Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

Рязанский государственный радиотехнический университет имени В.Ф. Уткина

Кафедра САПР ВС

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

по дисциплине

«Программирование на Python»

Тема:

«Клавиатурный тренажёр»

|  |  |  |
| --- | --- | --- |
| Выполнил студент группы 247 Балоян А.А. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  дата сдачи на проверку, подпись | |
| Руководитель работы доц. каф. САПР ВС Иванчикова М.А. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ оценка | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  дата защиты, подпись |

Оглавление

[**Введение** 2](#_Toc154355087)

[**1.** **Постановка задачи** 3](#_Toc154355088)

[**2.** **Разработка алгоритмов** 4](#_Toc154355089)

[**3.** **Разработка программы** 9](#_Toc154355090)

[**4. Экспериментальная проверка программы** 11](#_Toc154355091)

[**Заключение** 13](#_Toc154355092)

[**Приложение А** 14](#_Toc154355093)

## **Введение**

Курсовая работа (КР) представляет собой самостоятельную работу по теме «Клавиатурный тренажёр» написанную на языке Python с использованием библиотеки pygame.

Клавиатурный тренажер – это компьютерная программа, созданная для развития и улучшения навыков работы с клавиатурой. Он активно применяется в различных сферах: от обучения десятипальцевому методу печати до тренировки быстроты и точности набора текста. В настоящее время клавиатурные тренажеры являются востребованным инструментом, особенно в условиях цифровизации образования.

Целью данной курсовой работы является разработка клавиатурного тренажера с использованием библиотеки Pygame. Pygame предоставляет широкие возможности для создания игровых приложений и графических интерфейсов на языке программирования Python.

Разработанный клавиатурный тренажер будет иметь простой и интуитивно понятный интерфейс, что позволит использовать его как начинающими, так и опытными пользователями.

# **Постановка задачи**

Целью курсовой работы являются разработка и отладка приложения,

написанного на языке программирования Python.

Разрабатываемая программа будет работать следующим образом:

1. Пользователь нажимает кнопку “Играть”.

2. Из текстового файла “russian.txt” берётся случайное слово и выводится на экран, начинается 15-ти секундный таймер.

3. Пользователь вводит буквы этого слова, и в зависимости от их соответствия изменяется его счёт и кол-во ошибок.

4. Если слово закончилось, то алгоритм повторяется, пока не истечёт 15-ти секундный таймер.

5. По окончании таймера игра закончится и на экран будут выведены счёт, кол-во ошибок, рекорд и выбор “Повторить попытку” или “Выйти”.

6. Если выбрано “Повторить попытку”, то повторяется алгоритм 2-5.

