

# Индивидуальная работа №2 по дисциплине «Языки программирования»

Рязанова Кристина Михайловна ИТ-8

### Постановка задачи



- 1. Шарики. В одной компьютерной игре игрок выставляет в линию шарики разных цветов. Когда образуется непрерывная цепочка из трех и более шариков одного цвета, она удаляется из линии. Все шарики 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. При написании кода следует придерживаться методологии ООП. В программе обязательно наличие дружественного интерфейса и защиты от некорректного пользовательского ввода.

### Алгоритм 🖵

Ввод данных: \*Пользователь вводит количество шариков n и их

цвета.

\*Проверяется корректность ввода(число шариков соответствует количеству цветов,цвета в диапазоне 0-9) Создание цепочки: \*Создается объект BallChain с переданными шариками Обработка цепочки(process\_chain):

\*Инициализируется счетчик total\_removed=0
\*В цикле вызывается remove\_sequences до
тех пор,пока есть что удалять.

Повторение:

\*Пользователь может ввести новую цепочку или завершить программу. Вывод результата: \*После завершения обработки

выводится общее количество удаленных шариков(total\_removed). Удаление последовательностей(remove\_sequences): \*Поиск последовательностей из 3+ одинаковых

шариков.

\*Если найдена:

\*Удаление из цепочки.

\*Увеличение счетчика removed.

\*Рекурсивный вызов(т.к удаление могло создать новую последовательность).

\*Если последовательностей нет-возврат количества удаленных шариков.

### Тестирование 📳



### Тесты с корректным вводом:

```
#test1
#Программа подсчета удаляемых шариков
#Введите количество шариков и их цвета через пробел
#Пример: 5 1 3 3 3 2
#Введите данные: 4 0 0 0 0
#Результат: 4
#Хотите проверить другую цепочку? (да/нет): да
#Введите данные:
#test2
#Результат: 4
#test3
#Введите данные: 5 1 2 3 4 5
#Результат: 0
#test4
#Введите данные: 5 1 3 3 3 2
#Результат: 3
#test5
#Введите данные: 10 3 3 2 1 1 1 2 2 3 3
#Результат: 10
```

## Тестирование 🗐

#### Тесты с некорректным вводом:

```
#test6
#Введите данные:
#Ошибка: ввод не может быть пустым. Попробуйте снова.
#Введите данные:
#test7
#Введите данные: 4 1 2 2 3 3
#Ошибка: указано 4 шариков, но введено 5 цветов. Попробуйте снова.
#test8
#Введите данные: 4 a b c d
#Ошибка: все вводимые значения должны быть целыми числами. Попробуйте снова.
#test9
#Введите данные: 3 11 45 22
#Ошибка: цвета шариков должны быть числами от 0 до 9. Попробуйте снова.
#test10
#Введите данные: 0
#Ошибка: нужно ввести как минимум количество шариков и один цвет. Попробуйте снова.
```

# 

