

МИНОБРНАУКИ РОССИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Пермский государственный национальный
исследовательский университет»

Институт компьютерных наук и
технологий

ОТЧЁТ
по индивидуальной работе №2
по дисциплине «Язык программирования Python»
Вариант 12

Работу выполнила
Студентка группы ИТ-8 ,2025 1 курса
Рязанова Кристина Михайловна.
«12» июня 2025 г.

Работу проверила
Рубцова Марина Борисовна
«__» _____ 2025 г.

Пермь 2025

СОДЕРЖАНИЕ

| | |
|--|-----|
| Постановка задачи | 3 |
| Алгоритм решения..... | 4-5 |
| Тестирование..... | 6 |
| Код программы | 7 |
| Инструкция по применению стилей и оформлению работы..... | 8 |

Постановка задачи

1.Шарики. В одной компьютерной игре игрок выставляет в линию шарiki разных цветов. Когда образуется непрерывная цепочка из трех и более шариков одного цвета, она удаляется из линии. Все шарiki при этом сдвигаются друг к другу, и ситуация может повториться. Напишите программу, которая по данной ситуации определяет, сколько шариков будет сейчас уничтожено. Естественно, непрерывных цепочек из трех и более одноцветных шаров в начальный момент может быть не более одной. Входные данные: даны количество шариков в цепочке (не более 105) и цвета шариков (от 0 до 9, каждому цвету соответствует свое целое число). Выходные данные: требуется вывести количество шариков, которое будет уничтожено.

Примеры:

(входные данные) 5 1 3 3 3 2 (выходные данные) 3

(входные данные) 10 3 3 2 1 1 1 2 2 3 3 (выходные данные) 10

2.Определение алгоритма и идеи решения поставленной задачи

Все задачи подразумевают работу с динамическими структурами данных – списками (односвязные/двусвязные/циклические/стеки/очереди ...) или деревьями. Для реализации используем собственные классы. Специализированные библиотеки использовать нельзя! Обязательно наличие обработки исключений.

3.Написание и структурирование кода решения задачи

При написании кода следует придерживаться методологии ООП. Несоответствие код стайлу, принятому в группе, может быть основанием для оценивании работы в 0 баллов. В программе обязательно наличие дружественного интерфейса и защиты от некорректного пользовательского ввода.

Алгоритм решения

1.Инициализация:

- Создается экземпляр класса BallChain, который принимает список цветов шариков и инициализирует его как атрибут объекта.

2.Обработка цепочки:

- Метод process_chain управляет процессом удаления последовательностей. Он запускает цикл, в котором последовательно вызывает метод remove_sequences для удаления найденных последовательностей, пока они существуют

3.Поиск и удаление последовательностей:

- Метод remove_sequences проходит по списку шариков, ищет последовательности одинаковых шариков и удаляет их. Если удалены шарик, метод рекурсивно вызывается снова для поиска новых последовательностей, которые могли образоваться после удаления.

4.Пользовательский интерфейс:

- В функции main осуществляется ввод данных от пользователя. Программа проверяет корректность ввода, создает объект BallChain, обрабатывает цепочку и выводит количество удаленных шариков.

Обоснование выбранных структур и типов данных

1.Список (list):

- Списки используются для хранения цветов шариков в классе BallChain. Это позволяет динамически изменять размер списка, что необходимо для удаления последовательностей. Списки также обеспечивают удобный доступ по индексу и позволяют легко перебирать элементы, что важно для поиска последовательностей.

2. Целые числа (int):

- Целые числа используются для представления количества шариков и их цветов. Использование целых чисел также упрощает арифметические операции, такие как подсчет удаленных шариков.

3. Циклы и рекурсия:

- Цикл while используется для прохода по списку шариков и поиска последовательностей. Рекурсивный вызов метода remove_sequences позволяет эффективно обрабатывать случаи, когда после удаления одной последовательности могут возникнуть новые последовательности. Это

упрощает логику обработки, так как мы можем повторно использовать тот же метод для поиска новых последовательностей.

4. Обработка исключений:

- Ввод пользователя обрабатывается с использованием конструкции try-except, что позволяет ловить ошибки при некорректном вводе (например, если введены нечисловые значения). Это делает программу более устойчивой к ошибкам и улучшает пользовательский интерфейс, предоставляя пользователю четкие сообщения об ошибках.

5. Строки (str):

- Строки используются для обработки пользовательского ввода и вывода сообщений. Они обеспечивают удобный способ работы с текстом, что критично для взаимодействия с пользователем.

Тестирование

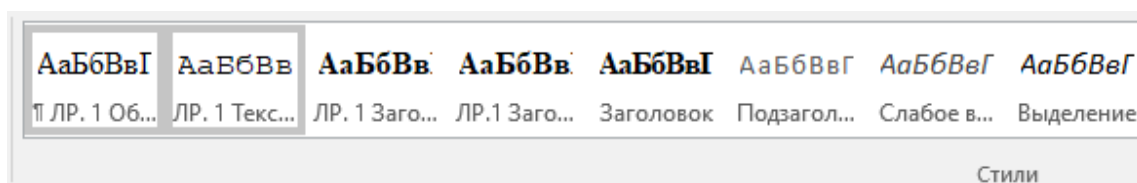
```
m4 README.md test.py x ikm.py .gitattributes
1 #test1
2 #Программа подсчета удаляемых шариков
3 #Введите количество шариков и их цвета через пробел
4 #Пример: 5 1 3 3 3 2
5 #Введите данные: 4 0 0 0 0
6 #Результат: 4
7 #Хотите проверить другую цепочку? (да/нет): да
8 #Введите данные:
9 #test2
10 #Введите данные: 6 1 1 1 1 2 2
11 #Результат: 4
12 #test3
13 #Введите данные: 5 1 2 3 4 5
14 #Результат: 0
15 #test4
16 #Введите данные: 5 1 3 3 3 2
17 #Результат: 3
18 #test5
19 #Введите данные: 10 3 3 2 1 1 1 2 2 3 3
20 #Результат: 10
21 #test6
22 #Введите данные:
23 #Ошибка: ввод не может быть пустым. Попробуйте снова.
24 #Введите данные:
25 #test7
26 #Введите данные: 4 1 2 2 3 3
27 #Ошибка: указано 4 шариков, но введено 5 цветов. Попробуйте снова.
28 #test8
29 #Введите данные: 4 a b c d
30 #Ошибка: все вводимые значения должны быть целыми числами. Попробуйте снова.
31 #test9
32 #Введите данные: 3 11 45 22
33 #Ошибка: цвета шариков должны быть числами от 0 до 9. Попробуйте снова.
34 #test10
35 #Введите данные: 0
36 #Ошибка: нужно ввести как минимум количество шариков и один цвет. Попробуйте снова.
37
```

Код программы

<https://github.com/Kriistarz/ikm>

Инструкция по применению стилей и оформлению работы

Для оформления частей отчёта следует использовать заранее созданные стили. Все стили, которые могут пригодиться, начинаются с «ЛР. 1 ...».



ЛР. 1 Обычный – для оформления текста задания и алгоритма решения.

ЛР. 1 Текст программы – для оформления кода программы.

ЛР. 1 Заголовок 1 – заголовок первого уровня (для того, чтобы озаглавить основные разделы отчета).

ЛР. 1 Заголовок 2 – заголовок второго уровня (для того, чтобы озаглавить подразделы).

Для того, чтобы перенести текст следующего блока на другую страницу, необходимо воспользоваться инструментом «Разрыв страницы» в разделе «Вставка».

