УДК 004.432
DOI 10.33514/1694-7851-2024-1-274-281

Барганалиева Ж.К.
ага окутуучу
И. Арабаев атындагы Кыргыз мамлекеттик университети
Бишкек ш.
barganalieva@mail.ru
Садырова М.Р.
ага окутуучу
И. Арабаев атындагы Кыргыз мамлекеттик университети
Бишкек ш.
sadyrova.m.r@mail.ru
Асанбекова Н.О.
физика-математика илимдеринин кандидаты, доценттин м.а.
И. Арабаев атындагы Кыргыз мамлекеттик университети
Бишкек ш.
asnuor@mail.ru

PYTHONДО TKINTER МЕНЕН ИНТЕРАКТИВДҮҮ ТЕСТ ТҮЗҮҮ

Аннотация: "Pythondo Tkinter менен интерактивдүү тестти түзүү" макаласы Python тилинде Tkinter библиотекасын колдонуу менен тестирлөө тиркемесин түзүү боюнча кеңири колдонмо болуп саналат. Tkinterдин негиздерин карап чыгуудан жана колдонмонун негизги терезесин түзүүдөн баштап, тест суроолорун, колдонуучу интерфейсин жана тест логикасын түзүү этап-этабы менен жүрөт.

Макалада авторлор Pythondун жөнөкөй синтаксисинин жаңы баштагандар үчүн анын жеткиликтүүлүгүн баса белгилеп, Tkinterдин колдонуунун оңойлугуна басым жасайт. Алар, ошондой, эле Tkinter кросс-платформа экенин белгилешип, бул ар кандай операциялык системаларда түзүлгөн тиркемелерди кеңири аудиторияга жеткиликтүү кылат.

Окуп жатып, сиз бир нече варианттуу суроолордун тизмесин түзүүнү жана жоопторду текшерүү жана натыйжаларды көрсөтүү логикасын кантип ишке ашырууну үйрөнөсүз. Колдонмо интерфейсинин коддогу мисалдары жана сүрөттөрү тестти түзүүнүн ар бир этабын түшүнүүнү жеңилдетет.

Бул окуу куралы Tkinter аркылуу интерактивдүү Python тиркемелерин түзүүнү үйрөнүүнү каалагандар үчүн баалуу булак болуп саналат. Бул башталгыч иштеп чыгуучулар үчүн да, Python менен мурдатан тааныш болгон жана графикалык интерфейстерди түзүү боюнча жөндөмдөрүн кеңейтүүнү каалагандар үчүн ылайыктуу. Акыр-аягы, ал колдонуучуга ыңгайлуу жана интуитивдик интерфейс менен ыңгайлаштырылган билим тесттерин түзүүгө мүмкүндүк берет.

Негизги сөздөр: Python, Tkinter, графикалык интерфейс, интерактивдүү тест, колдонуучу интерфейси, тестирлөө, жыйынтыктар, программалоо, окутуу.

Барганалиева Ж.К.

старший преподаватель
Кыргызский Государственный Университет имени И. Арабаева
г. Бишкек
barganalieva@mail.ru
Садырова М.Р.
старший преподаватель
Кыргызский Государственный Университет имени И. Арабаева
г. Бишкек
sadyrova.m.r@mail.ru
Асанбекова Н.О.
кандидат физико-математических наук, и.о. доцента
Кыргызский Государственный Университет имени И. Арабаева
г. Бишкек
asnuor@mail.ru

СОЗДАНИЕ ИНТЕРАКТИВНОГО ТЕСТА НА TKINTER В PYTHON

Аннотация: Статья "Создание интерактивного теста на Tkinter в Python" представляет собой подробное руководство по созданию приложения для проведения тестирования с использованием библиотеки Tkinter в Python. Начиная с обзора основ Tkinter и создания главного окна приложения, автор пошагово описывает процесс создания тестовых вопросов, пользовательского интерфейса и логики тестирования.

Статья акцентирует внимание на простоте использования Tkinter, подчеркивая его доступность для новичков благодаря простому синтаксису Python. Она также отмечает кроссплатформенность Tkinter, что делает созданные приложения доступными для широкой аудитории на различных операционных системах.

В ходе чтения вы узнаете, как создать список вопросов с вариантами ответов, а также как реализовать логику проверки ответов и отображение результатов. Примеры кода и изображения интерфейса приложения облегчают понимание каждого этапа создания теста.

