МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Балтийский федеральный университет имени Иммануила Канта»

ОНК «Институт высоких технологий»

ОТЧЁТ

о прохождении учебной практики по получению первичных

профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности

на базе Высшей школы компьютерных наук и прикладной математики образовательно-научного кластера "Институт высоких технологий"

Выполнил Тронина София Александровна \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студентка очной формы обучения 3 курса

специальности 10.05.01 Компьютерная безопасность

специализация «Математические методы защиты информации»

Руководитель практики от университета

доцент ОНК «ИВТ» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Киршанова Е.А.

г. Калининград 2022 г.

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc107438183)

[Глава 1. ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРАКТИКУ 4](#_Toc107438184)

Глава 2. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ В PYTHON. [4](#_Toc107438186)

ТЕОРИЯ. [4](#_Toc107438186)

[ПРАКТИЧЕСКАЯ ЧАСТЬ 5](#_Toc107438187)

[Глава 3. ВЕРСТКА ПРЕЗЕНТАЦИИ В СИСТЕМЕ LATEX 9](#_Toc107438188)

[Заключение 10](#_Toc107438190)

[Список литературы 11](#_Toc107438191)

## Введение

Вид практики – учебная практика по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности, далее Учебная практика.

Цель учебной практики: практические навыки владения LaTeX и объектно-ориентированного программирования в Python, а также улучшить навыки публичного выступления и защиты работы.

Задачи учебной практики:

* Запрограммировать одну из игр на языке Python, желательно используя объектно-ориентированное программирование, представить реализацию.
* Составить презентацию, объясняющую суть игры, методы реализации, используя систему верстки LaTeX.
* Подготовить 10 минутное выступление, опирающееся на презентацию. Ответить на возможные вопросы касательно решения.

## Глава 1. ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРАКТИКУ

**Задание 1.** Выбрать и запрограммировать одну из игр.

На реализацию была выбрана игра «Слова из слова».

Суть самой игры заключается в следующем: Рандомно выбирается слово, из которого можно составить другие слова. Первым ходит игрок. После того как он записал свое слово, компьютер составляет свое.

Каждая буква может встречаться в слове один раз и слова не могут повторяться. Минимальная длина слова - 3 буквы. За каждую использованную букву дается 1 балл.

Игра заканчивается, если все возможные слова были введены или игрок нажал на кнопку "Закончить игру".

Игрок выигрывает, если у него на счету больше баллов, чем у компьютера.

**Задание 2.** Вспомнить, как реализуется объектно-ориентированное программирование в Python.

**Задание 3.** Закрепить знания об верстки документов с помощью системы LaTeX.

**Задание 4.** Составить на их основе презентацию.

## Глава 2. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ В PYTHON.

**ТЕОРИЯ**

**Объектно-ориентированное программирование (ООП)** - это подход к программированию, основанный на **объектах** и **классах**.

Объектно-ориентированное программирование основано на следующих ключевых концепциях:

* Классы и объекты
* Наследование
* Инкапсуляция
* Полиморфизм
* Абстрактные классы

**Класс - это шаблон или схема, из которой будут создаваться экземпляры объектов.** Это похоже на форму, из которой могут быть созданы объекты. Класс состоит из набора состояний (также известных как **атрибуты** или **свойства**) и **поведений** (также известных как **методы**).

**Объект** - это **экземпляр класса.**Атрибуты объекта, сохраненные с использованием переменных для хранения данных, и поведение объекта реализованы в виде функций или процедур и могут использоваться для выполнения операций с данными.

**Наследование** — свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствованной функциональностью.

**Инкапсуляция**— свойство системы, позволяющее объединить данные и методы, работающие с ними, в классе.

**Полиморфизм подтипов** (либо просто полиморфизм) — свойство системы, позволяющее использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта.

**Абстрагирование** означает выделение значимой информации и исключение из рассмотрения незначимой.

В представленной программе будут использоваться некоторые из концепций, рассказанных выше. А именно – классы и объекты, поэтому о них далее будет идти речь.

В Python класс определяется с помощью ключевого слова – *class*. Это похоже на то, как определяются функции с помощью *def*. Как только класс определен, его можно вызывать. Вызов класса немного отличается от вызова функции. Когда вызывается функция, на выходе получается возвращаемое значение функции. Когда вызывается класс, на выходе получается **объект,** тип**которого соответствует этому классу**.

Действия над классом выполняются с помощью методов. **Метод**- это функция, которая живет в классе и, в частности, **работает с экземплярами этого класса**. Методы, как правило, либо получают доступ к данным из экземпляра класса, либо изменяют данные в экземпляре класса.

Прежде чем создавать какие-либо новые методы в классе, необходимо создать метод \_\_init\_\_. Он позволяет принимать аргументы нового класса. Что еще более важно, данный метод позволяет присваивать начальные значения различным атрибутам в экземплярах класса.

**ПРАКТИЧЕСКАЯ ЧАСТЬ**

После импорта всех нужных библиотек (tkinter, random, messagebox).

Необходимо выбрать основное слово.

