.btn2.clicked.connect(self.view.clearMessage)

    def sum(self, a, b):
            return a+b
```

- 실습 7.1.3 : ctrl.py

```
# ch 7.1.3 ctrl.py
class Control:

    def __init__(self, view):
        self.view = view
        self.connectSignals()

    def calculate(self):
        num1 = float(self.view.le1.text())
        num2 = float(self.view.le2.text())
        operator =self.view.cb.currentText()

        if operator =='+':
            return f'{num1} + {num2} = {self.sum(num1, num2)}'

        else:
            return"Calculation Error"

    def connectSignals(self):
        self.view.btn1.clicked.connect(lambda:\
                                       self.view.setDisplay(self.calculate()))
        self.view.btn2.clicked.connect(self.view.clearMessage)

    def sum(self, a, b):
        return a+b

    def sub(self, a, b):
        return a-b

    def mul(self, a, b):
        return a*b

    def div(self, a, b):
        return a/b
```

```
    def pow(self, a, b):
        return pow(a, b)
```

- 실습 7.5.1 : ctrl.py

```
# ch 7.5.1 ctrl.py
class Control:

    def __init__(self, view):
        self.view = view
        self.connectSignals()

    def calculate(self):
        num1 = float(self.view.le1.text())
        num2 = float(self.view.le2.text())
        operator =self.view.cb.currentText()

        if operator =='+':
            return f'{num1} + {num2} = {self.sum(num1, num2)}'

        else:
            return"Calculation Error"

    def connectSignals(self):
        self.view.btn1.clicked.connect(lambda:\
                                       self.view.setDisplay(self.calculate()))
        self.view.btn2.clicked.connect(self.view.clearMessage)

    def sum(self, a, b):
        return a+b

    def sub(self, a, b):
        return a-b

    def mul(self, a, b):
        return a*b

    def div(self, a, b):
        try:
            if(b==0):
                raise Exception("Divisor Error")

        except Exception as e:
```

```
            return e

        return a/b

    def pow(self, a, b):
        return pow(a, b)
```

- 실습 7.5.2 : ctrl.py

```
# ch 7.5.2 ctrl.py
class Control:
    def __init__(self, view):
        self.view = view
        self.connectSignals()

    def calculate(self):
        num1 = float(self.view.le1.text())
        num2 = float(self.view.le2.text())
        operator =self.view.cb.currentText()

        if operator =='+':
            return f'{num1} + {num2} = {self.sum(num1, num2)}'

        else:
            return"Calculation Error"

    def connectSignals(self):
        self.view.btn1.clicked.connect(lambda:\
                                       self.view.setDisplay(self.calculate()))
        self.view.btn2.clicked.connect(self.view.clearMessage)

    def sum(self, a, b):
        return a+b

    def sub(self, a, b):
        return a-b

    def mul(self, a, b):
        return a*b

    def div(self, a, b):
        return a/b

    def pow(self, a, b):
```

```
        try:
           if (a==0):
                raise Exception("Base Error")

        except Exception as e:
            return e

        return pow(a, b)
```

- 실습 7.6.1 : ui.py

```
# ch 7.6.1 ui.py
from PyQt5.QtWidgets import (QApplication,QWidget,QPushButton,QVBoxLayout,
                             QMessageBox, QPlainTextEdit, QHBoxLayout,
                             QLineEdit, QComboBox)
from PyQt5.QtGui import QIcon
from PyQt5 import QtCore


class View(QWidget):

    def __init__(self):
        super().__init__()
        self.initUI()

    def initUI(self):
        self.te1 = QPlainTextEdit()
        self.te1.setReadOnly(True)

        self.btn1=QPushButton('Calc',self)
        self.btn2=QPushButton('Clear',self)

        self.le1=QLineEdit('0',self)
        self.le1.setAlignment(QtCore.Qt.AlignRight)
        self.le1.setFocus(True)
        self.le1.selectAll()

        self.le2=QLineEdit('0',self)
        self.le2.setAlignment(QtCore.Qt.AlignRight)

