Практическая работа №16
Обработка событий в оконном приложении на Python

1 Цель работы

1.1 Научиться обрабатывать действия пользователя, события клавиатуры и мыши в программах на Python;

1.2 Закрепить навык составления программ методами процедурного и событийно-ориентированного программирования.

2 Литература

2.1 Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:https://znanium.com/read?id=390096. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – гл.12.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Написать оконное приложение, в которое добавить многострочное поле ввода и кнопку Сохранить. При нажатии на кнопку или клавиши Ctrl-S должно происходить сохранение файла на рабочий стол.

5.2 Написать оконное приложение с двумя полями ввода. Пользователь вводит два числа, при нажатии на кнопку с соответсвующим знаком должна выполятся математическая операция. Например, пользователь ввёл в первое поле 23, а во второе 67, при нажатии на кнопку «+», ему выведется «Сумма чисел 23 и 67 = 90».

5.3 При наведении курсора на кнопку, она должна менять цвет, при удалении курсора с кнопки, цвет должен меняться на прежний.

6 Порядок выполнения работы

6.1 Запустить Python IDLE и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

33

7.2 Цель работы

7.3 Ответы на контрольные вопросы
7.4 Вывод

8 Контрольные вопросы

8.1 Какова общая форма написания обработчиков событий?
8.2 Как добавить прослушиватель на события мыши?
8.3 Как добавить прослушиватель на события клавиатуры?
8.4 Какие виды событий мыши можно обработать в программах на Python?
8.5 Какие виды событий клавиатуры можно обработать в программах на Python?

9 Приложение

Tkinter позволяет обрабатывать события виджетов. Для обработки распространенных и наболее используемых событий Tkinter предоставляет интерфейс команд. Например, для обработки нажатия на кнопку ее параметру command надо передать функцию, которая будет вызываться при нажатии на кнопку

Однако что, если мы хотим обрабатывать другие события виджета. Например, для кнопки обработать получение фокуса, или обработать нажатие клавиши клавиатуры? Для подобных ситуаций Tkinter предоставляет ряд встроенных событий. Наиболее распространенные из них:

•Activate: окно становится активным.

•Deactivate: окно становится неактивным.

•MouseWheel: прокрутка колеса мыши.

•KeyPress: нажатие клавиши на клавиатуре.

•KeyRelease: освобождение нажатой клавиши

•ButtonPress: нажатие кнопки мыши.

•ButtonRelease: освобождение кнопки мыши.

•Motion: движение мыши.

•

•Destroy: удаление виджета

•FocusIn: получение фокуса

•FocusOut: потеря фокуса.

•Enter: указатель мыши вошел в пределы виджета.

•Leave: указатель мыши покинул виджет.

Привязка событий

Для привязки события к виджету применяется метод **bind()**:

```
bind(событие, функция)
```

В качестве первого параметра указывается обрабатываемое событие, а в качестве второго - функция, которая обрабатывает событие.

Например, обработаем события получения и потери фокуса для кнопки: (на след. странице)