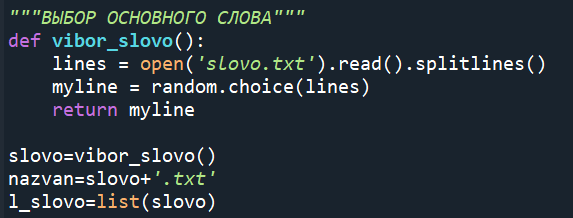


Рисунок 1. Функция для поиска основного слова

Также необходимо ввести некоторые дополнительные пометки:

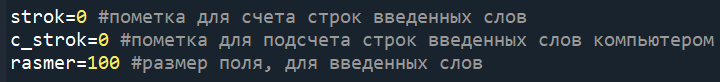
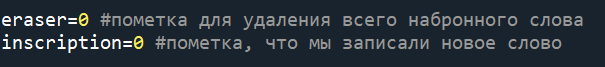


Рисунок 2. Дополнительные пометки

Затем создадим класс для кнопок, где пропишем кнопки для ввода букв, и также дополнительные кнопки "Записать" и "Очистить".

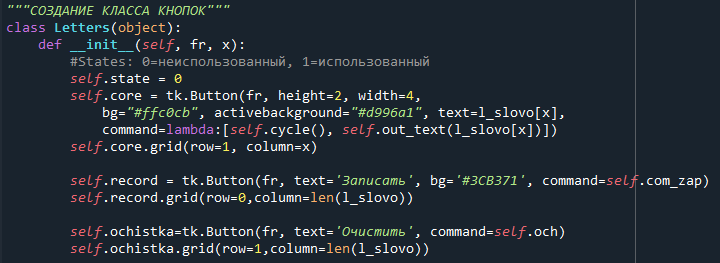


Рисунок 3. Создание класса кнопок

У кнопок-букв присутствует две команды. Первая отвечает за изменение статуса кнопки, а вторая - вывод буквы на экран:

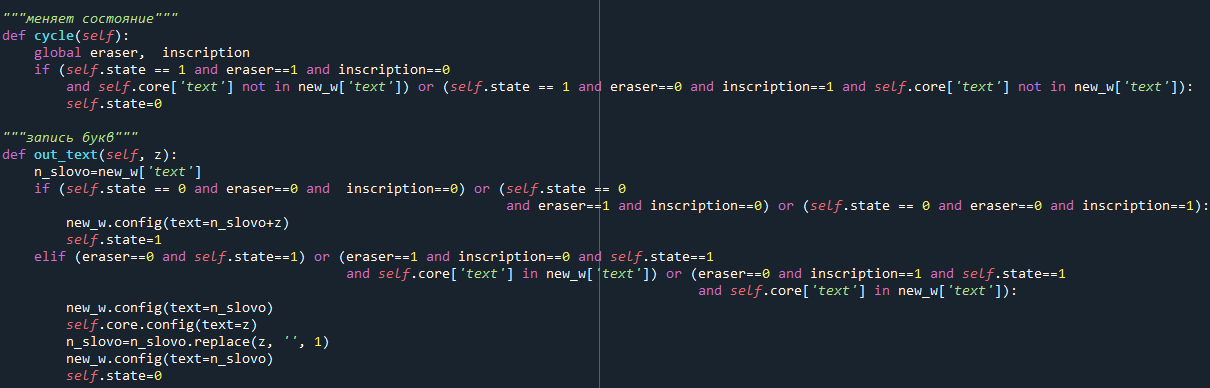


Рисунок 4. Команды для кнопок-букв

Кнопка "Записать" имеет одну команду, но она включает в себя несколько подпунктов: запись слова игрока и компьютера, подсчет баллов, появления информационных окон в случаях, если введенное слово уже присутствует на экране или все возможные слова были введены.

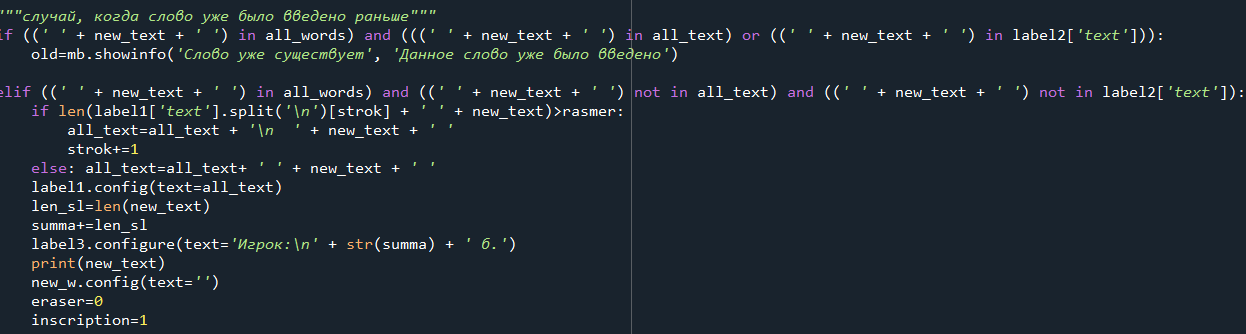


Рисунок 5. Первая часть команды для кнопки "Записать"

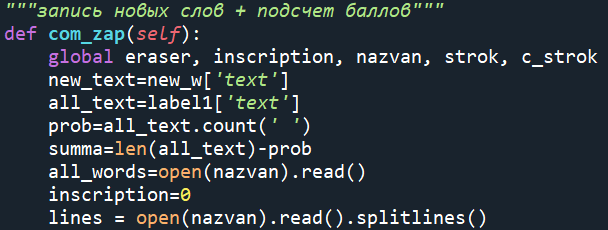


Рисунок 6. Вторая часть команды для кнопки "Записать"