        self.cb = QComboBox(self)
        self.cb.addItems(['+', '-', '*', '/', '^', '%', '//'])

        hbox_formular = QHBoxLayout()
```

```
        hbox_formular.addWidget(self.le1)
        hbox_formular.addWidget(self.cb)
        hbox_formular.addWidget(self.le2)

        hbox = QHBoxLayout()
        hbox.addStretch(1)
        hbox.addWidget(self.btn1)
        hbox.addWidget(self.btn2)

        vbox=QVBoxLayout()
        vbox.addWidget(self.te1)
        vbox.addLayout(hbox_formular)
        vbox.addLayout(hbox)
        vbox.addStretch(1)

        self.setLayout(vbox)

        self.setWindowTitle('Calculator')
        self.setWindowIcon(QIcon('icon.png'))
        self.resize(256,256)
        self.show()

    def setDisplay(self, text):
        self.te1.appendPlainText(text)

    def clearMessage(self):
        self.te1.clear()
```

- 실습 7.6.2 : ctrl.py

```
# ch 7.6.2 ctrl.py
class Control:

    def __init__(self, view):
        self.view = view
        self.connectSignals()

    def calculate(self):
        try:
            num1 = float(self.view.le1.text())
            num2 = float(self.view.le2.text())
            operator =self.view.cb.currentText()

            if operator =='+':
```

```
                return f'{num1} + {num2} = {self.sum(num1, num2)}'
            elif operator =='-':
                return f'{num1} - {num2} = {self.sub(num1, num2)}'
            elif operator =='*':
                return f'{num1} * {num2} = {self.mul(num1, num2)}'
            elif operator =='/':
                return f'{num1} / {num2} = {self.div(num1, num2)}'
            elif operator =='^':
                return f'{num1} ^ {num2} = {self.pow(num1, num2)}'
            elif operator =='%':
                return f'{num1} % {num2} = {self.mod(num1, num2)}'
            else :
                return"Calculation Error"

        except:
            return"Calculation Error"

    def connectSignals(self):
        self.view.btn1.clicked.connect(lambda:\
                                       self.view.setDisplay(self.calculate()))
        self.view.btn2.clicked.connect(self.view.clearMessage)

    def sum(self, a, b):
        return a+b

    def sub(self, a, b):
        return a-b

    def mul(self, a, b):
        return a*b

    def div(self, a, b):
        try:
            if(b==0):
                raise Exception("Divisor Error")

        except Exception as e:
            return e

        return a/b

    def pow(self, a, b):
        try:
```

```
            if (a==0):
                raise Exception("Base Error")

        except Exception as e:
            return e

        return pow(a, b)

    def mod(self, a, b):
        try:
            if(b==0):
                raise Exception("Divisor Error")

        except Exception as e:
            return e

        return a%b
```

- 실습 7.7.2 : ui.py

```
# ch 7.7.2 ui.py
from PyQt5.QtWidgets import (QApplication,QWidget,QPushButton,QVBoxLayout,
                             QMessageBox, QPlainTextEdit, QHBoxLayout,
                             QLineEdit, QComboBox)
from PyQt5.QtGui import QIcon
from PyQt5 import QtCore


class View(QWidget):

    def __init__(self):
        super().__init__()
        self.initUI()

    def initUI(self):
        self.te1 = QPlainTextEdit()
        self.te1.setReadOnly(True)

        self.btn1=QPushButton('Calc',self)
        self.btn2=QPushButton('Clear',self)

        self.le1=QLineEdit('0',self)
        self.le1.setAlignment(QtCore.Qt.AlignRight)
        self.le1.setFocus(True)
```