Это руководство представляет ценный ресурс для всех, кто хочет изучить создание интерактивных приложений на Python с помощью Tkinter. Оно подходит как для начинающих разработчиков, так и для тех, кто уже знаком с Python и хочет расширить свои навыки в области создания графических интерфейсов. В конечном итоге, оно позволяет создавать пользовательские тесты для проверки знаний с удобным и интуитивно понятным интерфейсом.

Ключевые слова: Python, Tkinter, графический интерфейс, интерактивный тест, пользовательский интерфейс, тестирование, результаты, программирование, обучение.

Barganalieva J.K.
senior lecturer
Kyrgyz State University named after I. Arabaev
s. Bishkek
barganalieva@mail.ru

Sadyrova M.R.
senior lecturer
Kyrgyz State University named after I. Arabaev
s. Bishkek
sadyrova.m.r@mail.ru
Asanbekova N.O.
Ph.D., Acting associate professor
Kyrgyz State University named after I. Arabaev
s. Bishkek
asnuor@mail.ru

CREATE AN INTERACTIVE TEST WITH TKINTER IN PYTHON

Abstract: The article "Creating an interactive test with Tkinter in Python" is a detailed guide to creating a testing application using the Tkinter library in Python. Starting with an overview of the basics of Tkinter and creating the main application window, the author walks through the step-by-step process of creating test questions, user interface, and test logic.

The article focuses on Tkinter's ease of use, emphasizing its accessibility for beginners thanks to Python's simple syntax. She also notes that Tkinter is cross-platform, which makes the applications created accessible to a wide audience on various operating systems.

As you read, you'll learn how to create a list of multiple-choice questions and how to implement the logic for checking answers and displaying the results. Code examples and images of the application interface make it easier to understand each stage of creating a test.

This tutorial provides a valuable resource for anyone who wants to learn how to create interactive Python applications using Tkinter. It is suitable for both beginner developers and those who are already familiar with Python and want to expand their skills in creating graphical interfaces. Ultimately, it allows you to create custom knowledge tests with a user-friendly and intuitive interface.

Keywords: Python, Tkinter, graphical interface, interactive test, user interface, testing, results, programming, training.

В мире программирования Python существует множество библиотек для создания графического пользовательского интерфейса (GUI). Каждая из них имеет свои особенности, преимущества и недостатки, что позволяет разработчикам выбирать наиболее подходящий инструмент для своих задач. Однако одной из наиболее популярных и широко используемых является Tkinter. Tkinter является стандартной библиотекой для создания GUI в Python и обеспечивает простой и эффективный способ создания интерактивных приложений.

С особым вниманием выделяем Tkinter, как одной из наиболее широко используемых и доступных для начинающих разработчиков. Она базируется на библиотеке Tk, которая была разработана для языка программирования Tcl. Tkinter предоставляет простой и интуитивно понятный способ создания интерфейсов, что делает ее идеальным выбором для новичков в программировании. Основные преимущества Tkinter включают:

- Простота использования: Tkinter предоставляет легкий и понятный интерфейс для создания графических приложений. Он использует простой и понятный синтаксис Python, что делает процесс разработки более доступным для новичков.

- Кроссплатформенность: Приложения, созданные с использованием Tkinter, могут запускаться на различных операционных системах, включая Windows, macOS и Linux. Это делает Tkinter универсальным инструментом для разработки приложений, которые будут доступны для широкой аудитории.
- Большое сообщество и документация: Tkinter имеет обширное сообщество пользователей и разработчиков, что обеспечивает доступ к множеству ресурсов, включая документацию, учебные пособия и форумы поддержки. Это позволяет быстро находить ответы на вопросы и решения для различных задач.
- Интеграция с Python: Tkinter интегрируется непосредственно с языком программирования Python, что обеспечивает прямой доступ к множеству библиотек и инструментов, доступных в экосистеме Python. Это позволяет разработчикам легко расширять функциональность своих приложений, используя сторонние модули и пакеты.

В этой статье мы рассмотрим процесс создания интерактивного теста с помощью библиотеки Tkinter. Мы шаг за шагом пройдемся по основным этапам разработки, начиная с создания главного окна и заканчивая реализацией логики тестирования и отображением результатов.

Основы Tkinter

Первый шаг в создании любого приложения на Tkinter – это импортировать саму библиотеку. Для этого в начале вашего кода добавьте следующую строку:

```
import tkinter as tk
```

Далее мы создадим главное окно нашего приложения, используя класс Tk из Tkinter:

```
root = tk.Tk()
root.title("Интерактивный тест")
root.geometry("600x400")
```

Этот код создает новое главное окно с заголовком "Интерактивный тест" и размерами 600x400 пикселей.

