Прототип программы содержится в папке Polymino и представляет собой несколько модулей:

1. main

Здесь происходит взаимодействие с пользователем, считывание из файла и запись в структуру данных. Заглушка тоже тут.

1. Polymino

В классе Polymino организована структура данных, включающих в себя двумерное

представление фигуры в виде массива, его площадь, количество идентичных фигур, размерность, строку группового представления, и проверку на корректность.

1. print, functions, errors, Coodinates

Вспомогательные структуры и методы

1. input\_check

Данный файл содержит основные методы проверки входных данных.

Контракт таков, что в первой строке содержится количество фигур. Далее идет primary\_key фигуры (для упрощения тестирования, в релизе не будет), размер массива и сам массив. Основные проверки входных данных связаны с тем, что в фигурах могут содержаться отверстия или в одном представлении могут содержаться несколько фигур. Также присутствуют проверки на некорректные данные в самом массиве (элементы отличные от 1 или 0).

Выходные данные имеют следующий вид:

* Строка содержит два числа:
  + Номер строки прямоугольника n
  + Количество новых фигур встречающихся на строке k
* Следующие k строк:
  + Id фигуры
  + Номер положения

Вариант не финальный, возможно поменяю форматы выходных данных.

В папках IOTest и OITest находятся тесты для входных и выходных данных соответственно.

Для IOTest в папке TestGenerator лежит программа генерирующая входные файлы, содержащие до 10000 объектов шириной до 15 единиц. Если подать такой файл на вход IOTest, то в выходном файле будет описан каждый из обьектов, и соотвественно содержащиеся в нем ошибки.

Для OITest в папке Tests лежат разные выходные файлы, которые можно подать на вход программе и на выходе получить данные о корректности файла.

Метод поиска отверстий:

Создается вектор и стек.

Для входного квадратного массива: идем по контуру массива и ищем нули. Для каждого найденного на контуре нуля применяем следующую операцию:

* В стек заносится координаты найденного нуля
* Пока стек не пуст:
  + Из стека извлекаются координаты элемента
  + Если их нет в массиве:
    - Добавляем их в массив
    - В стек добавляем координаты элементов, равных нулю и соседних с данным (соседние, значит общая сторона)
* Для каждого нуля из массива ищем его координаты в векторе
  + Если не нашли – значит элемент принадлежит отверстию

Метод поиска нескольких фигур в одном массиве

Создается вектор и стек.

Для входного квадратного массива: ищем первый элемент равный единице.

* В стек заносится координаты найденной единицы
* Пока стек не пуст:
  + Из стека извлекаются координаты элемента
  + Если их нет в массиве:
    - Добавляем их в массив
    - В стек добавляем координаты элементов, равных единице и соседних с данным (соседние, значит общая сторона)
* Для каждой единицы из массива ищем его координаты в векторе

Если не нашли – значит элемент принадлежит другой фигуре