

Травкин Евгений Иванович

канд. пед. наук, доцент

Тарасюк Владимир Борисович

канд. техн. наук, доцент

ФГБОУ ВО «Курский государственный университет»

г. Курск, Курская область

**ОБЩИЕ ПРИНЦИПЫ ОБУЧЕНИЯ РАЗРАБОТКЕ
ИНФОРМАЦИОННЫХ СИСТЕМ КОМПЬЮТЕРНОГО
ТЕСТИРОВАНИЯ ДЛЯ БУДУЩИХ
ПРЕПОДАВАТЕЛЕЙ ИНФОРМАТИКИ**

Аннотация: в статье рассматривается последовательность разработки информационной системы компьютерного тестирования будущими преподавателями информатики. Приведена структура и содержание основных программных блоков, позволяющих воспроизвести описываемый процесс.

Ключевые слова: информационная система компьютерного тестирования, преподаватель информатики, цифровая трансформация.

Цифровая трансформация общества в современных условиях диктует необходимость динамичного реагирования системы образования и ее модернизации и адаптации, как в содержательном, так и в организационно-методическом сопровождении. Особую роль в решении данных задач отводится преподавателям информатики и смежных дисциплин, подготовка которых должна реагировать на современные вызовы. Как отмечалось в статье [1] «Подготовка будущих преподавателей информатики к реализации профессиональной деятельности в современных условиях может быть эффективно организована междисциплинарно на основе проектирования, разработки, тестирования и использования в реальных учебных условиях информационной системы управления обучением (ИСУО)». Одним из компонентов которой являются информационная система компьютерного тестирования. «Система компьютерного тестирования – это универсальный инструмент для определения уровня знаний,

умений и навыков студентов на всех уровнях образовательного процесса [2, с. 16]».

Информационная система компьютерного тестирования представляет собой программно-аппаратную платформу для организации цифрового контроля знаний. Совершенство, функционал, методическая релевантность информационной системы компьютерного тестирования в значительной мере определяется не только содержательными модулями, но и правильным выбором программного инструментария для ее разработки, в свою очередь являющегося объектам изучения в рамках профильных дисциплин.

В качестве примера предлагаем рассмотреть последовательность разработки информационной системы компьютерного тестирования средствами Python [3]. Компьютерная система тестирования предназначена для автоматизации процесса тестирования с загрузкой вопросов из файла и повышение качества тестирования за счет автоматизированного сбора результатов.

Требования к системе: должна автоматизировать процесс тестирования, работать в интерактивном режиме, иметь удобный интерфейс, предоставлять возможность дальнейшего расширения модулей и контрольно-измерительных материалов.

Компьютерная система тестирования должна состоять из следующих блоков.

1. Блок подтверждения выбора ответа и перехода к следующему вопросу.
2. Блок генерирования базы вопросов из файла.
3. Блок генерирования интерфейса компьютерной системы тестирования.
4. Блок загрузки вопросов на форму и фиксации ответов.
5. Блок обработки результатов тестирования и подведения итогов.
6. Блок вывода результата тестирования.

Рассмотрим проектирование каждого функционального блока компьютерной системы тестирования.

1. Блок подтверждения выбора ответа и перехода к следующему вопросу.

#Функцию подтверждения ответа

```
def nextt():
    global var_delay
    var_delay.set(1)
```

2. Блок генерирования базы вопросов из файла.

```
#Считаем данные из файла
f = open('base.txt')
title_window = f.readline()
instructions = f.readline()
n = int(f.readline())
#Генерируем базу вопросов из файла
data = list()
for i in range(n):
    temp = list()
    for j in range(5):
        if j == 4:
            temp.append(int(f.readline()))
        else:
            temp.append(f.readline())
    data.append(temp)
```

3. Блок генерирования интерфейса компьютерной системы тестирования.

Для подключения библиотек tkinter используем следующий формат подключения.

```
from tkinter import *
from tkinter import ttk
```

Далее следует генерирование интерфейса компьютерной системы тестирования.

```
#Создаем окно программы
root = Tk()
root.title(title_window)
w = root.winfo_screenwidth()
```

```
h = root.winfo_screenheight()
w = w // 2 # середина экрана
h = h // 2
w = w - 300 # смещение от середины
h = h - 200
root.geometry('600x400+{ }+{ }'.format(w, h))
root.resizable(False, False) #запрет изменения размеров окна
#Создание и отображение виджета Label (надпись)
var_question = StringVar()
question = Label(root, textvariable = var_question, font = 'Arial 22',
wraplength = 500)
var_question.set(instructions)
question.pack()
image1 = PhotoImage(file=«1.png»)
label_image1 = ttk.Label(image=image1, compound=«top»)
label_image1.pack()
#Задержка до нажатия кнопки
var_delay = IntVar()
#Создание и отображение виджета Button (кнопка)
btn = Button(root, text = 'NEXT', font='Arial 16', command=nextt)
btn.pack()
btn.wait_variable(var_delay)
btn.destroy()
label_image1.destroy()
result = 0 #Количество верных ответов
var = IntVar() #Выбранный ответ
```

Представленный программный код данного функционального блока позволяет спроектировать следующий интерфейс компьютерной системы тестирования на, представленный на рисунке 1.

