Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«СибирскИЙ государственнЫЙ Университет

геоСИСТЕМ И ТЕХНОЛОГИЙ»

(СГУГИТ)

ОТЧЕТ

ЛАБОРАТОРНАЯ РАБОТА 7. РАЗРАБОТКА ГРАФИЧЕСКОГО ИНТЕРФЕЙСА

Выполнил обучающийся

Группы БИ-33

Ахиярова И.М.

Проверил

Воронкин Е.Ю.

Новосибирск – 2021

**Цель работы:** формирование базовых понятий построения графического интерфейса пользователя (GUI), углубление знаний обучающихся по программированию.

**Задачи работы**

1. Ознакомиться этапами разработки графического интерфейса;

2. Научиться создавать графический интерфейс для собственных приложений.

**Перечень обеспечивающих средств**

Для выполнения работы необходимо иметь компьютер с установленной операционной системой семейства Windows, установленным python и IDE PyCharm Professional.

**Задание**

Разработать графический интерфейс приложения согласно своему варианту, выданным преподавателем.

**Технология выполнения работы**

* Я занимаюсь разработкой программы для генерации случайных статьей из википедии. Для создания интерфейса были использованы средства библиотеки PyQT5. PyQt5 является одним из наиболее часто используемых модулей для создания GUI приложений в Python.
* Для создания программы «Генерация статей» требуются только:
* Buttons (для генерации заголовка и закрытия программы);
* QMessageBox (для предоставления выбора пользователю);
* QTextEdit (как контейнер для хранения контента статей и для хранения БД).

До генерации статей, программа собой представляет одну кнопку с надписью: «Случайная статья». После нажатия кнопки и выбора положительного ответа о заголовке статьи, мы можем наблюдать полноценный интерфейс нашего программного продукта. Интерфейс представляет собой два окна и одну кнопку для закрытия программы. В верхнем окне выводится выбранная нами статьи, благодаря чему мы можем с ней ознакомиться. Нижнее окно служит, для хранения истории генерации случайных статей. Кнопка «Закрыть» служит для закрытия программы.

**Кнопка**

Для реализации программы «Генерация статей» необходимо создание класса основной формы. Который включает в себя кнопку для генерации случайного заголовка статьи по средствам api Wikipedia.