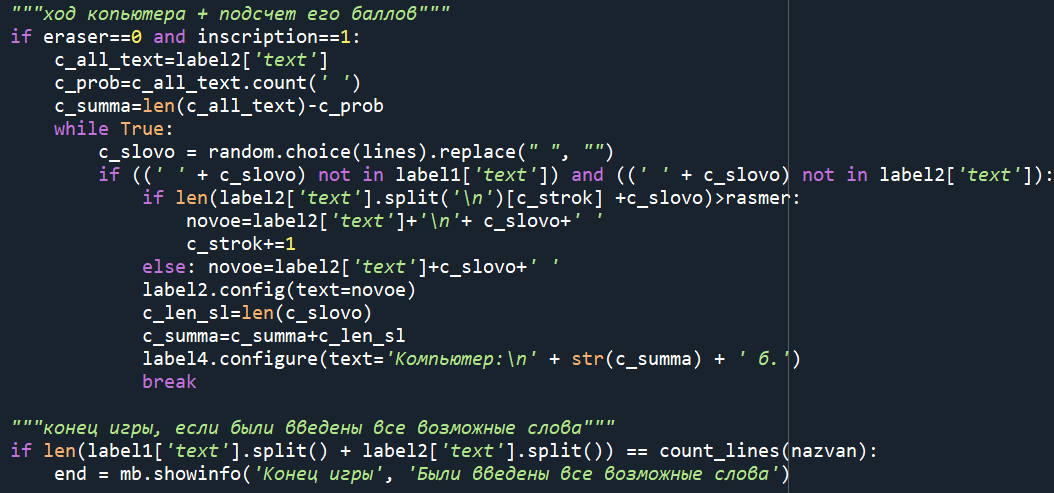


Рисунок 7. Третья часть команды для кнопки "Записать"

Количество возможных слов считается с помощью дополнительной функции:

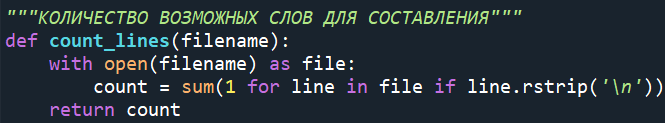


Рисунок 8. Функция для подсчета количества возможных составленных слов

Кнопка "Очистить" также имеет одну команду: она стирает написанное слово полностью:

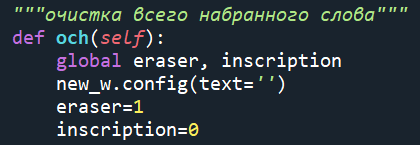


Рисунок 9. Команда для кнопки "Очистить"

Далее переходим к формированию главного окна:

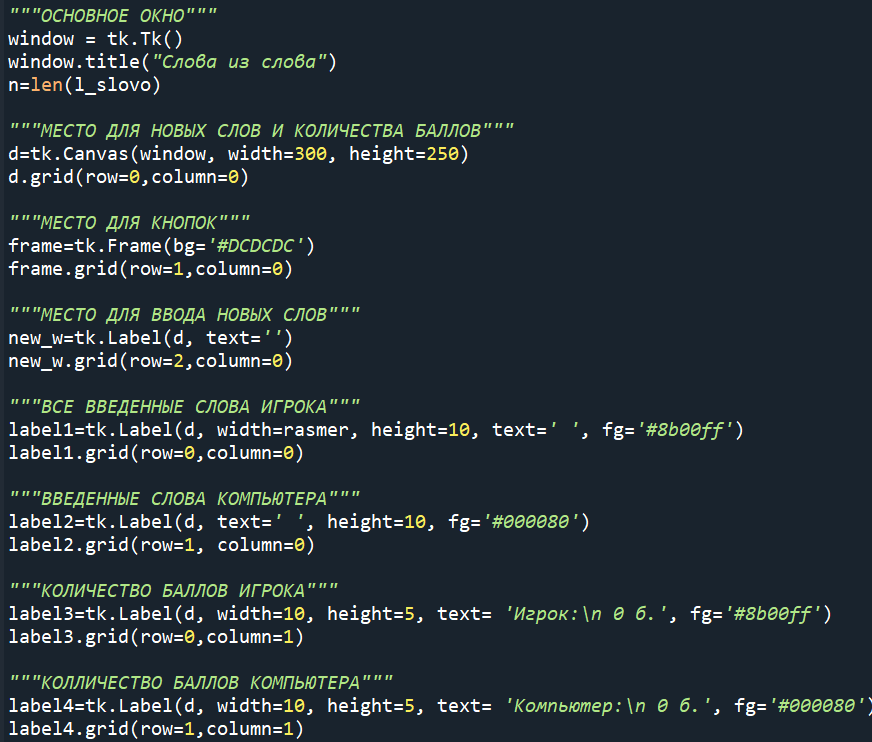


Рисунок 10. Формирование окна игры

С помощью цикла формируем буквы-кнопки:

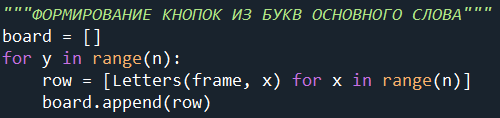


Рисунок 11. Формирование букв-кнопок

Затем необходима еще одна кнопка: "Закончить игру". Ее можно нажать в любой момент игры.