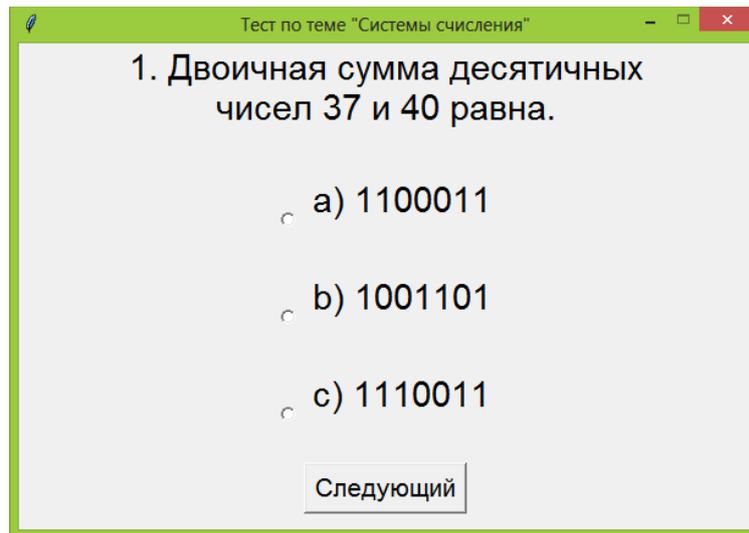


Рис. 1. Интерфейс компьютерной системы тестирования

При проектировании интерфейса использовались виджеты: Label (надпись), Button (кнопка), Radiobutton (переключатели).

4. Блок загрузки загрузки вопросов на форму и фиксации ответов.

```
#Загрузка вопросов на форму и фиксация ответов
for i in range(n):
    var.set(-1)
    var_question.set(data[i][0])
    r1 = Radiobutton(text=data[i][1], font='Arial 22', variable=var, value=1)
    r2 = Radiobutton(text=data[i][2], font='Arial 22', variable=var, value=2)
    r3 = Radiobutton(text=data[i][3], font='Arial 22', variable=var, value=3)
    r1.pack()
    r2.pack()
    r3.pack()
    btn = Button(root, text = «NEXT», font='Arial 16', command=5ext)
    btn.pack()
    btn.wait_variable(var_delay)
    if data[i][4] == var.get():
        result += 1
    r1.destroy()
```

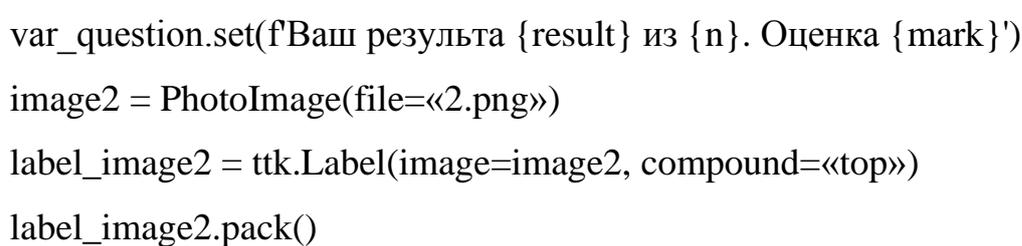
```
r2.destroy()
r3.destroy()
btn.destroy()
```

5. Блок обработки результатов тестирования и подведения итогов.

```
#Расчет оценки
if result >= 0.85 * n:
    mark = 5
elif 0.65 <= result < 0.85:
    mark = 4
elif 0.5 <= result < 0.65:
    mark = 3
else:
    mark = 2
```

6. Блок вывод результата тестирования.

```
#Вывод результата тестирования
var_question.set(f'Ваш результа {result} из {n}. Оценка {mark}')
```



```
image2 = PhotoImage(file=«2.png»)
label_image2 = ttk.Label(image=image2, compound=«top»)
label_image2.pack()

#Метод взаимодействия с пользователем у окна
root.mainloop()
```

На следующем этапе выполняется тестирование работы разработанного приложения, завершается физическое проектирование системы.

Опыт применения общих принципов обучения разработке информационных систем компьютерного тестирования для будущих преподавателей информатики в Курском государственном университете подтверждает возможности углубленного формирования их профессиональных компетенций.

Список литературы

1. Тарасюк В.Б. Педагогическая модель реализации системно-деятельностного подхода при подготовке преподавателей информатики в условиях цифровой трансформации / В.Б. Тарасюк, Е.И. Травкин // Педагогика и психология как ресурс развития современного общества: сборник материалов Междунар. науч.-практ. конф (Чебоксары, 25 дек. 2021 г.) / редкол.: Ж.В. Мурзина [и др.]. – Чебоксары: Среда, 2021. – С. 126–128. – ISBN 978-5-907561-00-7

2. Матушанский Г.У. Педагогическое тестирование в России / Г.У. Матушанский // Педагогика. – 2002. – №2. – С. 15–21.

3. Программирование на Python / пер. с англ. – 4-е изд. – Т. I. – СПб.: Символ-Плюс, 2011. – 992 с.