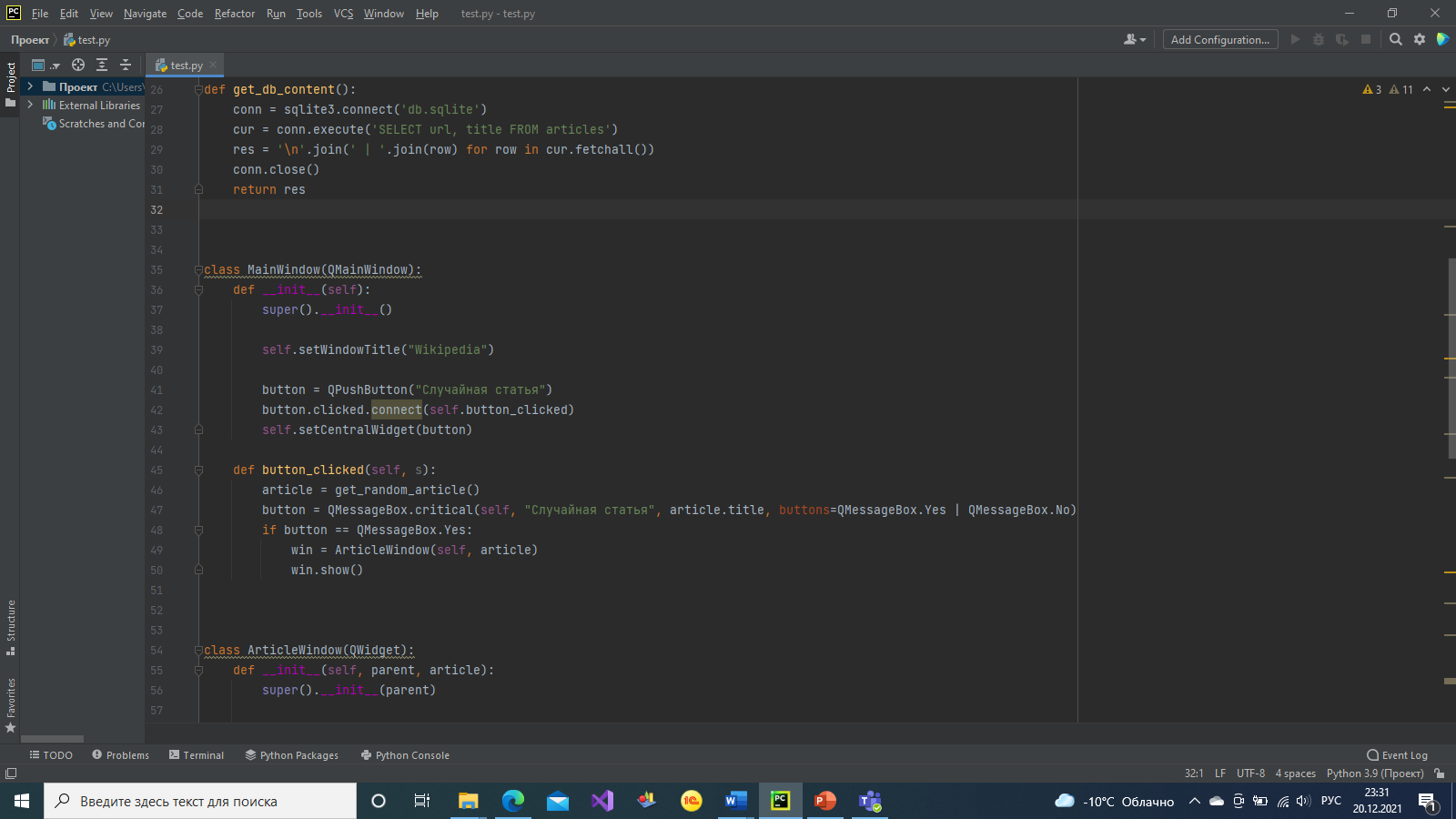


Рисунок 1 – Создание кнопки «Случайная статья» и кнопок выбора действия

В данной части кода прописаны действия для кнопки «Случайная статья». Принцип ее работы заключается в том, что при нажатии на кнопку, программа генерирует случайный заголовок и предлагает его пользователю. Далее пользователю предоставляется выбор: если он хочет прочитать сгенерированную статью, то он жмет кнопку «Yes», а если не хочет, то жмет кнопку «No».