# **Разработка алгоритмов**

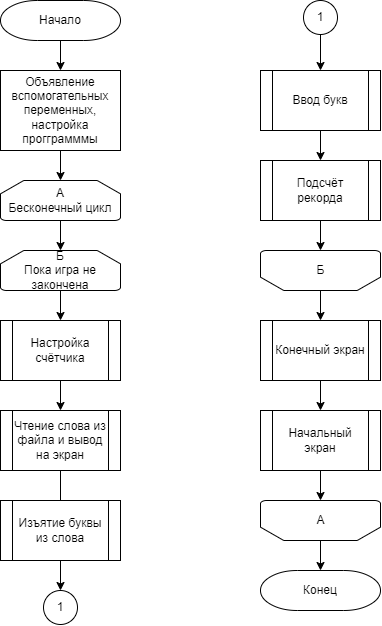
****

Рисунок 1 – Алгоритм основной программы

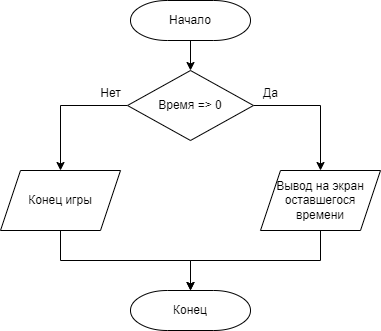
****

Рисунок 2 – Алгоритм настройки счётчика

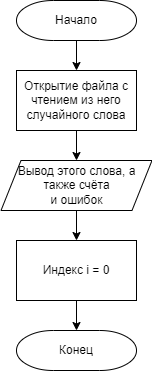
****

Рисунок 3 – Алгоритм чтения слова из файла и его вывода на экран

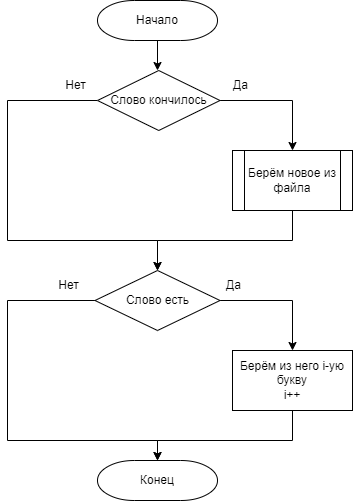
****

Рисунок 4 – Алгоритм изъятия буквы из слова

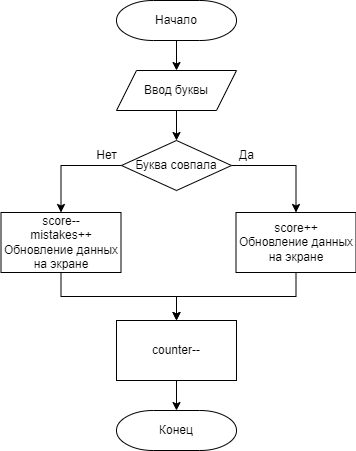
****

Рисунок 5 – Алгоритм ввода буквы

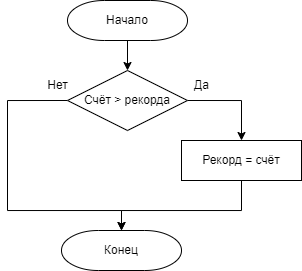
****

Рисунок 6 – Алгоритм подсчёта рекорда

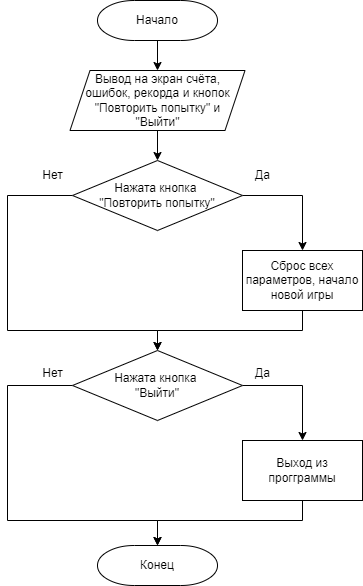
****

Рисунок 7 – Алгоритм конечного экрана

****

Рисунок 8 – Алгоритм начального экрана

# **Разработка программы**

Программа разрабатывалась на языке программирования Python. Он широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. При разработке активно использовалась библиотека pygame.

Pygame - это библиотека для разработки компьютерных игр и мультимедийных приложений, написанная на языке программирования Python. Она предоставляет удобные инструменты для создания графического интерфейса, обработки изображений, воспроизведения звуков, анимаций и работы с вводом управления.

Метод работы программы:

- С помощью функции StarrtGame() на экран выводится кнопка “Играть”, нажав на которую запустится цикл, в котором реализована работа клавиатурного тренажёра.

- Пока оставшееся на счётчике время не меньше нуля, оно будет выводиться на экран.

- Если нет слова, с которым можно работать, т.е. flag\_word = False, то открываем файл и читаем из него случайное слово, объявляем поверхности слова, счёт, ошибок для последующего вывода на экран.

- Если уже было проведено сравнение для буквы с клавиатуры или она ещё не выбрана (flag\_letter = False), то проверяем кончилось ли слово, если нет, то берём следующую её букву, если да – берём новое слово из файла.

- После выбора буквы проверяется совпадает ли введённое с клавиатуры пользователем значение, если да, то к счёту прибавляется 1 очко, слово красится в зелёный цвет и пропадает первая буква, если нет, то слово красится в красный, счёт убавляется на 1 очко, а к ошибкам прибавляется, также пропадает первая буква.