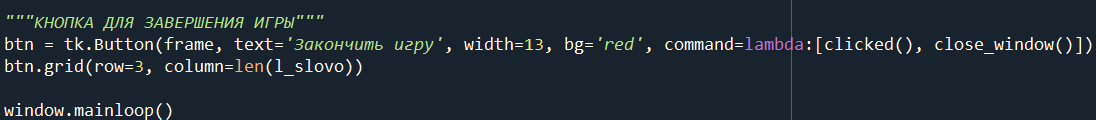


Рисунок 12. Создание кнопки "Закончить игру"

Она имеет две команды: вывод информационного окна с информацией о том, кто выиграл, и закрытие основного окна:

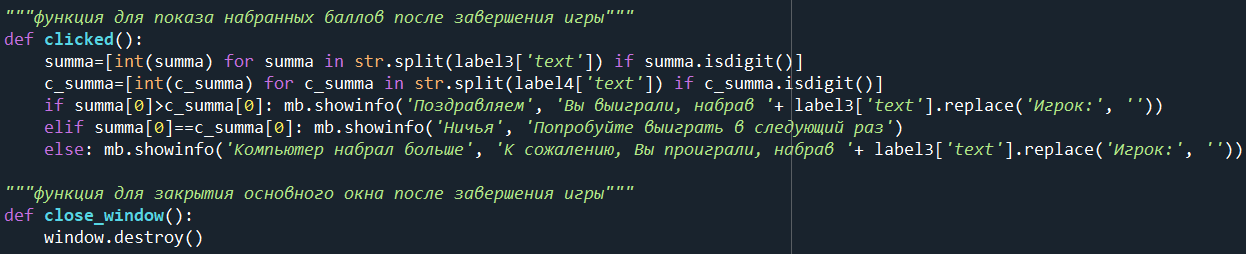


Рисунок 13. Команды для кнопки " Закончить игру "

В результате игра выглядит следующим образом:

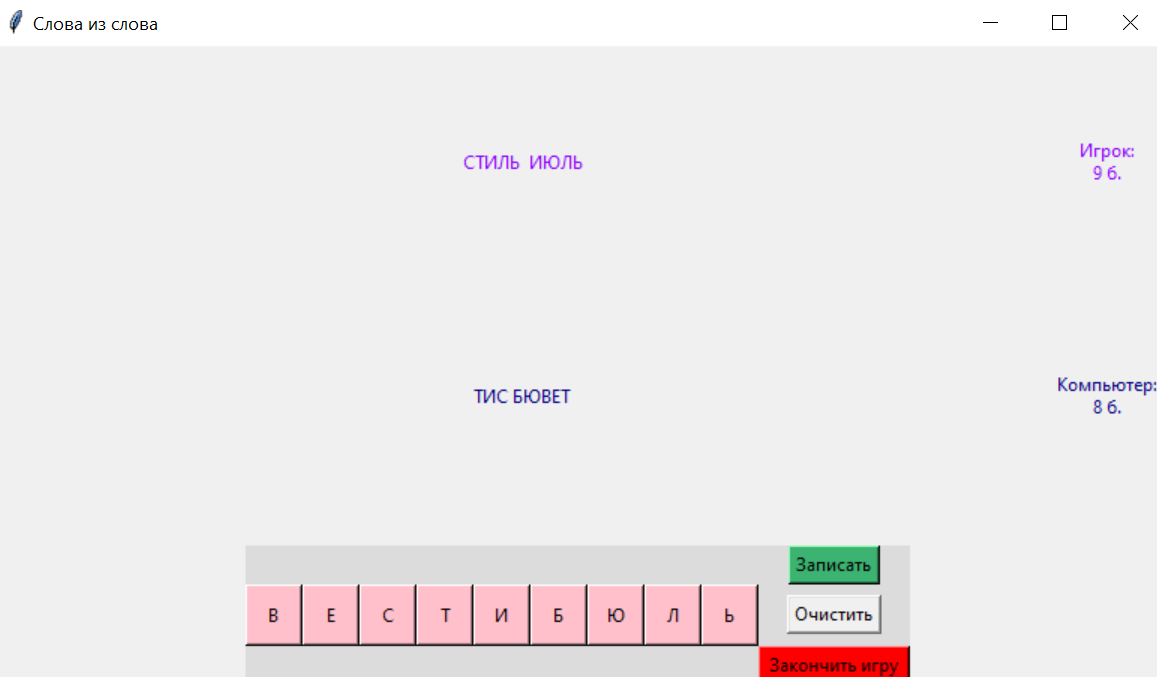


Рисунок 14. Вид игры

## Глава 3. ВЕРСТКА ПРЕЗЕНТАЦИИ В СИСТЕМЕ LATEX

С помощью программы LaTeX была составлена презентация, которая описывает суть и этапы работы программы. Был использован пакет Beamer.