```
        self.le1.selectAll()

        self.le2=QLineEdit('0',self)
        self.le2.setAlignment(QtCore.Qt.AlignRight)

        self.cb = QComboBox(self)
        self.cb.addItems(['+', '-', '*', '/'])

        hbox_formular = QHBoxLayout()
        hbox_formular.addWidget(self.le1)
        hbox_formular.addWidget(self.cb)
        hbox_formular.addWidget(self.le2)

        hbox = QHBoxLayout()
        hbox.addStretch(1)
        hbox.addWidget(self.btn1)
        hbox.addWidget(self.btn2)

        vbox=QVBoxLayout()
        vbox.addWidget(self.te1)
        vbox.addLayout(hbox_formular)
        vbox.addLayout(hbox)
        vbox.addStretch(1)

        self.setLayout(vbox)

        self.setWindowTitle('Calculator')
        self.setWindowIcon(QIcon('icon.png'))
        self.resize(256,256)
        self.show()

    def setDisplay(self, text):
        self.te1.appendPlainText(text)

    def clearMessage(self):
        self.te1.clear()
```

- 실습 7.7.2 : ctrl.py

```
# ch 7.7.2 ctrl.py
class Control:

    def __init__(self, view):
        self.view = view
```

```
        self.connectSignals()

    def calculate(self):
        try:
            num1 = float(self.view.le1.text())
            num2 = float(self.view.le2.text())
            operator =self.view.cb.currentText()

            if operator =='+':
                return f'{num1} + {num2} = {self.sum(num1, num2)}'
            elif operator =='-':
                return f'{num1} - {num2} = {self.sub(num1, num2)}'
            elif operator =='*':
                return f'{num1} * {num2} = {self.mul(num1, num2)}'
            elif operator =='/':
                return f'{num1} / {num2} = {self.div(num1, num2)}'
            else :
                return"Calculation Error"

        except:
            return"Calculation Error"

    def connectSignals(self):
        self.view.btn1.clicked.connect(lambda:\
                                       self.view.setDisplay(self.calculate()))
        self.view.btn2.clicked.connect(self.view.clearMessage)

    def sum(self, a, b):
        return a+b

    def sub(self, a, b):
        return a-b

    def mul(self, a, b):
        return a*b

    def div(self, a, b):
        try:
            if(b==0):
                raise Exception("Divisor Error")

        except Exception as e:
            return e
```

```
        return a/b
```

- 실습 8.1.3 : ui.py

```
# ch 8.1.3 ui.py
from PyQt5.QtWidgets import (QApplication,QWidget,QPushButton,QVBoxLayout,
                             QMessageBox, QPlainTextEdit, QHBoxLayout,
                             QLineEdit, QComboBox, QLabel)
from PyQt5.QtGui import QIcon, QFont
from PyQt5 import QtCore


class View(QWidget):

    def __init__(self):
        super().__init__()
        self.initUI()

    def initUI(self):
        self.te1 = QPlainTextEdit()
        self.te1.setReadOnly(True)

        self.lbl1 = QLabel('v2.3.0',self)
        self.lbl1.setFont(QFont('Consolas', 10))
        self.btn1=QPushButton('Calc',self)
        self.btn2=QPushButton('Clear',self)

        self.le1=QLineEdit('0',self)
        self.le1.setAlignment(QtCore.Qt.AlignRight)
        self.le1.setFocus(True)
        self.le1.selectAll()

        self.le2=QLineEdit('0',self)
        self.le2.setAlignment(QtCore.Qt.AlignRight)

        self.cb = QComboBox(self)
        self.cb.addItems(['+', '-', '*', '/', '^', '%'])

        hbox_formular = QHBoxLayout()
        hbox_formular.addWidget(self.le1)
        hbox_formular.addWidget(self.cb)
        hbox_formular.addWidget(self.le2)
```

```
        hbox = QHBoxLayout()
        hbox.addWidget(self.lbl1)
        hbox.addWidget(self.btn1)
        hbox.addWidget(self.btn2)

        vbox=QVBoxLayout()
        vbox.addWidget(self.te1)
        vbox.addLayout(hbox_formular)
        vbox.addLayout(hbox)
        vbox.addStretch(1)

        self.setLayout(vbox)

        self.setWindowTitle('Calculator')
        self.setWindowIcon(QIcon('icon.png'))
        self.resize(256,256)
        self.show()

    def setDisplay(self, text):
        self.te1.appendPlainText(text)

    def clearMessage(self):
        self.te1.clear()
```