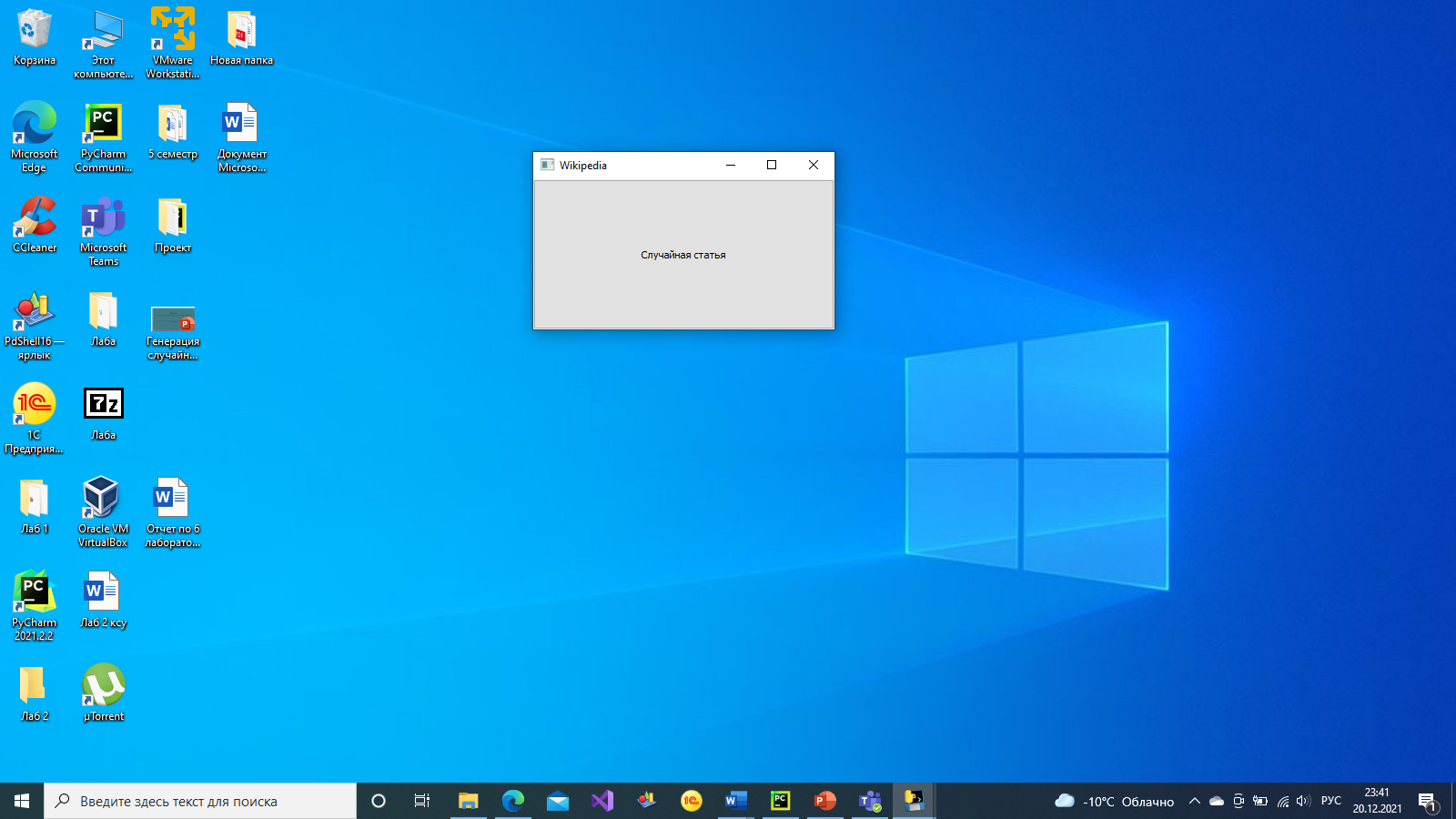


Рисунок 2 – Кнопка «Случайная статья»