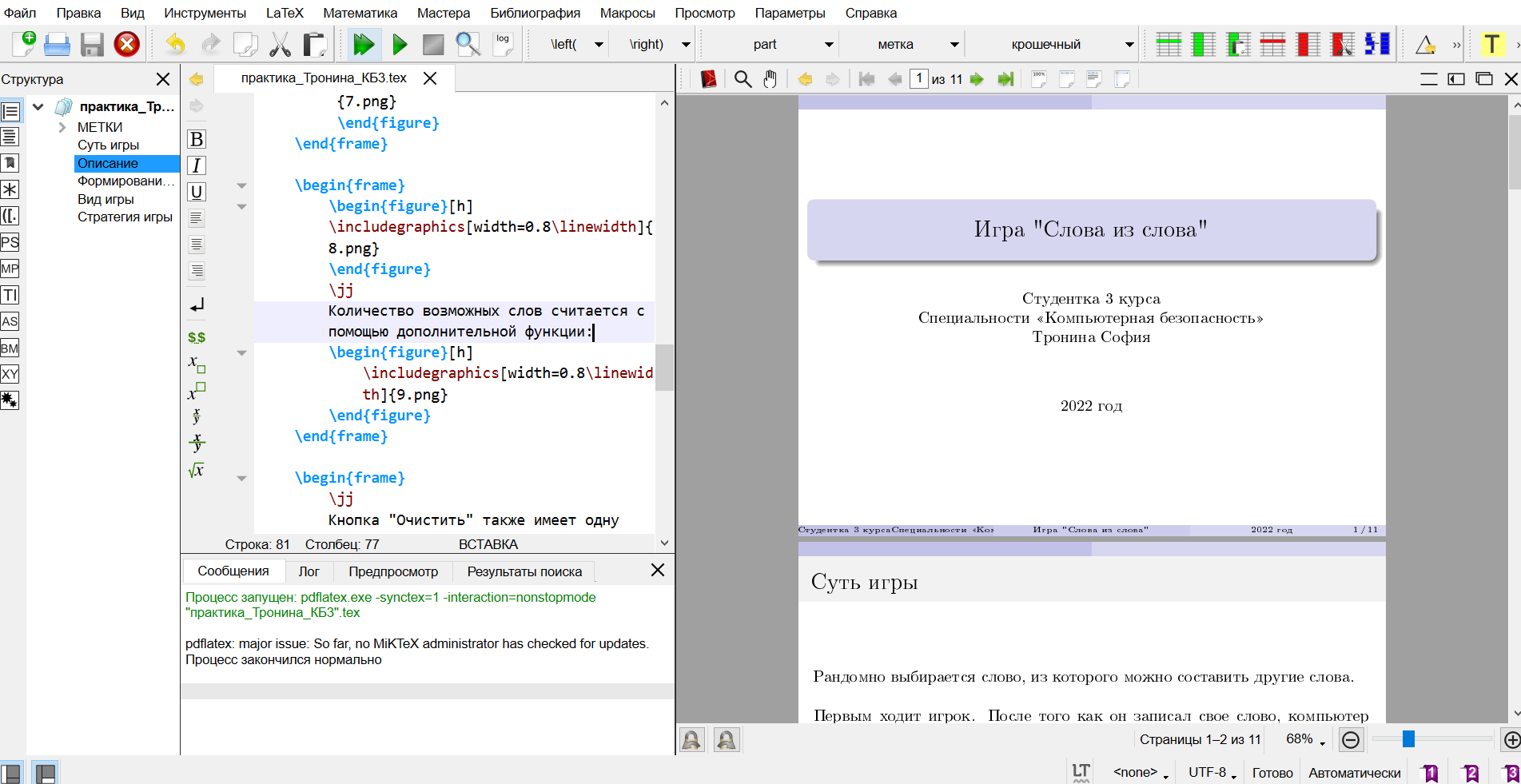


Рисунок 15. Пример интерфейса

## Заключение

В течение учебной практики все поставленные задачи были выполнены, а цели достигнуты. Я укрепила свои навыки программирования на Python и создания презентаций на LaTeX. Помимо этого, я отработала навыки самостоятельной работы и поиска информации, а также навыки публичного выступления, что, безусловно, благоприятно скажется на мою дальнейшую научную, и не только, деятельность.

В целом можно сказать, что система LaTeX позволила мне более гибко настраивать презентацию и достигать именно того результата, который я хочу. Структура верстки достаточно удобная и интуитивно понятная.

**Интернет – ресурсы:**

1. Классы в Python - <https://docs.python.org/3/tutorial/classes.html>
2. Концепция ООП - [https://www.101computing.net](https://www.101computing.net/object-oriented-programming-concepts/)
3. Python Morsels 3 - [https://www.pythonmorsels.com](https://www.pythonmorsels.com/articles/class/)
4. Объектно-ориентированное программирование - [https://ru.wikipedia.org](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)
5. Верстка презентации -