Создание тестовых вопросов

Теперь, когда у нас есть главное окно, мы можем перейти к созданию тестовых вопросов. Для этого определим список словарей, каждый из которых будет содержать информацию о вопросе, вариантах ответов и правильном ответе:

```
questions = [
    {
        "question": "Что такое Python?",
        "options": ["Язык программирования", "Фрукт", "Автомобиль", "Спорт"],
        "correct_answer": 0
    },
    {
        "question": "Какая функция используется для вывода текста в консоль в Python?",
        "options": ["print()", "input()", "read()", "write()"],
        "correct_answer": 0
    },
    # Добавьте еще вопросов по вашему усмотрению
]
```

Создание пользовательского интерфейса

Теперь, когда у нас есть список вопросов, давайте создадим пользовательский интерфейс для отображения вопросов и вариантов ответов. Для каждого вопроса мы будем использовать метку (Label) для отображения текста вопроса и радиокнопки (Radiobutton) для отображения вариантов ответов.

```
question_label = tk.Label(root, text=questions[current_question]["question"])
question_label.pack()
option_var = tk.IntVar()
for i, option in enumerate(questions[current_question]["options"]):
    option_radio = tk.Radiobutton(root, text=option, variable=option_var, value=i)
    option_radio.pack()
```

Логика тестирования

После того, как пользователь выберет ответ, мы должны проверить его правильность и обновить результаты. Мы можем сделать это, определив функцию check_answer:

```
def check_answer():
    selected_option = option_var.get()
    correct_answer = questions[current_question]["correct_answer"]
    if selected_option == correct_answer:
        # Обновляем счет
        score += 1
    # Переходим к следующему вопросу
    current_question += 1
```

Отображение результатов

По завершении теста мы можем отобразить пользователю его результаты:

```
result_label = tk.Label(root, text=f"Ваш результат: {score}/{len(questions)}")
result_label.pack()
```

Полный код программы выглядит следующим образом:

```
import tkinter as tk
from tkinter import messagebox
class QuizApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Тестирование")
        self.root.geometry("650x350")
        self.root.resizable(False, False)
        self.score = 0  # Переменная для хранения счета
        self.current_question = 0  # Текущий вопрос
        # Вопросы и варианты ответов
        self.questions = [
            {
                'question': 'Вопрос 1: Что такое Python?',
                'options': ['Язык программирования', 'Фрукт', 'Автомобиль', 'Спорт'],
                'correct_answer': 0  # Индекс правильного ответа
            },
            {
```

```
            'question': 'Вопрос 2: Какая функция используется для вывода текста в консоль в Python?',
            'options': ['print()', 'input()', 'read()', 'write()'],
            'correct_answer': 0
        },
        {
            'question': 'Вопрос 3: Что такое GUI?',
            'options': ['Графический интерфейс пользователя', 'Глобальная инициатива урбанизации',
                        'Гастрономическая учебная инициатива', 'Гренадин ультиматум'],
            'correct_answer': 0
        },
        {
            'question': 'Вопрос 4: Какой язык программирования используется в веб-разработке?',
            'options': ['HTML', 'JavaScript', 'Python', 'C++'],
            'correct_answer': 1
        },
        {
            'question': 'Вопрос 5: Какой символ используется для создания комментариев в Python?',
            'options': ['//', '/* */', '#', '%%'],
            'correct_answer': 2
        }
    ]
    self.question_label = tk.Label(root, text=self.questions[self.current_question]['question'])
    self.question_label.grid(row=0, column=0, columnspan=2)
    self.option_var = tk.IntVar()
    self.option_buttons = []
    for i, option in enumerate(self.questions[self.current_question]['options']):
        option_radio = tk.Radiobutton(root, text=option, variable=self.option_var, value=i)
        option_radio.grid(row=i // 2 + 1, column=i % 2)
        self.option_buttons.append(option_radio)
    submit_button = tk.Button(root, text="Далее", command=self.check_answer)
    submit_button.grid(row=3, column=1)
def check_answer(self):
    selected_option = self.option_var.get()
    correct_answer = self.questions[self.current_question]['correct_answer']
    if selected_option == correct_answer:
        self.score += 10
    self.current_question += 1
    self.option_var.set(-1)
    if self.current_question < len(self.questions):
        self.show_question()
    else:
        self.show_result()
```

```
    def show_question(self):
        if self.current_question < len(self.questions):
            question = self.questions[self.current_question]['question']
            options = self.questions[self.current_question]['options']
            self.question_label.config(text=question)
            for i, option_radio in enumerate(self.option_buttons):
                option_radio.config(text=options[i])
    def show_result(self):
        messagebox.showinfo("Результат", f"Вы набрали {self.score} баллов из {100} ")
        if self.score == 100:
            messagebox.showinfo("Обращение ", f"Да ты ГЕНИЙ")
        elif self.score > 70:
            messagebox.showinfo("Обращение ", f"неплохо неплохо !!!")
        elif self.score > 50:
            messagebox.showinfo("Обращение ", f"учится некогда не поздно !!!")
        else:
            messagebox.showinfo("Обращение ", f"учись !!!")
        self.root.destroy()
root = tk.Tk()
app = QuizApp(root)
root.mainloop()
```