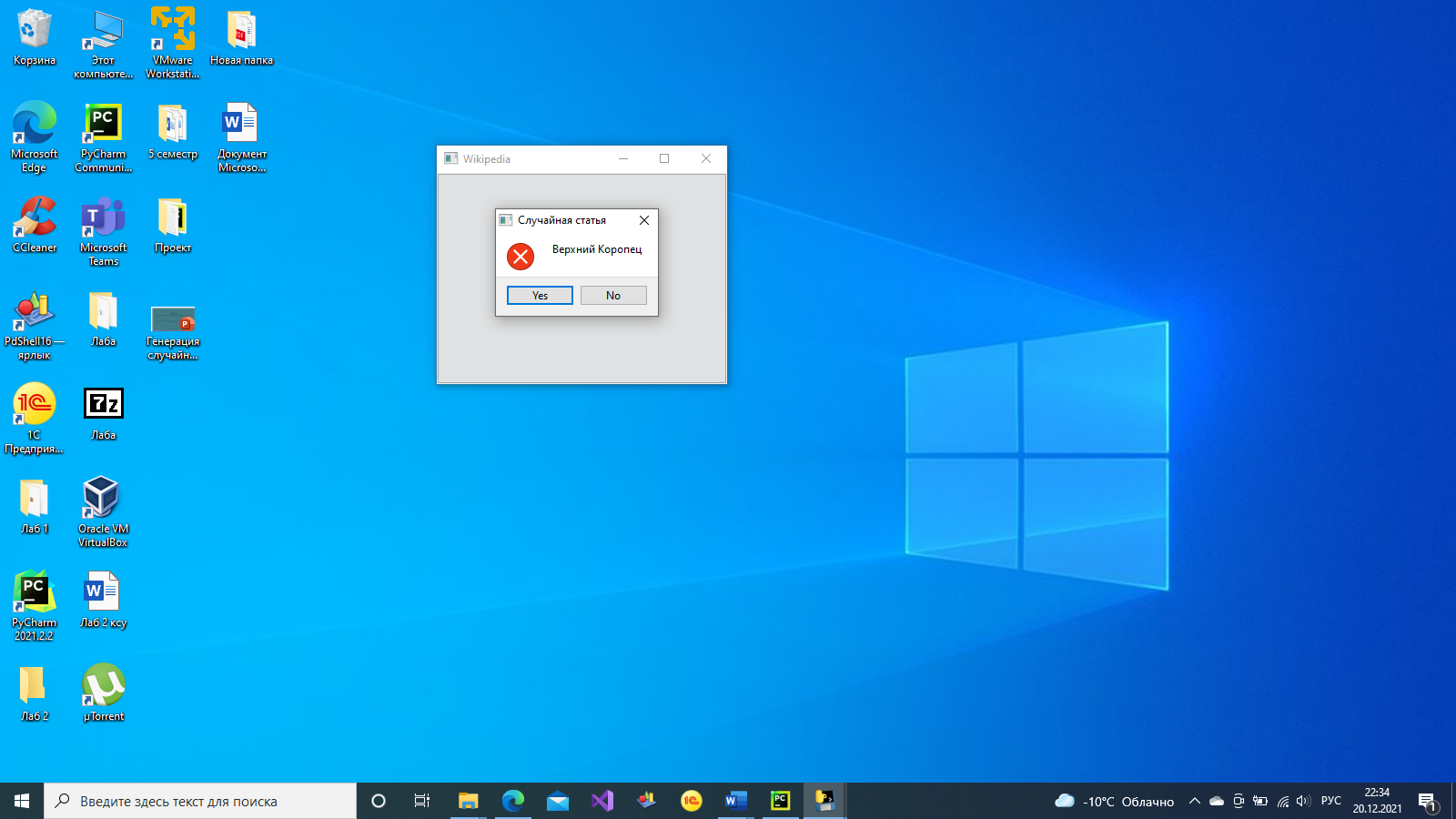


Рисунок 3 – Кнопки выбора действия

Так же в нашей программе имеется кнопка «Закрыть».