**Приложения**

Приложение 1

import tkinter as tk

import random

from tkinter import messagebox as mb

"""ВЫБОР ОСНОВНОГО СЛОВА"""

def vibor\_slovo():

lines = open('slovo.txt').read().splitlines()

myline = random.choice(lines)

return myline

slovo=vibor\_slovo()

nazvan=slovo+'.txt'

l\_slovo=list(slovo)

strok=0 #пометка для счета строк введенных слов

c\_strok=0 #пометка для подсчета строк введенных слов компьютером

rasmer=100 #размер поля, для введенных слов

"""КОЛИЧЕСТВО ВОЗМОЖНЫХ СЛОВ ДЛЯ СОСТАВЛЕНИЯ"""

def count\_lines(filename):

with open(filename) as file:

count = sum(1 for line in file if line.rstrip('\n'))

return count

eraser=0 #пометка для удаления всего набронного слова

inscription=0 #пометка, что мы записали новое слово

"""СОЗДАНИЕ КЛАССА КНОПОК"""

class Letters(object):

def \_\_init\_\_(self, fr, x):

#States: 0=неиспользованный, 1=использованный

self.state = 0

self.core = tk.Button(fr, height=2, width=4,

bg="#ffc0cb", activebackground="#d996a1", text=l\_slovo[x],

command=lambda:[self.cycle(), self.out\_text(l\_slovo[x])])

self.core.grid(row=1, column=x)

self.record = tk.Button(fr, text='Записать', bg='#3CB371', command=self.com\_zap)

self.record.grid(row=0,column=len(l\_slovo))

self.ochistka=tk.Button(fr, text='Очистить', command=self.och)

self.ochistka.grid(row=1,column=len(l\_slovo))

"""меняет состояние"""

def cycle(self):

global eraser, inscription

if (self.state == 1 and eraser==1 and inscription==0 and self.core['text'] not in new\_w['text']) or (self.state == 1 and eraser==0 and inscription==1 and self.core['text'] not in new\_w['text']):

self.state=0

"""запись букв"""

def out\_text(self, z):

n\_slovo=new\_w['text']

if (self.state == 0 and eraser==0 and inscription==0) or (self.state == 0 and eraser==1 and inscription==0) or (self.state == 0 and eraser==0 and inscription==1):

new\_w.config(text=n\_slovo+z)

self.state=1

elif (eraser==0 and self.state==1) or (eraser==1 and inscription==0 and self.state==1 and self.core['text'] in new\_w['text']) or (eraser==0 and inscription==1 and self.state==1 and self.core['text'] in new\_w['text']):

new\_w.config(text=n\_slovo)

self.core.config(text=z)

n\_slovo=n\_slovo.replace(z, '', 1)

new\_w.config(text=n\_slovo)

self.state=0

"""запись новых слов + подсчет баллов"""

def com\_zap(self):

global eraser, inscription, nazvan, strok, c\_strok

new\_text=new\_w['text']

all\_text=label1['text']

prob=all\_text.count(' ')

summa=len(all\_text)-prob

all\_words=open(nazvan).read()

inscription=0

lines = open(nazvan).read().splitlines()

"""случай, когда слово уже было введено раньше"""

if ((' ' + new\_text + ' ') in all\_words) and (((' ' + new\_text + ' ') in all\_text) or ((' ' + new\_text + ' ') in label2['text'])):

old=mb.showinfo('Слово уже существует', 'Данное слово уже было введено')

elif ((' ' + new\_text + ' ') in all\_words) and ((' ' + new\_text + ' ') not in all\_text) and ((' ' + new\_text + ' ') not in label2['text']):

if len(label1['text'].split('\n')[strok] + ' ' + new\_text)>rasmer:

all\_text=all\_text + '\n ' + new\_text + ' '

strok+=1

else: all\_text=all\_text+ ' ' + new\_text + ' '

label1.config(text=all\_text)

len\_sl=len(new\_text)

summa+=len\_sl

label3.configure(text='Игрок:\n' + str(summa) + ' б.')

print(new\_text)

new\_w.config(text='')

eraser=0

inscription=1

"""ход копьютера + подсчет его баллов"""

if eraser==0 and inscription==1:

c\_all\_text=label2['text']

c\_prob=c\_all\_text.count(' ')

c\_summa=len(c\_all\_text)-c\_prob

while True:

c\_slovo = random.choice(lines).replace(" ", "")

if ((' ' + c\_slovo) not in label1['text']) and ((' ' + c\_slovo) not in label2['text']):

if len(label2['text'].split('\n')[c\_strok] +c\_slovo)>rasmer:

novoe=label2['text']+'\n'+ c\_slovo+' '

c\_strok+=1

else: novoe=label2['text']+c\_slovo+' '

label2.config(text=novoe)

c\_len\_sl=len(c\_slovo)

c\_summa=c\_summa+c\_len\_sl

label4.configure(text='Компьютер:\n' + str(c\_summa) + ' б.')

break

"""конец игры, если были введены все возможные слова"""

if len(label1['text'].split() + label2['text'].split()) == count\_lines(nazvan):

end = mb.showinfo('Конец игры', 'Были введены все возможные слова')

"""очистка всего набранного слова"""

def och(self):

global eraser, inscription

new\_w.config(text='')

eraser=1

inscription=0

"""ОСНОВНОЕ ОКНО"""

window = tk.Tk()

window.title("Слова из слова")

n=len(l\_slovo)

"""МЕСТО ДЛЯ НОВЫХ СЛОВ И КОЛИЧЕСТВА БАЛЛОВ"""

d=tk.Canvas(window, width=300, height=250)

d.grid(row=0,column=0)

"""МЕСТО ДЛЯ КНОПОК"""

frame=tk.Frame(bg='#DCDCDC')

frame.grid(row=1,column=0)