На рисунке 1 показано изображение после запуска программы.

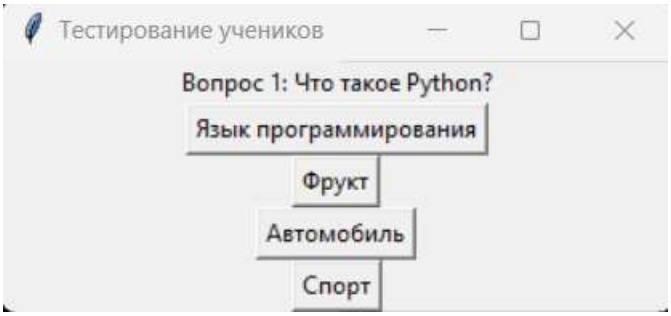

Рис.1. Изображение после запуска программы.

На рисунке 2 показан результат, после прохождения тестирования.

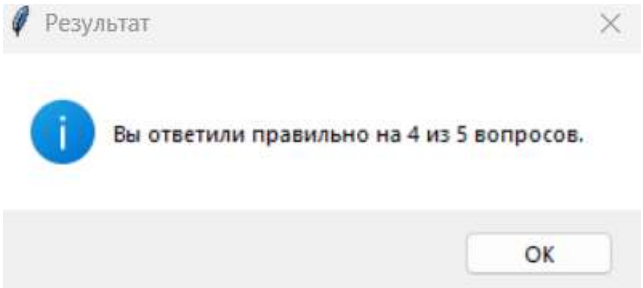

Рис.2. Результат, после прохождения тестирования.

На рисунке 3 отображается фрагмент кода на Python.

```python
import tkinter as tk
from tkinter import messagebox


class QuizApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Тестирование")
        self.root.geometry("650x350")
        self.root.resizable(False, False)
        self.score = 0  # Переменная для хранения счета
        self.current_question = 0  # Текущий вопрос
        # Вопросы и варианты ответов
        self.questions = [
            {
                'question': 'Вопрос 1: Что такое Python?',
                'options': ['Язык программирования', 'Фрукт', 'Автомобиль', 'Спорт'],
                'correct_answer': 0  # Индекс правильного ответа
            },
            {
                'question': 'Вопрос 2: Какая функция используется для вывода текста в консоль в Python?',
                'options': ['print()', 'input()', 'read()', 'write()'],
                'correct_answer': 0
```

Рис.3.Фрагмент кода.

Итак, в этой статье мы рассмотрели процесс создания интерактивного теста с использованием библиотеки Tkinter в Python. Мы изучили основы создания GUI, разработали тестовые вопросы, реализовали логику тестирования и отобразили результаты пользователю. Теперь вы можете создать свой собственный интерактивный тест и использовать его для проверки знаний ваших пользователей!

Список использованной литературы

1. Гэддис Т. Начинаем программировать на Python. – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, – 2019. – С. 768.
2. Лучано Рамальо Python. К вершинам мастерства. – М.: ДМК Пресс, 2016. – С. 768.
3. Любанович Билл Простой Python. Современный стиль программирования. – СПб.: Питер, 2016. – С. 480.: – (Серия «Бестсепперы O’Reilly»).
4. Рейтц К., Шлюссер Т. Автостопом по Python. – СПб.: Питер, – 2017. – С. 336.: ил. – (Серия «Бестселлеры O’Reilly»).
5. Свейгарт, Эл. Автоматизация рутиных задач с помощью Python: практическое руководство для начинающих. Пер. с англ. — М.: Вильямс, – 2016. – С. 592.

Рецензент: кандидат физико-математических наук, доцент Бексултанов Ж.Т.