|  |  |
| --- | --- |
| *voenmeh* | МИНОБРНАУКИ РОССИИ  федеральное государственное бюджетное образовательное учреждение  высшего образования  **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»**  **(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Факультет |  | И |  | Информационные и управляющие системы |
|  |  | шифр |  | наименование |
| Кафедра |  | И5 |  | Информационные системы и программная инженерия |
|  |  | шифр |  | наименование |
| Дисциплина |  | Технологии программирования | | |

**Практическая работа №1**

Использование API сторонних приложений

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент группы | | | | |  | И975 |
| Кабин А. К. | | | | | | |
| Фамилия И.О. | | | | | | |
| **ПРЕПОДАВАТЕЛЬ** | | | | | | |
| Гаврютина А.А. | |  |  | | | |
| Фамилия И.О. Подпись | | | | | | |
| Оценка | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |  | | |
| «\_\_\_\_\_» | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | 2020 г | | |

САНКТ-ПЕТЕРБУРГ

2020г.

Задача: познакомиться с программным интерфейсом стороннего приложения (англ. application programming interface, API), создать свое приложение с использованием приложения, указанного в вашем варианте.

**Вариант 12:** Steam API. Создать форму с combobox и 2 кнопками. В combobox – название игры. По 1-й кнопке происходит поиск новостей по введенной игре, по 2-й – получение глобальной статистики по игре (кол-во игроков и т.д.).

Начало работы приложения представлено на рисунке 1. В combobox на выбор дается 3 игры DOTA2, PUBG, Path of Exile.

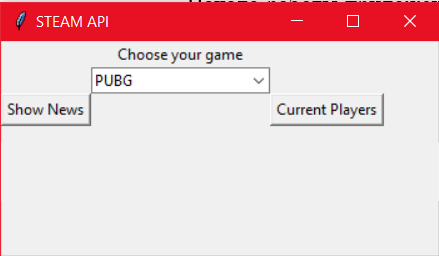


Рисунок 1 – начало работы

Новости по игре PUBG представлены на рисунке 2.

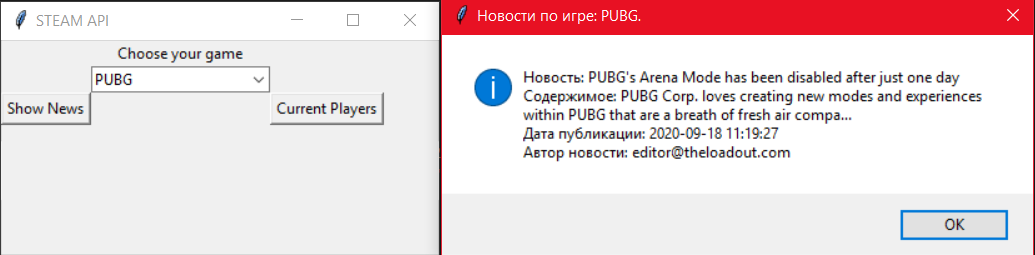


Рисунок 2 – новости по игре PUBG

При повторном нажатии на кнопку отображения новостей или кнопку отображения текущих игроков отображаемые значения обновятся. Результат представлен на рисунке 3.

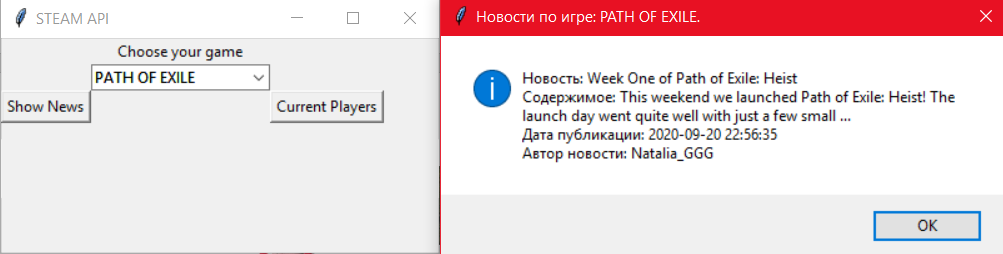


Рисунок 3 – повторное нажатие на Show News для отображения новостей

**Текст программы:**

main.py

import tkinter as tk

from tkinter import ttk

from tkinter import Button

from tkinter import messagebox

from datetime import datetime

from steamwebapi.api import ISteamUserStats, ISteamUser, \_SteamWebAPI

STEAM\_API\_KEY = '18BA4226434B86ED2AD10490BC510B56'