"""МЕСТО ДЛЯ ВВОДА НОВЫХ СЛОВ"""

new\_w=tk.Label(d, text='')

new\_w.grid(row=2,column=0)

"""ВСЕ ВВЕДЕННЫЕ СЛОВА ИГРОКА"""

label1=tk.Label(d, width=rasmer, height=10, text=' ', fg='#8b00ff')

label1.grid(row=0,column=0)

"""ВВЕДЕННЫЕ СЛОВА КОМПЬЮТЕРА"""

label2=tk.Label(d, text=' ', height=10, fg='#000080')

label2.grid(row=1, column=0)

"""КОЛИЧЕСТВО БАЛЛОВ ИГРОКА"""

label3=tk.Label(d, width=10, height=5, text= 'Игрок:\n 0 б.', fg='#8b00ff')

label3.grid(row=0,column=1)

"""КОЛЛИЧЕСТВО БАЛЛОВ КОМПЬЮТЕРА"""

label4=tk.Label(d, width=10, height=5, text= 'Компьютер:\n 0 б.', fg='#000080')

label4.grid(row=1,column=1)

"""ФОРМИРОВАНИЕ КНОПОК ИЗ БУКВ ОСНОВНОГО СЛОВА"""

board = []

for y in range(n):

row = [Letters(frame, x) for x in range(n)]

board.append(row)

"""функция для показа набранных баллов после завершения игры"""

def clicked():

summa=[int(summa) for summa in str.split(label3['text']) if summa.isdigit()]

c\_summa=[int(c\_summa) for c\_summa in str.split(label4['text']) if c\_summa.isdigit()]

if summa[0]>c\_summa[0]: mb.showinfo('Поздравляем', 'Вы выиграли, набрав '+ label3['text'].replace('Игрок:', ''))

elif summa[0]==c\_summa[0]: mb.showinfo('Ничья', 'Попробуйте выиграть в следующий раз')

else: mb.showinfo('Компьютер набрал больше', 'К сожалению, Вы проиграли, набрав '+ label3['text'].replace('Игрок:', ''))

"""функция для закрытия основного окна после завершения игры"""

def close\_window():

window.destroy()

"""КНОПКА ДЛЯ ЗАВЕРШЕНИЯ ИГРЫ"""

btn = tk.Button(frame, text='Закончить игру', width=13, bg='red', command=lambda:[clicked(), close\_window()])

btn.grid(row=3, column=len(l\_slovo))

window.mainloop()

Приложение 2

\documentclass[10pt,pdf,hyperref={unicode}]{beamer}

\usepackage{lmodern}

\usepackage[T2A]{fontenc}

\usepackage[utf8]{inputenc}

\usepackage{ragged2e}

\newcommand{\jj}{\righthyphenmin=20 \justifying}

\setbeamertemplate{navigation symbols}{}

\usetheme{CambridgeUS}

\usecolortheme{seahorse}

\title{Игра "Слова из слова"}

\author{Студентка 3 курса\\Специальности «Компьютерная безопасность»\\Тронина София}

\date{2022 год}

\begin{document}

\begin{frame}

\titlepage

\end{frame}

\begin{frame}

\frametitle{Суть игры}

\jj

Рандомно выбирается слово, из которого можно составить другие слова. \\

~\\Первым ходит игрок. После того как он записал свое слово, компьютер составляет свое. Каждая буква может встречаться в слове один раз и слова не могут повторяться. Минимальная длина слова - 3 буквы. За каждую использованную букву дается 1 балл. \\

~\\Игра заканчивается, если все возможные слова были введены или игрок нажал на кнопку "Закончить игру". \\

~\\Игрок выигрывает, если у него на счету больше баллов, чем у компьютера.

\end{frame}

\begin{frame}

\frametitle{Описание}

\jj

После импорта всех нужных библиотек (tkinter, random, messagebox).

Необходимо выбрать основное слово.

\begin{figure}[h]

\includegraphics[width=0.7\linewidth]{1.png}

\label{fig:mpr}

\end{figure}

\jj

Также необходимо ввести некоторые дополнительные пометки:

\begin{figure}[h]

\includegraphics[width=0.8\linewidth]{2.png}

\includegraphics[width=0.8\linewidth]{3.png}

\end{figure}

\end{frame}

\begin{frame}

\jj

Затем создадим класс для кнопок, где пропишем кнопки для ввода букв, и также дополнительные кнопки "Записать" и "Очистить".

\begin{figure}[h]

\includegraphics[width=0.7\linewidth]{4.png}

\end{figure}

\jj

У кнопок-букв присутствует две команды. Первая отвечает за изменение статуса кнопки, а вторая - вывод буквы на экран:

\begin{figure}[h]

\includegraphics[width=0.9\linewidth]{5.png}

\end{figure}

\end{frame}

\begin{frame}

\jj

Кнопка "Записать" имеет одну команду, но она включает в себя несколько подпунктов: запись слова игрока и компьютера, подсчет баллов, появления информационных окон в случаях, если введенное слово уже присутствует на экране или все возможные слова были введены.

\begin{figure}[h]

\includegraphics[width=0.8\linewidth]{6.png}

\includegraphics[width=0.9\linewidth]{7.png}

\end{figure}

\end{frame}

\begin{frame}

\begin{figure}[h]

