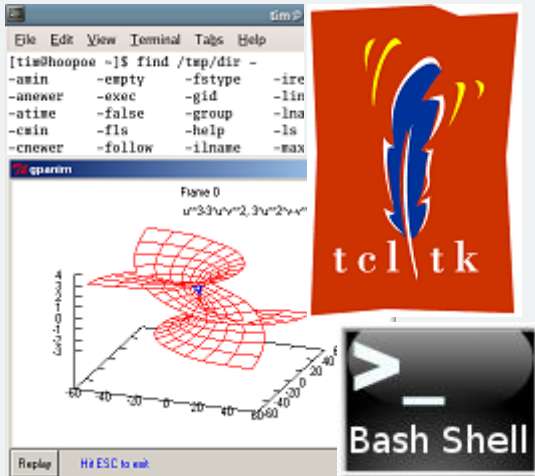


Интерпретируемые языки программирования



Лабораторная работа 6

ООП в Python. PyQt5, wxPython.



Объектно-ориентированное программирование в Python

Объявление класса:

<code>class User:</code>	←	Имя класса
<code> username = "Вася Пупкин"</code>	←	Переменная – член класса
<code> def print_name(self):</code>	←	Метод – член класса
<code> print("Привет, {}".format(self.username))</code>	←	Тело метода
<code>user = User()</code>	←	Создание объекта
<code>user.print_name()</code>	←	Вызов метода объекта
<code>user.username = "Почтальон Печкин"</code>	←	Обращение к переменной
<code>user.print_name()</code>	←	Вызов метода объекта



Классы в Python: конструктор

```
class User:

    username = "Просто юзер"

    def __init__(self, name):
        self.username = name

    def print_name(self):
        print("Привет, {}".format(self.username))

user = User("Почтальон Печкин")

user.print_name()
```



Классы в Python: деструктор

```
class User:

    username = "Просто юзер"

    def __init__(self, name):
        self.username = name
        print("В конструкторе")

    def print_name(self):
        print("Привет, {}".format(self.username))

    def __del__(self):
        print("В деструкторе")

user = User("Почтальон Печкин")
user.print_name()
del user
```



Особенности Python: переменные – члены класса

```
class User:

    def __init__(self, name):
        self.username = name
        print("В конструкторе")

    def print_name(self):
        print("Привет, {}".format(self.username))

    def __del__(self):
        print("В деструкторе")

user = User("Почтальон Печкин")
user.print_name()
del user
```



Наследование в Python (1)

```
class User:

    username = "Юзер"
    iq = 120

    def __init__(self, name, iq):
        self.username = name
        self.iq = iq

    def print_name(self):
        print ("Привет, {}".format(self.username))
        print ("Твой IQ={}".format(self.iq))
```



Наследование в Python (2)

```
class StupidUser(User):
```

```
    def __init__(self, name):
```

```
        User.__init__(self, name, 10)
```

```
    def __del__(self):
```

```
        User.__del__(self)
```

```
user = User("Леопольд")  
user.print_name()
```

```
user = StupidUser("Билли Бонс")  
user.print_name()
```



Программирование GUI на Python с Qt : PyQt5

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget

if __name__ == '__main__':

    app = QApplication(sys.argv)

    win = QWidget()
    win.resize(250, 150)
    win.move(300, 300)
    win.setWindowTitle('Simple')
    win.show()

    sys.exit(app.exec_())
```


PyQt5 : размещение виджетов (1)

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QPushButton

if __name__ == '__main__':

    app = QApplication(sys.argv)

    win = QWidget()
    win.resize(250, 150)
    win.move(300, 300)
    win.setWindowTitle('Simple')
    win.show()

    btn = QPushButton('Click me', win)
    btn.setGeometry(10, 10, 100, 35)

    sys.exit(app.exec_())
```

PyQt5 : размещение виджетов (2)

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QPushButton

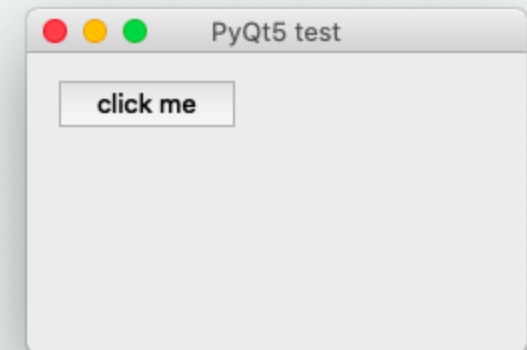
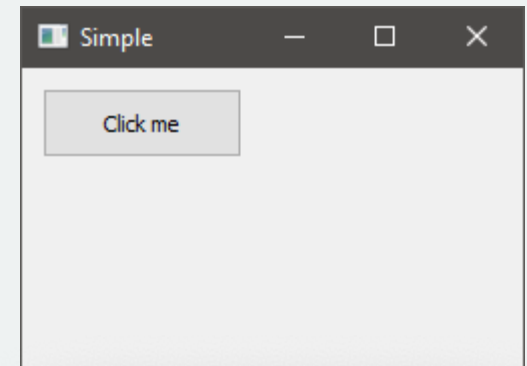
if __name__ == '__main__':

    app = QApplication(sys.argv)

    win = QWidget()
    win.resize(250, 150)
    win.move(300, 300)
    win.setWindowTitle('Simple')
    win.show()

    btn = QPushButton('Click me', win)
    btn.setGeometry(10, 10, 100, 35)
    btn.show()

    sys.exit(app.exec_())
```



PyQt5 : обработка сигналов

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QPushButton
```

```
→ def onClick():
    QMessageBox.information(None, 'PyQt5',
```

