Tallinna Tööstushariduskeskus **Logistika IT süsteemide spetsialist**

**Projekti „Wordle“ Pythoni keeles loomine**Praktiline töö

Juhendaja: Marina Oleinik  
Koostaja: Karina Linter  
Rühm: LOGITpv23

Tallinn 2024

**Содержание/Sisukord**

[Введение 4](#_Toc166489824)

[Цель работы: 4](#_Toc166489825)

[Töö eesmärk: 4](#_Toc166489826)

[Основные модули, используемые в проекте: 5](#_Toc166489827)

[Основные конструкции, используемые в коде: 5](#_Toc166489828)

[Teooria: 6](#_Toc166489829)

[Peamised projektis kasutatud moodulid: 6](#_Toc166489830)

[Koodis kasutatavad põhikonstruktsioonid: 6](#_Toc166489831)

[Практическая часть/Praktiline osa 7](#_Toc166489832)

[Заключение 13](#_Toc166489833)

[Kokkuvõte 13](#_Toc166489834)

[Источники/Allikad 14](#_Toc166489835)

[Библиография/Bibliograafia 14](#_Toc166489836)

**Картинки/Pildid**

[Pilt 1 "Mängu välimus" 9](#_Toc166484728)

[Pilt 2 "Dialoogiaken" 9](#_Toc166484729)

[Pilt 3 "Teine dialoogiaken" 9](#_Toc166484730)

# Введение

Цель работы: Цель этого проекта - создать игру "Wordle".

Это игра, где нужно угадать одно существительное на эстонском языке из пяти букв. На это даётся шесть попыток. Для начала нужно вписать любое слово в нижнюю строчку игрового поля.

Если хотя бы одна буква в слове соответствует загаданному, то она загориться зелёным, если стоит в нужном месте, и жёлтым, когда стоит не там, где надо. Если Вы не угадали букву, она остаётся гореть черным. Затем, зная угаданные буквы, можно вписать новое слово, и так пока не угадаете или не закончатся попытки.

Создана игра с использованием модуля **Tkinter** в Python. Игра предназначена для развлечения и развития логического мышления. Она также предоставляет пользователю возможность добавлять и удалять слова из списка для игры.

Töö eesmärk: Selle projekti eesmärk on luua mäng nimega "Wordle".

See on mäng, kus tuleb ära arvata üks viietäheline sõna eesti keeles. Selleks antakse kuus katset. Kõigepealt tuleb kirjutada mängulauda alumisele reale ükskõik milline sõna.

Kui vähemalt üks täht sõnas vastab ära arvatud tähega, süttib see roheliselt, kui see on õiges kohas, ja kollaselt, kui see on vales kohas. Kui sa ei oska tähte ära arvata, jääb see mustaks põlema. Siis, teades äraarvatud tähti, saate kirjutada uue sõna ja nii edasi, kuni arvate ära või katsed otsa.

Mäng on loodud Pythoni **Tkinter**-mooduliga. Mäng on mõeldud meelelahutuslikuks ja loogilist mõtlemist arendavaks. Samuti annab see kasutajale võimaluse lisada ja eemaldada sõnu mängu nimekirja.Теория:

Игра разработана на языке программирования Python. Игра на эстонском языке, которая проверяет Ваши знания и грамматику.

Задумка по реализации языка появилась в конце 1980-х годов, а разработка его реализации началась в 1989 году сотрудником голландского института CWI Гвидо ван Россумом. Для распределённой операционной системы Amoeba требовался расширяемый скриптовый язык, и Гвидо начал разрабатывать Python на досуге, позаимствовав некоторые наработки для языка ABC (Гвидо участвовал в разработке этого языка, ориентированного на обучение программированию). В феврале 1991 года Гвидо опубликовал исходный текст в группе новостей alt.sources. С самого начала Python проектировался как объектно-ориентированный язык.

## Основные модули, используемые в проекте:

* **tkinter**: для создания графического интерфейса
* **random**: для выбора случайного слова из списка
* **simpledialog** из **tkinter**: для взаимодействия с пользователем при добавлении и удалении слов из списка

## Основные конструкции, используемые в коде:

* **Функции:** используются для организации кода и выполнения определенных задач, таких как загрузка слов, проверка введенного слова, обновление интерфейса и т. д.
* **Циклы:** используются для выполнения повторяющихся операций, например, перебора символов в слове или ячеек в интерфейсе.
* **Условные операторы:** используются для принятия решений на основе определенных условий, например, проверка правильности введенного слова.

Teooria: Mäng on välja töötatud Python programmeerimiskeeles. Mäng on eesti keeles, mis paneb proovile teie teadmised ja grammatika.