\includegraphics[width=0.8\linewidth]{8.png}

\end{figure}

\jj

Количество возможных слов считается с помощью дополнительной функции:

\begin{figure}[h]

\includegraphics[width=0.8\linewidth]{9.png}

\end{figure}

\end{frame}

\begin{frame}

\jj

Кнопка "Очистить" также имеет одну команду: она стирает написанное слово полностью:

\begin{figure}[h]

\includegraphics[width=0.7\linewidth]{10.png}

\end{figure}

\end{frame}

\begin{frame}

\frametitle{Формирование окна}

\jj

Теперь необходимо приступить к оформлению самого окна игры:

\begin{figure}[h]

\includegraphics[width=0.7\linewidth]{11.png}

\end{figure}

\end{frame}

\begin{frame}

\jj

С помощью цикла формируем буквы-кнопки:

\begin{figure}[h]

\includegraphics[width=0.7\linewidth]{12.png}

\end{figure}

\jj

Затем необходима еще одна кнопка: "Закончить игру". Ее можно нажать в любой момент игры.

\begin{figure}[h]

\includegraphics[width=1.0\linewidth]{13.png}

\end{figure}

\jj

Она имеет две команды: вывод информационного окна с информацией о том, кто выиграл, и закрытие основного окна:

\begin{figure}[h]

\includegraphics[width=1.0\linewidth]{14.png}

\end{figure}

\end{frame}

\begin{frame}

\frametitle{Вид игры}

\begin{figure}[h]

\includegraphics[width=1.0\linewidth]{15.png}

\end{figure}

\end{frame}

\begin{frame}

\frametitle{Стратегия игры}

\jj

Так как в свой ход компьютер записывает слова, которые ранее еще не использовались, то в начале лучше составлять слова как можно длиннее, а затем переходить к более коротким. Таким образом, можно набрать баллов больше, чем у компьютера.

\end{frame}

\end{document}

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Балтийский федеральный университет имени Иммануила Канта»

ОНК «Институт высоких технологий»

ДНЕВНИК

учебной практики по получению первичных

профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности

г. Калининград 2022 г.

**1.Информационная часть**

Тронина София Александровна, студентка очной формы обучения 3 курса группы 05\_КБ\_19\_О\_/ специальности 10.05.01 Компьютерная безопасность, специализация «Математические методы защиты информации» в соответствии с приказом № 1527ст от 06 мая 2022 г. направляется на учебную практику по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности в Высшую школу компьютерных наук и прикладной математики.

Период практики – с 20.06.2022 г. по 02.07.2022 г.

Руководитель практики от университета – доцент ОНК «ИВТ» Киршанова Елена Алексеевна.

ОНК «Институт высоких технологий»

Контактный номер телефона +7 (4012) 338 217

Первый заместитель

директора ОНК «ИВТ» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Шпилевой А.А.

**2. Программа практики**

**2.1. План работы**

|  |  |  |
| --- | --- | --- |
| **№ п.п.** | **Рабочее место практиканта, вид работы** | **Продолжительность**  **(в днях)** |
| **1.** | Компьютерный класс, ауд. 324 | **13 дней** |
| **2.** | Защита работ. Ауд. 324 | **1 день** |

**2.2. Индивидуальное задание по профилю подготовки/специальности**

1. Пройти инструктаж по технике безопасности.
2. Ознакомиться и выполнить задачи на практику.
3. Написать отчет по практике.

Руководитель практики от университета

доцент ОНК «ИВТ» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Киршанова Елена Алексеевна

**3. Ход выполнения практики**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п.п.** | **Дата** | **Описание выполненной работы** | **Отметки**  **руководителя практики от профильной организации** |
| 1 | 20.06.2022 | Пройден инструктаж по технике безопасности.  Получение индивидуального задания на практику |  |
| 2 | 21.06.2022-  23.06.2022 | Выбор игры, разработка возможного плана решения |  |
| 3 | 24.06.2022-28.06.2022 | Написание программы на языке программирования Python |  |
| 4 | 29.06.2022-30.06.2022 | Создана презентация на основе системы верстки LaTeX |  |
| 5 | 01.07.2022 | Подготовка отчета по учебной практике |  |
| 6 | 02.07.2022 | Защита отчета по учебной практике |  |

**4. Отзыв руководителя практики**

Тронина София Александровна, студентка очной формы обучения 3 курса группы 05\_КБ\_19\_О\_/ специальности 10.05.01 Компьютерная безопасность, специализация «Математические методы защиты информации» в соответствии с приказом № 1527ст от 06 мая 2022 г. направляется на учебную практику по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности в Высшую школу компьютерных наук и прикладной математики.

Период практики – с 20.06.2022 г. по 02.07.2022 г.

Программа практики и индивидуальное задание на практику выполнены. Отчёт по практике сдан и защищён на отчётной конференции.

Студентка Тронина София Александровна в процессе прохождения практики справилась с поставленными задачами, приобрела первичные профессиональные навыки и компетенции, в том числе:

* способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;
* способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения.

Учебная практика оценена на оценку \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики от университета –

доцент ОНК «ИВТ» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Киршанова Елена Алексеевна

«02» июля 2022 г.