- После того, как 15-ти секундный таймер закончится, считается рекорд, и выводится конечный экран с помощью EndGame(score, mistakes, top\_score). На нём будут изображены счёт, кол-во ошибок и рекорд пользователя. Также представлены кнопки “Повторить попытку” и “Выйти”.

- При нажатии на кнопку “Повторить попытку” повторится цикл, в котором реализована работа клавиатурного тренажёра.

- При нажатии на кнопку “Выйти” приложение завершит работу.

# **4. Экспериментальная проверка программы**

Изображение выглядит как текст, снимок экрана, Графика

Автоматически созданное описание

Рисунок 9 -Начальный экран программы

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 10 – Вывод слова на экран

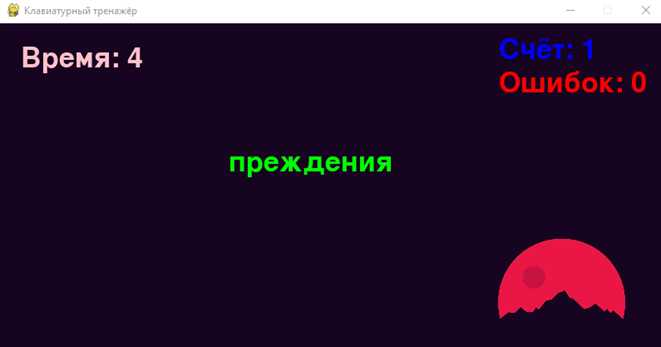


Рисунок 11 – Ввод верной буквы

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рисунок 12 – Ввод неверной буквы

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Рисунок 13 – Конечный экран программы

# **Заключение**

В ходе данной курсовой работы были получены навыки разработки самостоятельных программ. Улучшились общие навыки в программировании. Была сделана работающая программа, прошедшая полную отладку и готовая к эксплуатации.

# **Приложение А**

import pygame  
import random  
from sys import exit  
  
pygame.init()  
W = 800  
L = 400  
score, mistakes = 0, 0  
top\_score = 0  
flag\_letter, flag\_word, flag\_game\_over, flag\_first\_game = False, False, True, True  
  
counter, text = 1, '1'.rjust(3)  
pygame.time.set\_timer(pygame.USEREVENT, 1000)  
screen = pygame.display.set\_mode((W, L))  
test\_font = pygame.font.Font(None, 50)  
pygame.display.set\_caption("Клавиатурный тренажёр")  
clock = pygame.time.Clock()  
  
fon\_surface = pygame.image.load('graphics/fon.jpg').convert\_alpha()  
new\_fon = pygame.transform.scale(fon\_surface, (400, 200))  
fon\_rect = new\_fon.get\_rect(bottomright=(W, L))  
  
  
def EndGame(score, mistakes, top\_score):  
 score\_surface = test\_font.render("Ваш счёт: " + str(score), True, 'Blue')  
 screen.blit(score\_surface, (300, 50))  
 mistake\_surf = test\_font.render('Ошибок: ' + str(mistakes), True, 'Red')  
 screen.blit(mistake\_surf, (300, 100))  
 top\_score\_surf = test\_font.render("Рекорд: " + str(top\_score), True, "Gold")  
 screen.blit(top\_score\_surf, (600, 15))  
 yes\_surf = test\_font.render("Повторить попытку", True, "Green")  
 no\_surf = test\_font.render("Выйти", True, "Red")  
 yes\_rect = yes\_surf.get\_rect(midleft=(250, 200))  
 screen.blit(yes\_surf, yes\_rect)  
 no\_rect = no\_surf.get\_rect(midright=(450, 300))  
 screen.blit(no\_surf, no\_rect)  
 if (yes\_rect.collidepoint(mouse\_pos) and pygame.mouse.get\_pressed(3) == (True, False, False)) or (  
 event.type == pygame.KEYDOWN and event.key == pygame.K\_RETURN): # Повторить попытку  
 return True  
 if no\_rect.collidepoint(mouse\_pos) and pygame.mouse.get\_pressed(3) == (True, False, False): # Выйти  
 pygame.quit()  
 exit()  
  