Keele rakendamise idee tekkis 1980ndate lõpus ja selle rakendamise arendamine

alustas 1989. aastal Guido van Rossum Hollandi CWIst.

Rossum. Amoeba hajutatud operatsioonisüsteem vajas laiendatavat

skriptimiskeelt ja Guido hakkas Pythoni keelt vabal ajal arendama, laenates

mõningaid töid ABC-ga (Guido osales selle keele arendamisel,

orienteeritud programmeerimise õpetamisele). 1991. aasta veebruaris alustas Guido

avaldas lähtekoodi uudisgrupis alt.sources. Algusest peale.

Python oli kavandatud objektorienteeritud keelena.

## Peamised projektis kasutatud moodulid:

* **tkinter:** graafilise kasutajaliidese loomiseks
* **random:** juhusliku sõna valimiseks nimekirjast
* **simpledialog** **tkinterist**: kasutajaga suhtlemiseks sõnade lisamisel ja eemaldamisel nimekirjast.

## Koodis kasutatavad põhikonstruktsioonid:

* **Funktsioonid:** kasutatakse koodi organiseerimiseks ja konkreetsete ülesannete täitmiseks, näiteks sõnade laadimiseks, sisestatud sõna kontrollimiseks, kasutajaliidese uuendamiseks jne.
* **Silmused:** kasutatakse korduvate toimingute sooritamiseks, näiteks sõna tähemärkide või liidese lahtrite üle loopimiseks.
* **Tingimuslaused:** kasutatakse teatud tingimustel põhinevate otsuste tegemiseks, näiteks sisestatud sõna õigsuse kontrollimiseks.

# Практическая часть/Praktiline osa

# import tkinter as tk

import random

from tkinter import simpledialog

**#Загрузка списка слов из файла:** Cоздаётся функция **laadi\_sõnad**, которая читает файл со словами и возвращает список слов, удовлетворяющих определенным критериям. **/ Sõnade loendi laadimine failist:** Loome funktsiooni laadi\_sõnad, mis loeb sõnafaili ja tagastab teatud kriteeriumidele vastavate sõnade loendi.

def laadi\_sõnad(sõnad):

with open(sõnad, 'r', encoding='utf-8') as fail:

sõnad = [rivi.strip().upper() for rivi in fail if len(rivi.strip()) == 5]

return sõnad

**#Проверка введенного слова:** Функция **kontrolli\_sõna** проверяет введенное слово на соответствие текущему загаданному слову и возвращает список цветов для каждой буквы слова (зеленый - верная буква на верном месте, желтый - верная буква, но не на верном месте, черный - неверная буква). **/ Sisestatud sõna kontrollimine:** Funktsioon **kontrolli\_sõna** kontrollib sisestatud sõna praeguse mõistatatud sõna suhtes ja tagastab iga sõna tähe kohta värvilise nimekirja (roheline - õige täht õiges kohas, kollane - õige täht, kuid mitte õiges kohas, must - vale täht).

def kontrolli\_sõna(arvamine, sõna):

tulemus = []

for i, täht in enumerate(arvamine):

if täht == sõna[i]:

tulemus.append('green')

elif täht in sõna:

tulemus.append('yellow')

else:

tulemus.append('black')

return tulemus

**#Обновление интерфейса:** Функция **uuenda\_liidest** обновляет интерфейс, отображая введенное пользователем слово и соответствующие ему цвета. **/** **Kasutajaliidese uuendamine:** funktsioon **uuenda\_liidest** uuendab kasutajaliidese, kuvades kasutaja sisestatud sõna ja sellele vastavad värvid.

def uuenda\_liidest(arvamine, tulemus):

for i, täht in enumerate(arvamine):

sisestus = sisestused[hetke\_üritus][i]

sisestus.delete(0, 'end')

sisestus.insert('end', täht)

sisestus.config(disabledforeground=tulemus[i], state='disabled')

**#Обработка действий пользователя:** Функция **saatmine** вызывается при нажатии кнопки "Отправить" и осуществляет проверку введенного слова, обновляя интерфейс в соответствии с результатом. Если у пользователя закончились попытки или он угадал слово, игра завершается. / **Kasutaja tegevuste käsitlemine:** funktsiooni saatmine kutsutakse üles, kui vajutatakse nuppu submit, ja see kontrollib sisestatud sõna, uuendades vastavalt tulemusele kasutajaliidest. Kui kasutajal saavad katsed otsa või kui ta arvab sõna ära, siis mäng lõpeb.

def saatmine():

global hetke\_üritus

arvamine = arvamine\_muutuja.get().upper()

if len(arvamine) != 5:

tulemus\_silt.config(text="Sõna peab olema 5 tähemärki pikk!")

