

Проектная работа

«Программа для генерации вариантов тестового тура заключительного этапа  
всероссийской олимпиады школьников по биологии»

по Информатике

Выполнил ученик 10 «Б» класса Петров А.А.

Челябинск, 2025 год

## ОГЛАВЛЕНИЕ

<b>Глава 1. Теория.....</b>	<b>2</b>
Актуальность.....	2
Проблема.....	2
Цель.....	3
Задачи.....	3
<b>Глава 2. Практика.....</b>	<b>4</b>
Написание программы.....	4
Создание полной базы заданий.....	5
<b>Заключение.....</b>	<b>6</b>

# Глава 1. Теория

## Актуальность

Мой проект является особенно актуальным для людей, которые занимаются олимпиадной биологией. Программа позволит участникам тренироваться и решать большее количество заданий, что повысит их шансы на успешное выступление. Генерация различных вариантов позволит ученикам ознакомиться с широким спектром тем и форматов, углубляя их понимание и знание предмета. Участники смогут сосредоточиться на решении заданий, не тратя время на поиск материалов, что сделает их подготовку более эффективной. Регулярная практика с использованием программы поможет участникам повысить их уверенность в своих силах перед олимпиадой.

## Проблема

Основной проблемой, с которой сталкиваются участники ВсОШ по биологии, является недостаток качественных и разнообразных тренировочных материалов, соответствующих формату заключительного этапа. Это ограничивает возможность глубокой подготовки, так как:

Учащимся трудно найти задания, которые бы отражали уровень сложности и специфику заключительного этапа олимпиады, что может негативно сказаться на их результатах.

Множество заданий может отличаться по уровню сложности и специфике, и необходимость их подбора вручную затрудняет процесс подготовки.

Эти проблемы нуждаются в решении, и разработка программы для генерации вариантов олимпиады по биологии может стать эффективным инструментом для их преодоления.

## Цель

Целью данного проекта является создание программного обеспечения, которое позволит генерировать разнообразные наборы тренировочных заданий, которые смогут помочь в подготовке к заключительному этапу ВсОШ по биологии.

## Задачи

**Разработка алгоритмов генерации заданий:** Создать алгоритмы, которые будут использоваться для автоматической генерации наборов тренировочных заданий разных типов.

**Создание базы данных:** Сформировать обширную базу данных с вопросами и задачами.

**Разработка пользовательского интерфейса:** Создать интуитивно понятный и удобный интерфейс для пользователей, позволяющий легко настраивать параметры генерации наборов заданий и получать результаты в удобном формате.

## Глава 2. Практика

### Написание программы

**Выбор языка программирования:** Выбор языка программирования - важный этап написания программы. Для начала я решил определить список языков программирования пригодных для выполнения задач проекта. Выбор пал на: C++, PHP и Python. Первым я решил проверить PHP, но для него требуется выделенная серверная часть, чего с бюджетом проекта добиться не получилось. Реализация всего запланированного на C++ заняла бы слишком много времени и усилий. Поэтому я решил выбрать Python. Python подошел для проекта лучше всего, потому что у него есть огромное количество библиотек, предназначенных для реализации всех задуманных функций программы. Например для реализации пользовательского интерфейса я использовал библиотеку DearPyGui.

**Создание тестовой базы заданий:** Чтобы создать базу заданий, нужно было сначала найти откуда эти задания брать, а затем преобразовать их в формат, который программа сможет обработать. Задания заключительного этапа ВСОШ по биологии прошлых лет находятся в открытом доступе. Для тестовой базы я взял вариант 2024 года. Далее я обработал его в читаемый программой формат и приступил к написанию кода.

**Написание кода программы:** Код, как говорилось выше, я писал на языке программирования Python. Сначала, я скачал все нужные мне библиотеки. Потом я приступил к написанию кода пользовательского интерфейса. Было очевидно, что для удобной работы с программой требуется только 5 полей ввода числовых данных, для обозначения количества заданий для каждой из 5 частей варианта, а также кнопка, при нажатии которой генерируется файл с

вариантом. Для реализации я воспользовался открытой библиотекой DearPyGui. Далее я приступил к написанию части программы, которая должна заниматься генерацией самого варианта. Вариант состоит из 5 частей в каждой из которых разные типы заданий. Соответственно код для генерации последовательности заданий также делится на 5 частей. В каждой из частей строится последовательность из не повторяющихся чисел. Эти числа представляют из себя номер задания в базе заданий. Далее по этим числовым последовательностям строится HTML файл в котором и будет итоговый вариант. После завершения генерации в файле zakl.html будет содержаться сгенерированный вариант. Этот файл можно открыть и распечатать при помощи стандартных средств браузера.

### Создание полной базы заданий

Задания, как говорилось выше, я брал из вариантов прошлых лет. Мне оставалось только превратить все варианты заключительных этапов ВсОШ по биологии за 2015-2024 год в читаемый программой формат. Это был один из самых длительных этапов. Набор заданий для базы заключался в однотонной работе, а затем в распределении всех заданий по части в которой он содержится. Таким образом, я собрал базу из: 459 заданий первой части; 218 заданий второй части; 129 заданий третьей части; 120 заданий четвертой части и 47 заданий пятой части. Базы таких размеров должно хватить для помощи в подготовке.

