

## Модель непосредственного управления. (Drag and Drop)

Наберите текст программы и прокомментируйте как можно больше строк в ней, в данной работе применять QT дизайнер и конвертировать интерфейс нет необходимости.

«Программа позволяющая открыть файл \*.txt с кодировкой utf-8»

```
1 import sys
2 from PyQt5.QtWidgets import (QPushButton, QWidget, QLineEdit, QApplication, QPlainTextEdit, QFileDialog)
3 from PyQt5.QtCore import Qt, QMimeData
4
5
6 class PlainTextEdit(QPlainTextEdit):
7     def __init__(self, title, parent):
8         super().__init__(title, parent)
9         self.setAcceptDrops(True)
10
11     def dragEnterEvent(self, e):
12         if e.mimeData().hasFormat("text/uri-list"):
13             e.accept()
14         else:
15             e.ignore()
16
17     def dropEvent(self, e): #определяем, что произойдёт после отпущания кнопки мыши
18         try:
19             print("нуть")
20             self.data = e.mimeData().text().replace("file:///", "")
21             print(self.data)
22             with open(self.data, "r", encoding = "utf-8") as file:
23                 self.setPlainText(file.read())
24         except Exception as e:
25             print(e)
26
27 class Example (QWidget):
28     def __init__(self):
29         super().__init__()
30         self.initUI()
31
32     def initUI(self):
33         self.setAcceptDrops(True)
34         self.setWindowTitle("Метод Drag and Drop")
35         self.setGeometry(300, 300, 400, 225)
36         self.textEdit = PlainTextEdit("", self)
37         self.textEdit.move(20, 20)
38
39 if __name__ == "__main__":
40     app = QApplication(sys.argv)
41     myapp = Example()
42     myapp.show()
43     sys.exit(app.exec_())
44
```

## «Программа позволяющая удалить выделенную строку из виджета PlainTextedit»

```
1 import sys
2 from PyQt5.QtWidgets import (QPushButton, QWidget,
3 QPlainTextEdit, QApplication)
4
5 class Button(QPushButton):
6     def __init__(self, title, parent):
7         super().__init__(title, parent)
8         self.setAcceptDrops(True) #разрешение перетаскивания для виджета
9
10    def dragEnterEvent(self, e): #сообщаем тип данных(текстовый)
11        if e.mimeData().hasFormat('text/plain'):
12            e.accept()
13        else:
14            e.ignore()
15
16    def dropEvent(self,e): #определяет что мы делаем после события перетаскивания (меняем текст виджета)
17        myapp.edit.cut()
18
19 class Example(QWidget):
20    def __init__(self):
21        super().__init__()
22        self.initUI()
23
24    def initUI(self): #виджет QLineEdit имеет встроенную поддержку операций перетаскивания метод setDragEnabled().
25        self.edit = QPlainTextEdit('', self)
26        ##self.edit.setDragEnabled(True)
27        self.edit.move(30, 65)
28
29        button = Button('Перенесите сюда и всё удалится', self)
30        button.move(300, 65)
31
32        self.setWindowTitle('Метод Drag and Drop')
33        self.setGeometry(300, 300, 500, 300)
34
35
36
37
38
39 if __name__ == "__main__":
40     app = QApplication(sys.argv)
41     myapp = Example()
42     myapp.show()
43     sys.exit(app.exec_())
44
```