return

tulemus=kontrolli\_sõna(arvamine, hetke\_sõna)

uuenda\_liidest(arvamine, tulemus)

if arvamine==hetke\_sõna or hetke\_üritus==5:

tulemus\_silt.config(text=f"Sõna: {hetke\_sõna}")

for sisestusrida in sisestused:

for sisestus in sisestusrida:

sisestus.config(state='disabled')

else:

hetke\_üritus += 1

tulemus\_silt.config(text="")

arvamine\_muutuja.set("")

def algus\_uuesti():

global hetke\_sõna, hetke\_üritus

hetke\_sõna=random.choice(sõnad)

hetke\_üritus=0

arvamine\_muutuja.set("")

tulemus\_silt.config(text="")

for sisestusrida in sisestused:

for sisestus in sisestusrida:

sisestus.config(state='normal')

sisestus.delete(0, 'end')

**#Добавление и удаление слов:** Функции **lisa\_sõna** и **eemalda\_sõna** вызываются при нажатии соответствующих кнопок и позволяют пользователю добавлять и удалять слова из списка для игры. **/** **Sõnade lisamine ja eemaldamine:** funktsioonid lisa\_sõna ja eemalda\_sõna kutsutakse välja vastavate nuppude vajutamisel ja võimaldavad kasutajal lisada ja eemaldada sõnu mängu nimekirjast.

def lisa\_sõna():

uus\_sõna=simpledialog.askstring("Lisa sõna", "Sisestage sõna, mida soovite lisada:")

if uus\_sõna:

sõnad.append(uus\_sõna.upper())

def eemalda\_sõna():

sõna\_kustutada=simpledialog.askstring("Eemalda sõna", "Sisestage sõna, mida soovite eemaldada:")

if sõna\_kustutada:

if sõna\_kustutada.upper() in sõnad:

sõnad.remove(sõna\_kustutada.upper())

else:

tulemus\_silt.config(text="Seda sõna pole loendis!")

#Initsialiseerimine

sõnad=laadi\_sõnad('Sõnad.txt')

hetke\_sõna=random.choice(sõnad)

hetke\_üritus=0

aken = tk.Tk()

aken.title("Sõnapilt")

aken.geometry("900x900")

aken.config(bg='lightblue')

#Создание рамки для выравнивания содержимого по левому краю

põhiraam = tk.Frame(aken, bg='lightblue')

põhiraam.pack(expand=True)

# Создание полей ввода для букв слова

sisestused = [[tk.Entry(põhiraam, font=('Arial', 36), width=2, bg='lightblue', fg='black') for \_ in range(5)] for \_ in range(6)]

for i, rida in enumerate(sisestused):

for j, sisestus in enumerate(rida):

sisestus.grid(row=i, column=j, padx=5, pady=5, sticky='w') #Изменено расположение ячеек на левый край

#Поле ввода для предполагаемого слова

arvamine\_muutuja=tk.StringVar()

arvamine\_sisestus=tk.Entry(põhiraam, textvariable=arvamine\_muutuja, font=('Arial', 36), width=8, bg='lightblue', fg='black')

arvamine\_sisestus.grid(row=7, column=0, columnspan=5, pady=15, sticky='w') #Изменено расположение поля ввода на левый край

saatmis\_nupp=tk.Button(põhiraam, text="Saada", command=saatmine, bg='lightblue', fg='black', width=20, height=3)

saatmis\_nupp.grid(row=8, column=0, columnspan=5, pady=15, sticky='w')

tulemus\_silt=tk.Label(põhiraam, text="", font=('Arial', 36), bg='lightblue', fg='white')

tulemus\_silt.grid(row=8, column=0, columnspan=5, pady=15, sticky='w')

algus\_uuesti\_nupp=tk.Button(põhiraam, text="Alusta uuesti", command=algus\_uuesti, bg='lightblue', fg='black', width=20, height=3)

algus\_uuesti\_nupp.grid(row=10, column=0, columnspan=5, pady=15, sticky='w')

lisa\_sõna\_nupp=tk.Button(põhiraam, text="Lisa sõna", command=lisa\_sõna, bg='lightblue', fg='black', width=20, height=3)