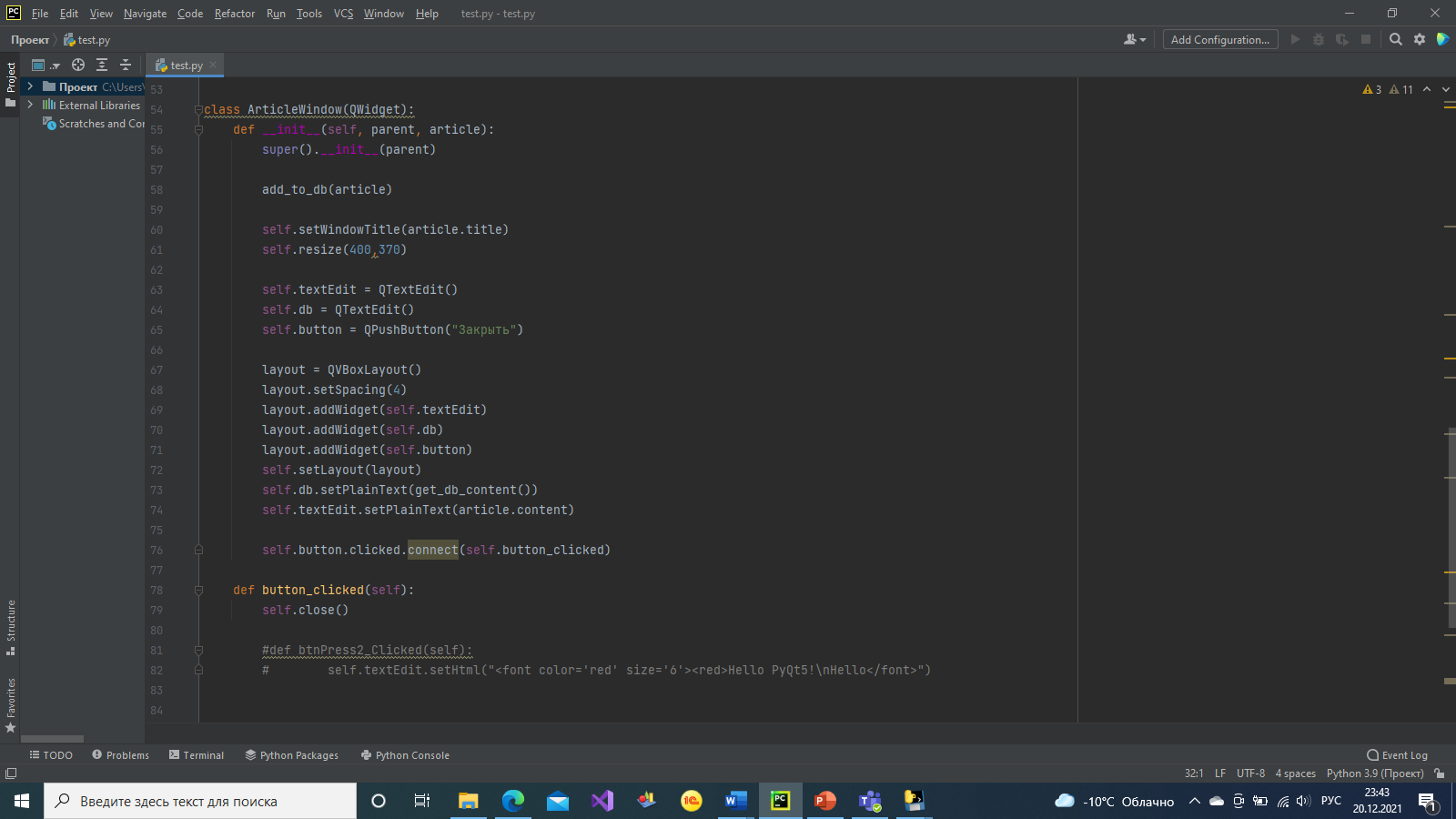


Рисунок 4 – Часть программного кода для кнопки «Закрыть»

**Контейнер для хранения контента статей и для хранения БД**

В данной программе имеется 2 контейнера. В первый выводится случайная статья из википедии, а во второй контейнер выводится база данных, в которой находится история просмотренных пользователем статей.