  
def StartGame():  
 text\_surface = test\_font.render("Играть", True, "White")  
 text\_rect = text\_surface.get\_rect(center=(W//2, L//2))  
 screen.fill((23,4,32)) # Темно-фиолетовый цвет  
 screen.blit(new\_fon, fon\_rect)  
 screen.blit(text\_surface, text\_rect)  
 pygame.display.flip()  
 if text\_rect.collidepoint(mouse\_pos) and pygame.mouse.get\_pressed(3) == (True, False, False):  
 return True  
  
  
while True:  
 mouse\_pos = pygame.mouse.get\_pos()  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT:  
 pygame.quit()  
 exit()  
  
 while flag\_game\_over == False:  
 if counter >= 0: # Настраиваем счётчик  
 text = str(counter).rjust(3)  
 time\_surf = test\_font.render('Время: ' + str(counter), True, 'Pink')  
 else:  
 flag\_game\_over = True  
  
 if flag\_word == False: # Читаем слово из файла, выводим его со счётом и ошибками на экран  
 word = random.choice(open("russian.txt", encoding='utf-8').readlines())  
 text\_surface = test\_font.render(word[:-1], True, 'White')  
 screen.blit(text\_surface, (275, 150))  
 score\_surf = test\_font.render('Счёт: ' + str(score), False, 'Blue')  
 mistake\_surf = test\_font.render('Ошибок: ' + str(mistakes), True, 'Red')  
 flag\_word = True  
 i = 0  
  
 if flag\_letter == False:  
 if (i == len(word[:-1])): # Если слово кончилось  
 flag\_word = False  
 if (flag\_word): # Берём букву из слова  
 letter = word[i]  
 flag\_letter = True  
 i += 1  
  
 for event in pygame.event.get(): # Ввод с клавиатуры  
 if event.type == pygame.QUIT:  
 pygame.quit()  
 exit()  
 if event.type == pygame.KEYDOWN:  
 if event.unicode == letter.lower(): # Буква совпала  
 score += 1  
 text\_surface = test\_font.render(word[i:-1], True, 'Green')  
 score\_surf = test\_font.render('Счёт: ' + str(score), True, 'Blue')  
 flag\_letter = False  
 else: # Не совпала  
 score -= 1  
 mistakes += 1  
 text\_surface = test\_font.render(word[i:-1], True, 'Red')  
 score\_surf = test\_font.render('Счёт: ' + str(score), True, 'Blue')  
 mistake\_surf = test\_font.render('Ошибок: ' + str(mistakes), True, 'Red')  
 flag\_letter = False  
 if event.type == pygame.USEREVENT:  
 counter -= 1  
  
 if score > top\_score: # Подсчёт рекорда  
 top\_score = score  
  
 screen.fill((18, 6, 31)) # Темно-фиолетовый цвет  
 screen.blit(new\_fon, fon\_rect)  
 screen.blit(text\_surface, (275, 150))  
 screen.blit(score\_surf, (600, 15))  
 screen.blit(mistake\_surf, (600, 55))  
 screen.blit(time\_surf, (25, 25))  
 pygame.display.flip()  
 clock.tick(60)  
  
 screen.fill((18, 6, 31)) # Темно-фиолетовый цвет  
 screen.blit(new\_fon, fon\_rect)  
 if flag\_game\_over and flag\_first\_game == False: # Конечный экран  
 if (EndGame(score, mistakes, top\_score)):  
 counter = 15  
 score = 0  
 mistakes = 0  
 flag\_letter = False  
 flag\_word = False  
 flag\_game\_over = False  
  
 if flag\_game\_over and flag\_first\_game: # Начальный экран  
 if (StartGame()):  
 counter = 15  
 flag\_game\_over = False  
 flag\_first\_game = False  
  
 pygame.display.update()  
 clock.tick(60)