DOTA2\_ID = 570

PUBG\_ID = 578080

POE\_ID = 238960

STEAM\_WEB\_API = \_SteamWebAPI(STEAM\_API\_KEY)

STEAM\_USER\_STATS = ISteamUserStats(steam\_api\_key=STEAM\_WEB\_API.apikey)

GAME\_ID\_DICT = {

   "DOTA2": DOTA2\_ID,

   "PUBG": PUBG\_ID,

   "PATH OF EXILE": POE\_ID

}

def create\_news\_url(appID, count = 1, maxlength = 100, format = "json"):

   url\_params\_dict = {

      "appid": appID,

      "count": count,

      "maxlength": maxlength,

      "format": format

   }

   return STEAM\_WEB\_API.create\_request\_url("ISteamNews", "GetNewsForApp", 2, parameters = url\_params\_dict)

def get\_news\_for\_app(appID, count = 1, maxlength = 100, format = "json"):

   url = create\_news\_url(appID, count, maxlength, format)

   data = STEAM\_WEB\_API.retrieve\_request(url)

   return STEAM\_WEB\_API.return\_data(data, format=format)

def update\_news(app\_name):

   messagebox.showinfo(title = "Новости по игре: %s." % (app\_name),

                       message = get\_parsed\_news(get\_news\_for\_app

                       (appID = GAME\_ID\_DICT[app\_name], format = "json")))

def get\_parsed\_news(raw\_news):

   preparsed\_news = raw\_news["appnews"]["newsitems"][0]

   title = str(preparsed\_news["title"])

   author = str(preparsed\_news["author"])

   contents = str(preparsed\_news["contents"])

   date = datetime.utcfromtimestamp(int(preparsed\_news["date"])).strftime('%Y-%m-%d %H:%M:%S')

   return "Новость: " + title + \

          "\nСодержимое: " + contents + \

          "\nДата публикации: " + date + \

          "\nАвтор новости: " + author

def update\_stat(app\_name):

   messagebox.showinfo(title = "Количество игроков в игре: %s." % (app\_name),

                       message =  get\_parsed\_stat(STEAM\_USER\_STATS.get\_number\_of\_current\_players

                       (appID = GAME\_ID\_DICT[app\_name], format = "json")))

def get\_parsed\_stat(raw\_stat):

   return "Количество текущих игроков: " + str(raw\_stat["response"]["player\_count"])

def create\_combobox():

   app = tk.Tk()

   app.geometry('350x100')

   app.title('STEAM API')

   labelTop = tk.Label(app, text="Choose your game")

   labelTop.grid(column=1, row=0)

   game\_combobox = ttk.Combobox(app,

                                values=["DOTA2",

                                        "PUBG",

                                        "PATH OF EXILE"], state="readonly")

   game\_combobox.grid(column=1, row=1)

   game\_combobox.current(1)

   news\_btn = Button(app,

                     text="Show News",

                     command= lambda: update\_news(game\_combobox.get()))

   news\_btn.grid(column=0, row=2)

   glob\_stat\_btn = Button(app,

                          text="Current Players",

                          command= lambda: update\_stat(game\_combobox.get()))

   glob\_stat\_btn.grid(column=3, row=2)

   app.mainloop()

def main():

   create\_combobox()

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Вывод:** в результате выполнения данной практической работы было создано приложение с использованием Steam Api для ознакомления с программным интерфейсом стороннего приложения. Как пример работоспособности были созданы две кнопки, осуществляющие отображение новостей и количества игроков для выбранной игры.