## «Программа «Тест на соответствие»»

```
1 import sys
2 from PyQt5.QtWidgets import QPushButton, QWidget, QApplication, QFrame, QLabel
3 from PyQt5.QtCore import Qt, QMimeData
4 from PyQt5.QtGui import QDrag, QFont
5 from tkinter import Tk, messagebox
6 class button(QPushButton):
7     def __init__(self, title, parent):
8         super().__init__(title, parent)
9         self.setStyleSheet("background-color: rgb(170, 170, 255);")
10        font = QFont()
11        font.setPointSize(12)
12        self.setFont(font)
13
14    def mousePressEvent(self, e): #функция перетаскивания кнопки
15        if e.buttons() != Qt.LeftButton: #перетаскивание левой кнопки мыши
16            return
17        mimeData = QMimeData()
18        drag = QDrag(self) #создается объект, обеспечивает поддержку перетаскивания данных, основанную на MIME-типе.
19        drag.setMimeData(mimeData)
20        drag.setHotSpot(e.pos() - self.rect().topLeft())
21        dropAction = drag.exec_(Qt.MoveAction) #начинается операция перетаскивания
22
23class Example(QWidget):
24    def __init__(self):
25        super().__init__()
26        self.initUI()
27
28    def initUI(self):
29        font = QFont()
30        font.setPointSize(12)
31        self.fra1 = QFrame(self)
32        self.fra1.setStyleSheet("background-color: rgb(170, 170, 255);")
33        self.fra1.setGeometry(30, 20, 211, 81)
34        self.lb11 = QLabel(self.fra1)
35        self.lb11.setFont(font)
36        self.lb11.setText('Самый вкусный\пнапиток?')
37        self.lb11.setGeometry(30, 20, 151, 35)
38        self.fra2 = QFrame(self)
39        self.fra2.setStyleSheet("background-color: rgb(170, 170, 255);")
40        self.fra2.setGeometry(30, 140, 211, 81)
41        self.lb12 = QLabel(self.fra2)
42        self.lb12.setFont(font)
43        self.lb12.setText('Когда отменили\крепостное право?')
44        self.lb12.setGeometry(20, 20, 191, 35)
45
46        self.fra3 = QFrame(self)
47        self.fra3.setStyleSheet("background-color: rgb(170, 170, 255);")
48        self.fra3.setGeometry(30, 260, 211, 81)
49        self.lb13 = QLabel(self.fra3)
50        self.lb13.setFont(font)
51        self.lb13.setText('Как зовут автора\п программы?')
52        self.lb13.setGeometry(40, 20, 151, 35)
53        self.fra4 = QFrame(self)
54        self.fra4.setStyleSheet("background-color: rgb(170, 170, 255);")
55        self.fra4.setGeometry(30, 380, 211, 81)
56        self.lb14 = QLabel(self.fra4)
57        self.lb14.setFont(font)
58        self.lb14.setText('Хотите ли вы 5?')
59        self.lb14.setGeometry(30, 20, 151, 35)
60
61        self.fra5 = QFrame(self)
62        self.fra5.setStyleSheet("background-color: rgb(152, 152, 152);")
63        self.fra5.setGeometry(280, 20, 211, 81)
64        self.fra6 = QFrame(self)
65        self.fra6.setStyleSheet("background-color: rgb(152, 152, 152);")
66        self.fra6.setGeometry(280, 140, 211, 81)
67        self.fra7 = QFrame(self)
68        self.fra7.setStyleSheet("background-color: rgb(152, 152, 152);")
69        self.fra7.setGeometry(280, 260, 211, 81)
70        self.fra8 = QFrame(self)
71        self.fra8.setStyleSheet("background-color: rgb(152, 152, 152);")
72        self.fra8.setGeometry(280, 380, 211, 81)
73
74        self.ans1 = button('Максим', self)
75        self.ans1.setGeometry(530, 20, 211, 81)
76        self.ans2 = button('1861', self)
77        self.ans2.setGeometry(530, 140, 211, 81)
78        self.ans3 = button('Блэк Монстр', self)
79        self.ans3.setGeometry(530, 260, 211, 81)
80        self.ans4 = button('Да', self)
81        self.ans4.setGeometry(530, 380, 211, 81)
82
83        self.but = QPushButton('Ответить', self)
84        self.but.setGeometry(698, 497, 85, 33)
85        self.but.setStyleSheet("background-color: rgb(170, 170, 255);")
86        self.but.setFont(font)
87        self.but.clicked.connect(self.otvet)
88
89        self.setAcceptDrops(True)
```



```

89     self.setWindowTitle('Тест на соответствие')
90     self.setGeometry(300, 100, 783, 530)
91
92     def dragEnterEvent(self, e):
93         e.accept()
94
95     def dropEvent(self, e): #определяем, что произойдет после отпускания кнопки мыши
96         position = e.pos() #определяем текущую позицию
97         s = str(position)
98         s = s[(s.find('.') + 1):-1]
99         num = s.split(',')
100        num[0], num[1] = int(num[0]), int(num[1])
101        e.source().setParent(self) #назначаем источник для e (выбранный вариант ответа)
102        if num[0] > 270 and num[0] < 501:
103            if num[1] > 10 and num[1] < 111:
104                e.source().move(280, 20)
105            elif num[1] > 130 and num[1] < 231:
106                e.source().move(280, 140)
107            elif num[1] > 250 and num[1] < 351:
108                e.source().move(280, 260)
109            elif num[1] > 370 and num[1] < 471:
110                e.source().move(280, 380)
111        else:
112            e.source().move(position)
113        e.setDropAction(Qt.MoveAction)
114        e.accept()
115
116    def otvet(self):
117        Tk().withdraw()
118        otv1 = str(self.ans3.pos())
119        otv1 = otv1[(otv1.find('.') + 1):-1]
120        otv1 = otv1.split(',')
121        otv1[0], otv1[1] = int(otv1[0]), int(otv1[1])
122
123        otv2 = str(self.ans2.pos())
124        otv2 = otv2[(otv2.find('.') + 1):-1]
125        otv2 = otv2.split(',')
126        otv2[0], otv2[1] = int(otv2[0]), int(otv2[1])
127
128        otv3 = str(self.ans1.pos())
129        otv3 = otv3[(otv3.find('.') + 1):-1]
130        otv3 = otv3.split(',')
131        otv3[0], otv3[1] = int(otv3[0]), int(otv3[1])
132
133        otv4 = str(self.ans4.pos())
134        otv4 = otv4[(otv4.find('.') + 1):-1]
135        otv4 = otv4.split(',')
136
137        otv4[0], otv4[1] = int(otv4[0]), int(otv4[1])
138        if otv1[0] == 280 and otv2[0] == 280 and otv3[0] == 280 and otv4[0] == 280:
139            if otv1[1] == 20 and otv2[1] == 140 and otv3[1] == 260 and otv4[1] == 380:
140                messagebox.showinfo('Поздравляю', 'Все вопросы отвечены верно')
141            else:
142                messagebox.showerror('Ошибка', 'Ответы неправильные')
143        else:
144            messagebox.showerror('Ошибка', 'Не все поля заполнены')
145    if __name__ == '__main__':
146        app = QApplication(sys.argv)
147        ex = Example()
148        ex.show()
149        app.exec_()

```

Самостоятельно: измените вопросы и варианты ответов, законспектируйте основные моменты в тетрадь.

**Drag-and-drop** (в переводе с англ. — «тащи-и-бросай», «бери-и-брось») — способ оперирования элементами интерфейса в интерфейсах пользователя (как графическим, так и текстовым, где элементы GUI реализованы при помощи псевдографики) при помощи манипулятора «мышь» или сенсорного экрана.

Способ реализуется путём «захвата» (нажатием и удержанием главной (*первой*, чаще левой) кнопки мыши) отображаемого на экране компьютера объекта, программно доступного для подобной операции, и перемещении его в другое место (для изменения расположения) либо «бросания» его на другой элемент (для вызова соответствующего, предусмотренного программой, действия). По отношению к окнам (также способным к перемещению подобным способом) данный термин обычно не употребляется.