## Заключение

В ходе работы над проектом было создано программное обеспечение, которое позволит генерировать разнообразные наборы тренировочных заданий, которые смогут помочь в подготовке к теоретическому туру заключительного этапа ВсОШ по биологии.

У этого проекта также имеются и дальнейшие перспективы развития. Например:

- 1) Расширение базы данных заданий: Постепенное добавление новых заданий и тем, а также включение задач из различных источников и учебных программ.
- 2) Создание мобильного приложения: Разработка мобильной версии программного обеспечения для удобного доступа к заданиям и возможность заниматься в любое время и в любом месте.
- 3) Использование программы на различных курсах: Такое программное обеспечение может использоваться на различных курсах и УТС по подготовке к заключительному этапу ВсОШ по биологии. Например варианты сгенерированные программой можно давать ученикам для домашнего решения.

Программа уже используется в проекте Арсения Семенова, для генерации вариантов для обучения.

## Список литературы

- 1) <https://olimpiada.ru> сайт с задачами предудущих лет
- 2) <https://dearpygui.readthedocs.io/en/latest/> документация библиотеки DearPyGui



# Приложение

```

1 import dearpygui.dearpygui as dpg
2 import random as r
3
4 dpg.create_context()
5 dpg.create_viewport(title='Custom Title', width=1200, height=600)
6 dpg.set_global_font_scale(2)
7
8 def gen():
9     count_of_parts = [459, 218, 47, 0, 0]
10    num1 = dpg.get_value("1")
11    num2 = dpg.get_value("2")
12    num3 = dpg.get_value("3")
13    num4 = dpg.get_value("4")
14    num5 = dpg.get_value("10")
15    print(num1)
16    used = []
17    used2 = []
18    used3 = []
19    used4 = []
20    used5 = []
21    out = open("zakl.html", "w")
22    out.write("<html><body style='display: grid;'>\n")
23    # first part
24    out.write("<h1>Часть 1.</h1>\n")
25    for i in range(num1):
26        nn = r.randrange(1, count_of_parts[0])
27        while (used.count(nn) != 0):
28            nn = r.randrange(1, count_of_parts[0])
29        used.append(nn)
30        out.write("<div style='display: flex; >ch2>" + str(i + 1) + "</h2><img src='part1/" + str(nn) + ".png' style='width: 700px; margin-bottom: 50px;'></div>\n")
31    # second part
32    out.write("<h1>Часть 2.</h1>\n")
33    for i in range(num2):
34        nn = r.randrange(1, count_of_parts[1])
35        while (used2.count(nn) != 0):
36            nn = r.randrange(1, count_of_parts[1])
37        used2.append(nn)
38        out.write("<div style='display: flex; >ch2>" + str(i + 1) + "</h2><img src='part2/" + str(nn) + ".png' style='width: 700px; margin-bottom: 50px;'></div>\n")
39    # 3 part
40    out.write("<h1>Часть 3.</h1>\n")
41    for i in range(num3):
42        nn = r.randrange(1, count_of_parts[2])
43        while (used3.count(nn) != 0):
44            nn = r.randrange(1, count_of_parts[2])
45        used3.append(nn)
46        out.write("<div style='display: flex; >ch2>" + str(i + 1) + "</h2><img src='part3/" + str(nn) + ".png' style='width: 700px; margin-bottom: 50px;'></div>\n")
47    # 4 part
48    out.write("<h1>Часть 4.</h1>\n")
49    for i in range(num4):
50        nn = r.randrange(1, count_of_parts[3])
51        while (used4.count(nn) != 0):
52            nn = r.randrange(1, count_of_parts[3])
53        used4.append(nn)
54        out.write("<div style='display: flex; >ch2>" + str(i + 1) + "</h2><img src='part4/" + str(nn) + ".png' style='width: 700px; margin-bottom: 50px;'></div>\n")
55    # 5 part
56    out.write("<h1>Часть 5.</h1>\n")
57    for i in range(num5):
58        nn = r.randrange(1, count_of_parts[4])
59        while (used5.count(nn) != 0):
60            nn = r.randrange(1, count_of_parts[4])
61        used5.append(nn)
62        out.write("<div style='display: flex; >ch2>" + str(i + 1) + "</h2><img src='part5/" + str(nn) + ".png' style='width: 700px; margin-bottom: 50px;'></div>\n")
63    out.write("</body></html>")
64    print("generate")
65
66 with dpg.window(label="", width=1200, height=600):
67     dpg.add_input_int(tag="1", label="1")
68     dpg.add_input_int(tag="2", label="2")
69     dpg.add_input_int(tag="3", label="3")
70     dpg.add_input_int(tag="4", label="4")
71     dpg.add_input_int(tag="10", label="5")
72     dpg.add_button(label="Generate", callback=gen)
73
74 dpg.setup_dearpygui()
75 dpg.show_viewport()
76 dpg.start_dearpygui()
77 dpg.destroy_context()

```