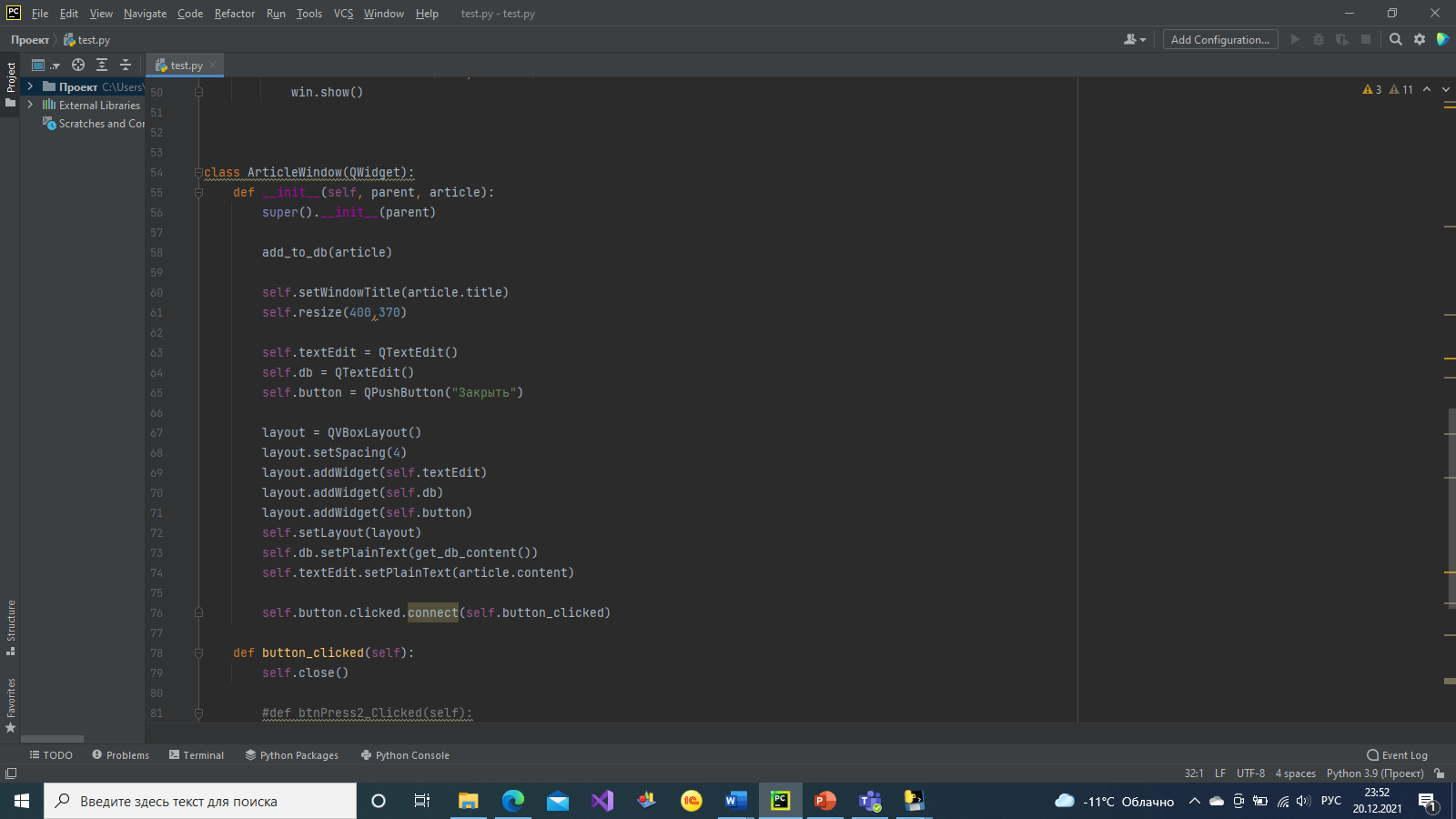


Рисунок 5 – Программный код для контейнеров

В конечном итоге наша программа имеет вид:

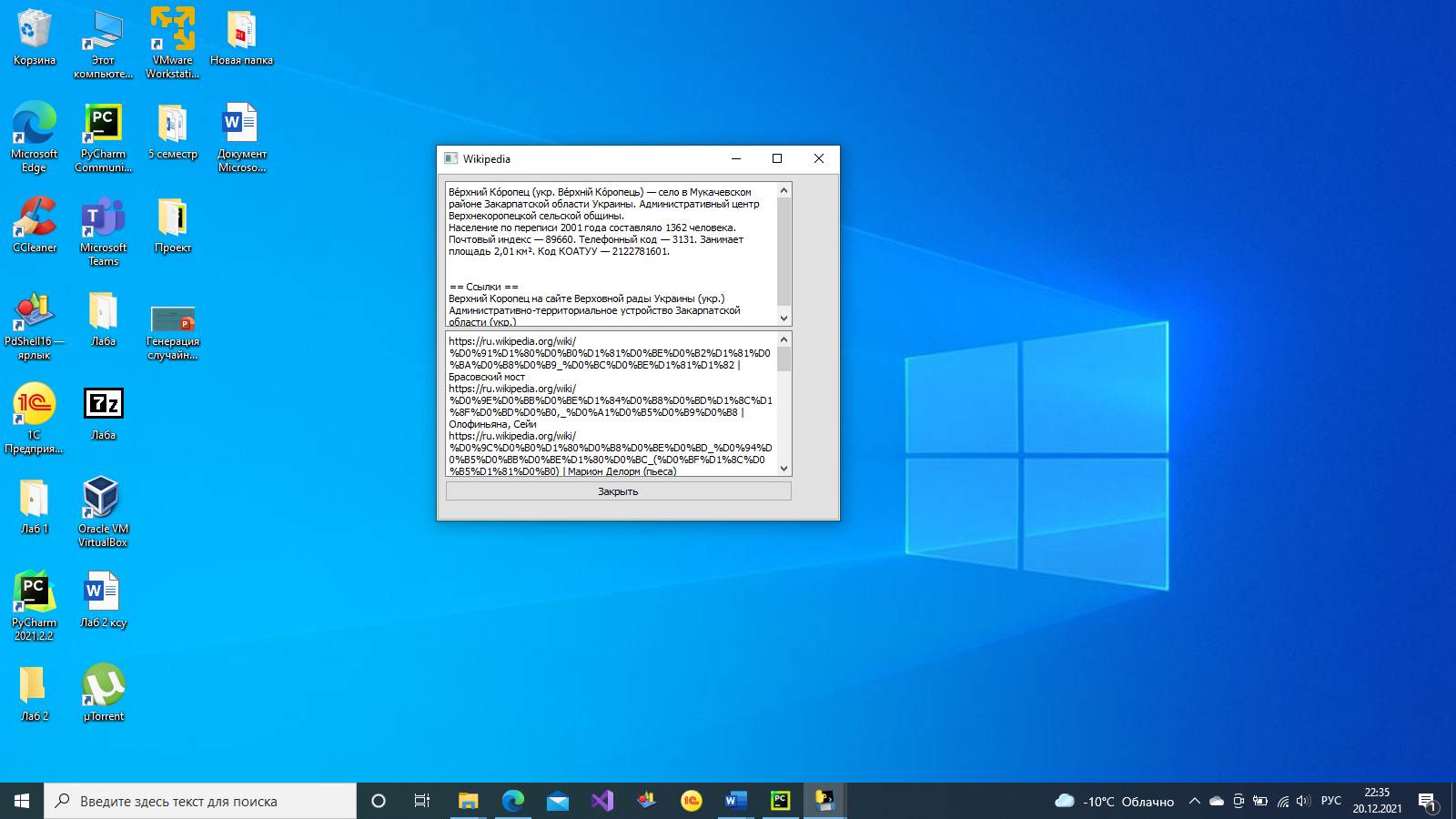


Рисунок 6 – Интерфейс программы

**Вывод:** я разработала графический интерфейс приложения согласно своему варианту, выданным преподавателем.

**Контрольные вопросы**

1. Что такое событийно-ориентированное программирование?

Объектно-ориентированное программирование (ООП) – методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования.

1. Дайте определение понятий «событие», «обработчик событий» и «цикл обработки событий».

Событие – некоторое действие пользователя.

Обработчик событий – функция, которая выполняется, когда происходит событие.

Цикл обработки событий – цикл, который проверяет все произошедшие события и в котором вызываются все соответствующие обработчики.

1. Перечислите основные инструменты для создания графического интерфейса.

В python есть три основные библиотеки для создания графических интерфейсов: tkinter, wxPython и PyQT.

1. Опишите алгоритм построения интерфейса на базе главного окна и способы размещения виджетов.

Алгоритм построения интерфейса:

1. Импорт библиотеки

2. Создание главного окна

3. Создание виджетов

4. Установка свойств виджетов (размеров, положения, вида и др.)

5. Определение событий

6. Определение обработчиков событий

7. Расположение виджет на главном окне

8. Отображение главного окна

1. Каковы синтаксис создания главного окна и его основные атрибуты?

Главное окно является объектом класса Tk, обычно ему дают имя root.

root = Tk()

У главного окна есть атрибуты title – размер окна. Обычно объект создается без аттрибутов.

1. Опишите особенности построения текстовых виджетов.

Надписи – это виджеты, содержащие строку (или несколько строк) текста и служащие в основном для информации пользователя. Могут иметь различные шрифты.

Однострочное поле ввода создается с помощью класса **Entry**. Такое поле позволяет ввести только одну строчку.

Класс **Text** предназначен для создания многострочного текстового поля. У этого класса есть свойство **wrap**. Свойство **wrap** в зависимости от своего значения позволяет переносить текст, вводимый пользователем либо по символам, либо по словам, либо вообще не переносить, пока пользователь не нажмет Enter.

1. Каковы особенности создания управляющих кнопок?

Объект-кнопка создается вызовом класса **Button** модуля tkinter. Обязательным аргументом является родительский виджет. Другие свойства могут указываться при создании кнопки или задаваться позже. Синтаксис кнопки:

переменная = Button (родительский виджет, свойство = значение…)

У кнопки есть свойства: text, width, height, bg (background), и другие.

1. Как создать меню?

В tkinter экземпляр меню создается от класса **Menu**, далее его надо привязать к виджету, на котором оно будет расположено. Обычно таковым выступает главное окно приложения. Его опции **menu** присваивается экземпляр **Menu** через имя связанной с экземпляром переменной.

1. Как реализовать связывание событий, инициированных пользователем, с обработчиком этих событий? Опишите синтаксис соответствующего метода.

Связывание событий реализуется при помощи метода bind. При вызове метода bind событие передается в качестве первого аргумента.

Название события заключается в кавычки, а также в угловые скобки ‘< >’. События описывается с помощью зарезервированных ключевых слов.

Обработчик события указывается вторым аргументом.

root.bind(‘<событие>’, обработчик)

1. Опишите особенности построения виджета canvas и работу с основными графическими примитивами.

В tkinter от класса **Canvas** создаются объекты-холсты, на которых можно "рисовать", размещая различные фигуры и объекты. Делается это с помощью вызовов соответствующих методов.

При создании экземпляра **Canvas** необходимо указать его ширину и высоту. При размещении геометрических примитивов и других объектов указываются их координаты на холсте.

Создание прямоугольников – метод **create\_rectangle.** Указываются координаты верхнего левого и нижнего правого углов. Можно указывать цвет, рамку и др.

Создание полигонов – метод **create\_polygon**. Рисуется произвольный многоугольник путем задания координат каждой его точки.

Создание эллипсов – метод **create\_oval**. При этом задаются координаты гипотетического прямоугольника, описывающего эллипс.