



*School of
Computer
Science*

ГРАФИЧЕСКИЙ ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ TKINTER

ПРОГРАММИРОВАНИЕ НА PYTHON

Лекции для IT-школы



GUI-ПРИЛОЖЕНИЯ

ПРИНЦИПЫ ПОСТРОЕНИЯ

- GUI = Graphical User Interface
- Общая схема построения GUI:
 - Главный цикл ожидания событий
 - Callback-функция – обработчик события
 - События, для которых не назначены обработчики, игнорируются
 - На каждом витке главного цикла анализируются наступившие события и вызываются их обработчики
 - Набор графических компонентов (виджетов) и средства их компоновки в окне приложения



- Возможности построения GUI на Python:
 - Реализуются с помощью библиотек tkinter, PyQt, PyGTK, wxPython, и др.
 - Независимость от платформы
- Мы будем использовать **tkinter**:
 - набор инструментов для создания GUI
 - создавать и заполнять рамки
 - создавать и применять:
 - кнопки (buttons)
 - текстовые поля и области
 - флажки (check buttons)
 - переключатели (radio buttons)



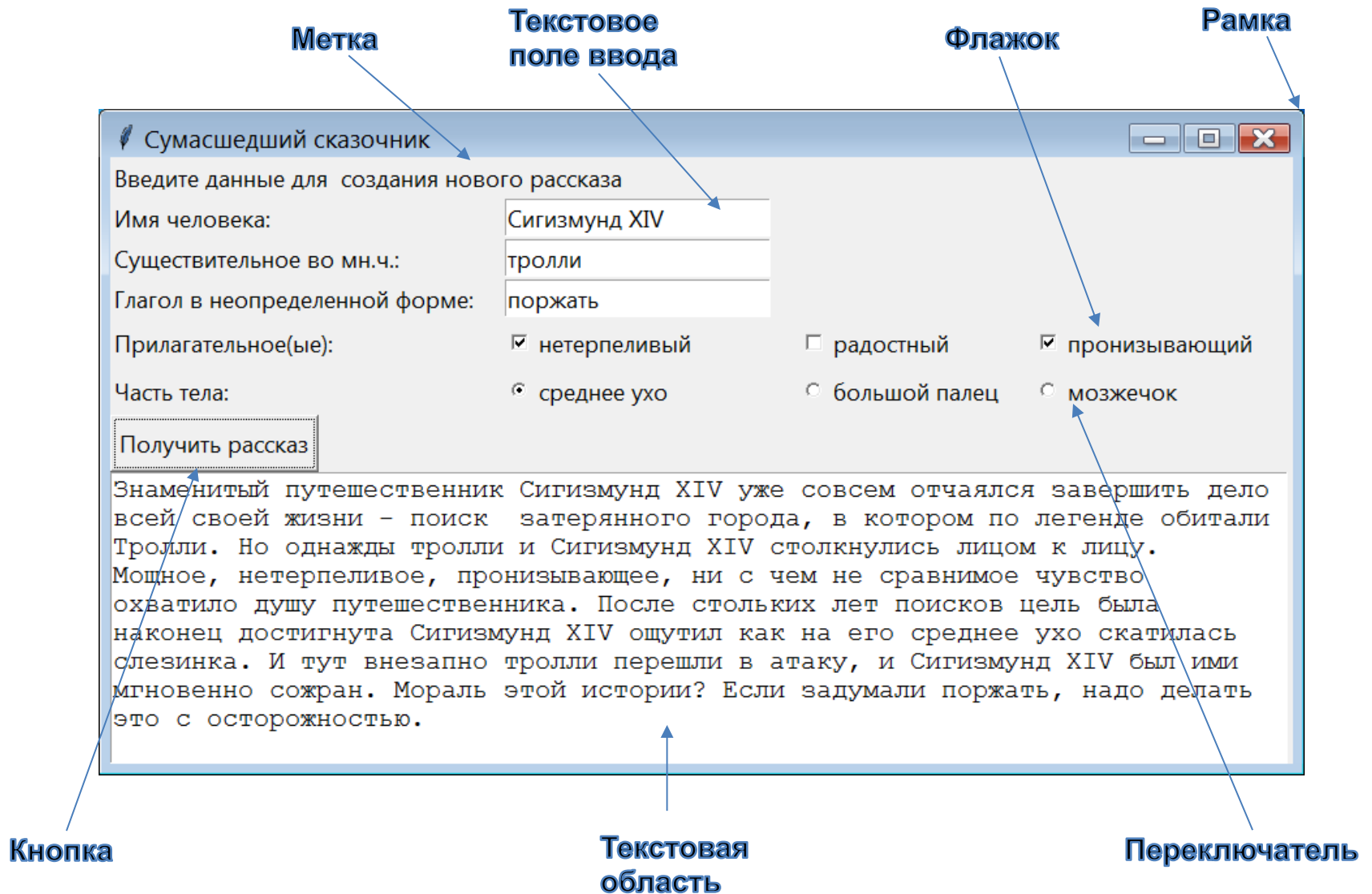
ИЗБРАННЫЕ ЭЛЕМЕНТЫ GUI В TKINTER

Элемент	Класс	Описание
Рамка	Frame	Служит базой для других элементов
Метка	Label	Отображает значок или текст, который нельзя изменять
Кнопка	Button	Выполняет какое-либо действие в ответ на нажатие
Текстовое поле	Entry	Позволяет вводить и отображать одну строку текста
Текстовая область	Text	Позволяет вводить и отображать несколько строк в виде единого многострочного текста
Флажок	Checkbutton	Позволяет включить или выключить какой-либо параметр
Переключатель	Radiobutton	Позволяет выбрать один параметр из нескольких сгруппированных





