

Программный комплекс на Python для мониторинга и развития навыков клавиатурного ввода.

*Языков М.Д.,
Российский университет дружбы народов, г. Москва,
ул. Миклухо-Маклая, 6.
тел. +7-927-210-21-46
E-mail: yazykov_m@inbox.ru*

Информатика является самой перспективной и быстроразвивающейся областью в современном мире. Умение и навыки работы с информационными средствами являются необходимым условием для успешного овладения программ среднего и высшего образования, а также для приобретения перспективной и востребованной профессией. В связи с этим, увеличивается значимость умения работы с электронным текстом и таким устройством ввода, как клавиатура.

Навык быстрой печати необходим каждому человеку, особенно ученику и студенту, так как именно у них возникает необходимость в наборе рефератов, докладов и научных работ. Для повышения производительности и ускорения работы необходимо тренировать скорость набора текста. Одним из самых эффективных способов овладения навыком является постоянная практика. Для данной цели рекомендуется использовать программы по набору текстов.

На данный момент существует достаточно большое количество программ, направленных на выполнение данной задачи. В ходе анализа существующих программных продуктов по набору текста выявлены следующие недостатки:

1. Перед выполнением авторами зачастую рекомендуется прочитка специальных текстов, которые предшествуют упражнениям. Данный факт не нравится многим пользователям, так как тормозит выполнение практических заданий

2. Задания можно выполнять исключительно по порядку

На основании данного анализа был сделан вывод о целесообразности создания собственной программы. Целью данной научной работы является разработка и реализация программы по тестированию скорости набора текста на языке Python.

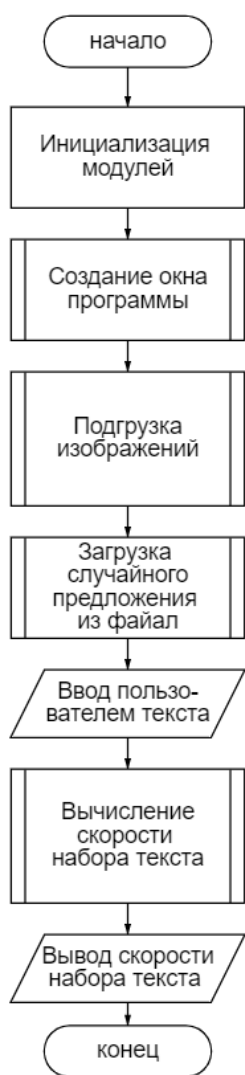
Для разработки программы был выбран объектно-ориентированный подход так как он поддерживает принцип модульности.

Алгоритм работы программы подразделяется на три этапа:

- 1) Начальный. Подготовка к работе программы, загрузка из файлов текста для набора
- 2) Основной. Пользователь набирает слова
- 3) Заключительный. Производится запись данных в файл рекордов в случае, если пользователь показал один из трех лучших результатов за все время работы с данной программой. В этом случае производится запись имени с клавиатуры в файл рекордов.

В ходе разработки была создана блок-схема с описанием основных этапов работы программы (см. рис. 1).

В начале происходит импорт всех необходимых для работы модулей. Для создания графического интерфейса используется библиотека `pygame`, которую используют для работы с графикой [1]. При помощи данной библиотеки текст и изображение отображаются на экране. Далее идет создание окна программы, этот этап реализован при помощи функции. В рамках подготовительного этапа идет загрузка изображений при помощи модуля `pygame`. Для создания текста, который служит в качестве образца, используют текстовый файл, из которого случайным образом при помощи модуля `random` забирается любое предложение. Данный файл можно редактировать, чтобы была возможность тренировки набора различных текстов.



```

speed_typing.py - C:\Users\79272\Desktop\speed_typing.py (3.9.6)
File Edit Format Run Options Window Help
import pygame
from pygame.locals import *
import sys
import time
import random

# 750 x 500

class Game:

    def __init__(self):
        self.w=750
        self.h=500
        self.reset=True
        self.active = False
        self.input_text=''
        self.word = ''
        self.time_start = 0
        self.total_time = 0
        self.accuracy = '0%'
        self.results = 'Time:0 Accuracy:0 % Wpm:0 '
        self.wpm = 0
        self.end = False
        self.HEAD_C = (255,213,102)
        self.TEXT_C = (240,240,240)
        self.RESULT_C = (255,70,70)

    pygame.init()
    self.open_img = pygame.image.load('type-speed-open.png')
    self.open_img = pygame.transform.scale(self.open_img, (sel

    self.bg = pygame.image.load('background.jpg')
    self.bg = pygame.transform.scale(self.bg, (500,750))

    self.screen = pygame.display.set_mode((self.w,self.h))
    pygame.display.set_caption('Type Speed test')

    def draw_text(self, screen, msg, y ,fsize, color):
        font = pygame.font.Font(None, fsize)
  
```

Рис.1 Блок-схема программы

После отображения на экране образца текста, пользователю предоставляется возможность самому напечатать текст. Далее происходит расчет скорости набора текста при помощи функции. В данной функции фиксируется время начала ввода текста и время его окончания, рассчитывается общее время расчета текста, а также среднее количество слов в минуту (за среднее значение символов в слове выбрана 5). Более того, текст проверяется на ошибки путем сравнения введенного пользователем текста и эталона предложения и вычисляется значение точности набора текста. На заключительном этапе происходит вывод результатов работы программы на экран. Так же в программе имеется возможность сброса данных и ее повторного запуска при помощи кнопки Reset.

Программа была протестирована, в конечной версии программы не обнаружены ошибки, в последствии планируется усовершенствование данной программы путем подсвечивания мест, где были допущены ошибки, а также добавления советов по усовершенствованию показателей скорости и точности набора текста.

Список литературы:

1. Щерба А.В. Программирование на Python®: первые шаги / Щерба А.В.. — Москва: Лаборатория знаний, 2022. — 251 с. — ISBN 978-5-93208-578-3. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/120878.html> (дата обращения: 11.11.2022). — Режим доступа: для авторизир. пользователей