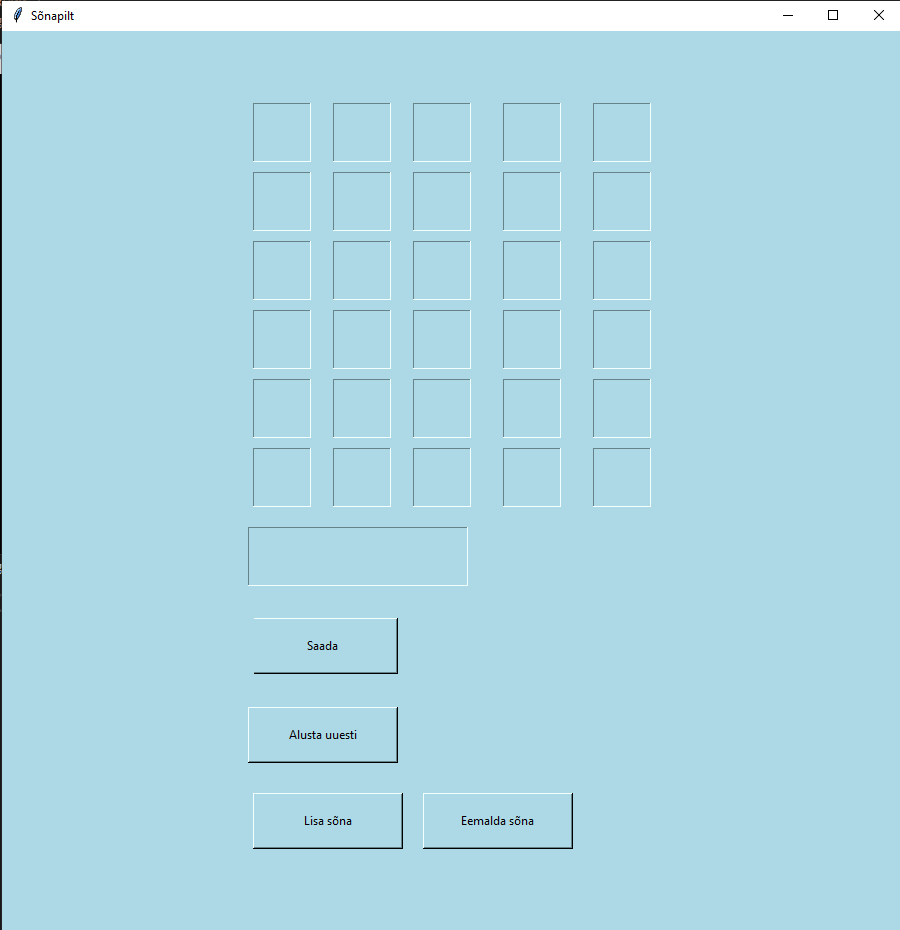
lisa\_sõna\_nupp.grid(row=11, column=0, columnspan=2, pady=15, padx=5, sticky='w')

eemalda\_sõna\_nupp=tk.Button(põhiraam, text="Eemalda sõna", command=eemalda\_sõna, bg='lightblue', fg='black', width=20, height=3)

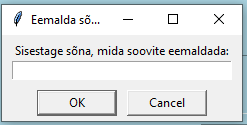
eemalda\_sõna\_nupp.grid(row=11, column=2, columnspan=2, pady=15, padx=15, sticky='w')

aken.mainloop()

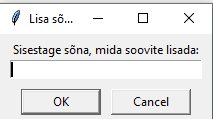
После выполнения всех шагов код запускается, создавая графический интерфейс игры, где пользователь может вводить слова и взаимодействовать с интерфейсом для продолжения игры или изменения списка слов. / Kui kõik sammud on lõpetatud, käivitatakse kood, mis loob graafilise mänguliidese, kus kasutaja saab sisestada sõnu ja interakteeruda liidese abil mängu jätkamiseks või sõnade nimekirja muutmiseks.



Pilt "Mängu välimus"



Pilt "Dialoogiaken"



Pilt "Teine dialoogiaken"

# Заключение

Я сделала игру, где нужно угадать одно существительное на эстонском языке из пяти букв. Я довольна своим результатом, игра получилась достаточно хорошей, но не получилось написать функцию, которая позволяет вписывать каждую букву отдельно сразу в ячейку, поэтому я заменила ее на функцию, которая создает специальное поле, куда игрок вписывает полностью слово. Также в будущем я хочу усовершенствовать игру – добавить больше слов.

# Kokkuvõte

Ma tegin mängu, kus tuleb ära arvata üks viietäheline nimisõna eesti keeles. Olen oma tulemusega rahul, mäng tuli päris hea välja, aga ma ei saanud kirjutada funktsiooni, mis võimaldab iga tähe eraldi kasti kirjutada, seega asendasin selle funktsiooniga, mis loob spetsiaalse välja, kuhu mängija kirjutab kogu sõna. Ka tulevikus tahan ma mängu täiustada - lisada rohkem sõnu.

# Источники/Allikad

Информация об истории создания языка Python/ Teave Pythoni keele ajaloo kohta - <https://ru.wikipedia.org/wiki/Python>

# Библиография/Bibliograafia

Rossum, G. v.