ВИД ДИАЛоговых ЭЛЕМЕНТОВ GUI





ПРОСТЕЙШЕЕ ОКНО

- Базовое окно – это инстанция класса Tk
- Используем методы базового окна:
 - `title()` – для задания заголовка окна
 - `geometry()` – для задания размеров
 - `mainloop()` – для запуска главного цикла ожидания событий
- См. пример в скрипте `simple_gui.pyw`



ПРИМЕНЕНИЕ МЕТОК

- Метка – это просто текст, который нельзя редактировать
- Все новые виджеты привязываются к своему родителю – так рамка создается внутри окна, а метка – в рамке
- Метод `grid()` у виджета управляет расположением элемента
- См. пример в скрипте `labeler.pyw`



ПРИМЕНЕНИЕ КНОПОК

- Кнопка – элемент управления класса `Button`
- Вид диалогового элемента можно изменять методом `configure()`
- Можно управлять параметрами виджета через атрибуты, подобные словарю
- См. пример в скрипте `lazy_buttons.pyw`



СОЗДАНИЕ GUI С ПОМОЩЬЮ КЛАССА

- Создаем новый класс `Application`, производный от `tkinter.Frame`
- Инициализируем стандартную рамку в конструкторе и вызываем наш метод `self.create_widgets()`
- Создаваемые кнопки сохраняются как атрибуты объекта `Application`
- См. пример в скрипте `lazy_buttons2.pyw`



СВЯЗЫВАНИЕ ДИАЛоговых ЭЛЕМЕНТОВ С ОБРАБОТЧИКАМИ СОБЫТИЙ

- Обработчики событий – функции, либо метод-функции класса, которые вызываются при наступлении событий
- Связывание обработчика с событием = сохранение ссылки на функцию-обработчик в определенном параметре диалогового элемента
- См. пример в скрипте [click_counter.pyw](#)



ТЕКСТОВЫЕ ПОЛЯ И ОБЛАСТИ. МЕНЕДЖЕР РАЗМЕЩЕНИЯ GRID

- Текстовое поле удобно для ввода и отображения одной строки текста
- Текстовая область – для текста из нескольких строк
- Менеджер размещения `grid` располагает элементы в рамке, привязывая их к воображаемой сетке
- См. пример в скрипте `longevity.pyw`



МЕНЕДЖЕР РАЗМЕЩЕНИЯ GRID. ПРИМЕР РАЗМЕТКИ ОКНА LONGEVITY.PY

Заголовок и стандартные кнопки управления окном

row = 0 column = 0	row = 0 column = 1	row = 0 column = 2
row = 1 column = 0	row = 1 column = 1	row = 1 column = 2
row = 2 column = 0	row = 2 column = 1	row = 2 column = 2



ПРИМЕНЕНИЕ ФЛАЖКОВ

- Флажки (пометки) позволяют выбрать из группы альтернатив одну или несколько, вплоть до всех
- Виджеты, к которым не нужно обращаться в коде, можно разместить сразу после создания и не сохранять в переменных
- Флажок связывается с объектом типа `BooleanVar()` для хранения выбора
- См. пример в скрипте `movie_chooser.pyw`



ПРИМЕНЕНИЕ ПЕРЕКЛЮЧАТЕЛЕЙ

- Переключатели (радио-кнопки) позволяют выбрать одно единственное значение из нескольких вариантов
- Переключатель связывается с объектом типа `StringVar()` для хранения выбора
- См. пример в скрипте `movie_chooser2.pyw`



ПРОГРАММА «СУМАСШЕДШИЙ СКАЗОЧНИК»

- Скомбинируем рассмотренные диалоговые элементы в одном приложении
- Программа выдает текст сказания, который изменяется в зависимости от значений, выбранных в полях ввода, с помощью флажков и переключателей
- См. пример в скрипте `mad_tale.pyw`



СВОБОДНОЕ ЗАДАНИЕ №1

- Перепишите игру «Угадай число», создайте для нее графический интерфейс с классом
- В решении обязательно нужно применить собственный **class**
- Используйте алгоритм игры из консольной реализации программы в скрипте **console_guess_number.py**



СВОБОДНОЕ ЗАДАНИЕ №2

МЕНЮ В РЕСТОРАНЕ

- Напишите GUI-программу с классом, которая показывает несложное ресторанное меню с блюдами (не более 10 шт), их ценами, принимает заказ и выводит на экран сумму счета. Пример реализации:

Ресторанное меню

Ресторан абхазской кухни 'Приходи - порадуем!'

	Цена, р	Количество, шт
Чахохбили с рыбой	220	<input type="text" value="1"/>
Курица в томатном соусе	185	<input type="text"/>
Блины домашние с укропом	75	<input type="text" value="2"/>
Хинкали	220	<input type="text"/>
Бурум в лаваше	430	<input type="text"/>
Шашлык из баранины	380	<input type="text" value="3"/>
Овощи в собственном соку	120	<input type="text"/>
Кебаб	360	<input type="text"/>

Рассчитать стоимость заказа

Стоимость: 1510

СПАСИБО ЗА ВНИМАНИЕ !
ВОПРОСЫ ?



*School of
Computer
Science*