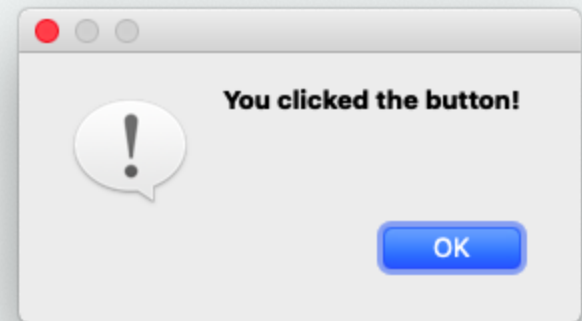
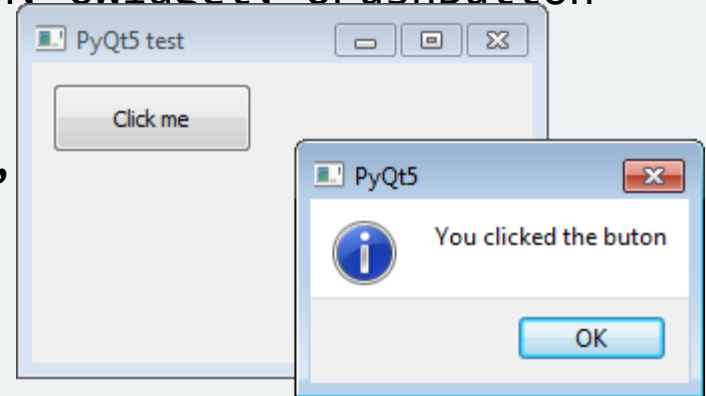
```
if __name__ == '__main__':
    app = QApplication(sys.argv)
```

```
...
```

```
btn = QPushButton('Click me', win)
btn.setGeometry(10, 10, 100, 35)
→ btn.clicked.connect(onClick)
```

```
win.show()
```

```
sys.exit(app.exec_())
```





PyQt5 : QMainWindow

```
import sys
from PyQt5.QtWidgets import QApplication, QMainWindow

class MainWindow(QMainWindow):
    def __init__(self):
        super().__init__()
        self.resize(250, 150)
        self.move(100, 100)
        self.setWindowTitle('PyQt5 MainWindow')

if __name__ == '__main__':
    app = QApplication(sys.argv)

    win = MainWindow()
    win.show()

    sys.exit(app.exec_())
```



PyQt5 : QAction, QMenu

```
import sys
from PyQt5.QtWidgets import QApplication, QMainWindow

class MainWindow(QMainWindow):
    def __init__(self):
        super().__init__()
        self.resize(250, 150)
        self.move(100, 100)
        self.setWindowTitle('PyQt5 MainWindow')

        actExit = QAction('&Exit', self)
        actExit.setShortcut('Ctrl+Q')

        menubar = self.menuBar()
        fileMenu = menubar.addMenu('&File')
        fileMenu.addAction(actExit)
```

PyQt5 : Обработчики событий

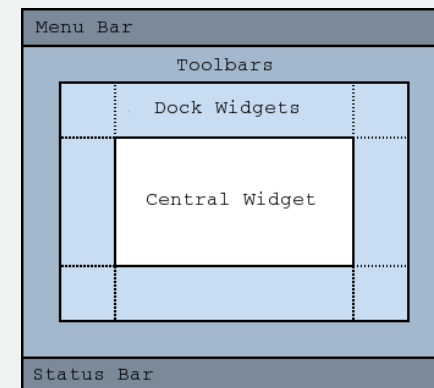
```
class MainWindow(QMainWindow):
    def __init__(self):
        super().__init__()
        self.resize(250, 150)
        self.move(100, 100)
        self.setWindowTitle('PyQt5 MainWindow')

        actExit = QAction('&Exit', self)
        actExit.setShortcut('Ctrl+Q')
        actExit.triggered.connect(self.OnExit)
        ...

    def OnExit(self):
        self.close()
```

PyQt5 : размещение виджетов

```
class MainWindow(QMainWindow):  
    def __init__(self):  
        super().__init__()  
        self.resize(250, 150)  
        self.move(100, 100)  
        self.setWindowTitle('PyQt5 MainWindow')  
  
        ...  
  
        text = QTextEdit()  
        self.setCentralWidget(text)
```





PyQt5 : рисование с QPainter (2)

```
def paintEvent(self, event):  
    p = QPainter()  
    p.begin(self)  
  
    p.drawRect(10, 10, 100, 100)  
  
    p.setPen(QColor(0, 0, 200))  
    p.setBrush(QColor(0, 200, 0))  
    p.drawEllipse(10, 55, 90, 105)  
    p.drawText(event.rect(), Qt.AlignCenter, 'Hello!')  
  
    p.end()
```




PyQt5: диалоги - открытия файлов

```
class MainWindow(QMainWindow):  
  
    ...  
  
    def OnFileOpen(self):  
        fname = QFileDialog.getOpenFileName(self, 'Open file', 'D:/')  
  
        if fname[0]:  
            f = open(fname[0], 'r')  
            ...
```

Пример на wxPython

```
import wx
```

```
class Example(wx.Frame):  
    def __init__(self, title):  
        super(Example, self).__init__(None, title=title,  
                                       size=(300, 200))  
  
        self.Move((800, 250))
```

```
def main():  
    app = wx.App()  
    ex = Example(title='Тест xwPython')  
    ex.Show()  
    app.MainLoop()
```

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

