Работу выполнил Ханнанов Альберт. 11-906 группа.

1. Класс Figure

Класс Figure отвечает за создание основной единицы нашей программы, вокруг которой построены остальные модули – фигура. В этом классе имеется три приватных поля: type – тип фигуры(Type), points – координаты фигуры(int[]), color – цвет фигуры(Color). Так же имеются два enum`a, которые отвечают за Type фигуры и за Color.

**Конструктор**

В данном классе имеется только один конструктор с параметрами: тип фигуры, координаты(4 аргумента подряд), цвет фигуры. Конструктор сравнивает переданную строку типа элемента и сопоставляет её значению из enum`а, так же происходит и с полем color. Если таких значений в enum`ах найдено не было, то мы выбрасываем исключение. *СЛОЖНОСТЬ ПО ВРЕМЕНИ И ПАМЯТИ О(1)*

**public double getArea() – получение площади фигуры**

Данный метод высчитывает площадь фигуры по формулам в зависимости от её типа. *СЛОЖНОСТЬ ПО ВРЕМЕНИ И ПАМЯТИ O(1)*

**public String getFigureInfo() – вывод всех полей фигуры (всей информации)**

Данный метод выводит в консоль всю информацию о фигуре. Здесь для экономии места массив points выводится через Arrays.toString(). *СЛОЖНОСТЬ ПО ВРЕМЕНИ И ПАМЯТИ**О(1)*

**СЛЕДУЮЩИЙ ЛИСТ**

2. Класс GraphicPicElement

Класс GraphicPicElement реализует элементы списка GraphicPic. Данный класс имеет два приватных поля: next – ссылка на следующий элемент в списке, figure – сама фигура. Так же данный класс реализует интерфейс Comparable<Figure>, т.к. для удобства в методе класса GraphicPic нам потребуется сравнивать элементы списка, а если быть точнее поле figure с фигурой.

**Конструктор**

Реализован одни конструктор, который принимает в параметр фигуру(Figure) и присваивает это значение полю figure, полю next присваивается значение null. *СЛОЖНОСТЬ ПО ВРЕМЕНИ И ПАМЯТИ О(1)*

**@Override public int compareTo(Figure figure)**

В данном переопределённом методе мы сравниваем поле figure, у элемента, для которого мы вызвали этот метод, с переданной в аргумент фигурой на все поля, то есть сравниваются поля type, points и color. *СЛОЖНОСТЬ ПО ВРЕМЕНИ И ПАМЯТИ О(1)*

**private boolean isAllPointsEquals(Figure figure)**

Данный метод проверят, все ли координаты у поля figure и переданной фигуры совпадают. Проходится циклом одновременно по двум массивам, у которых длина всегда равна 4 *СЛОЖНОСТЬ ПО ВРЕМЕНИ И ПАМЯТИ О(1)*

**СЛЕДУЮЩИЙ ЛИСТ**

1. Класс GraphicPic

Данный класс реализует список из фигур. Имеет приватные поля head и tail, которые хранят первый и последний соответственно элементы списка.

**Конструктор**

В данном классе имеется два конструктора: без параметров и с параметром «путь к текстовому файлу». Конструктор без параметров создаёт пустой список. СЛОЖНОСТЬ О(1);

Конструктор с параметром «путь к текстовому файлу» создаёт список из фигур, которые перечислены на каждой отдельной строчке в передаваемом файле. *СЛОЖНОСТЬ ПО ВРЕМЕНИ И ПАМЯТИ* конструктора с параметром *будет равна O(количество строк в файле)*

**public void show()**

Данный метод выводит в консоль все элементы списка(подробно про все фигуры). *СЛОЖНОСТЬ ПО ВРЕМЕНИ О(количество элементов в списке), А ПО ПАМЯТИ O(1)*

**public void insert(Figure figure)**

Данный метод уже имеет две кардинально разные реализации. Первая заключается в том, что если в списке нет эквивалентных, с переданной в аргумент фигурой, фигур, то просто переданная фигура добавляется в конец списка. Иначе же с помощью рефлексии и случайных чисел меняется одно из полей в имеющийся фигуре. *СЛОЖНОСТЬ О(1) – в лучшем случае*(когда фигура первого элемента в списке совпадает с переданной фигурой). *СЛОЖНОСТЬ О(количество элементов в списке)* – худший случай(когда в списке нет эквивалентных фигур и переданная фигура добавляется в конец списка), *НО СЛОЖНОСТЬ ПО ПАМЯТИ В ОБОИХ СЛУЧАЯХ БУДЕТ О(1)*

**public void delete(Figure.Type i)**

Данный элемент проходится по всем элементам в списке и убирает все ссылки на элементы, у которых тип фигуры совпадает с переданным в аргумент. *СЛОЖНОСТЬ ПО ВРЕМЕНИ О(количество элементов в списке), А СЛОЖНОСТЬ ПО ПАМЯТИ О(1)*

**public GraphicPic commonWith(Figure figure)**

Данный метод возвращает новый список фигур, которые пересекаются с прямоугольником figure. Данный функционал определяет пересечение фигур через проекции этих фигур и на оси координат. Причём в случае с кругом, то учитывается радиус. С отрезком всё несколько сложнее: мы составляем уравнение прямых: верхняя грань прямоугольника, нижняя, левая и правая; и уравнение прямой, на которой лежит этот отрезок, затем находим пересечение этих прямых с помощью геометрической формулы. Если данная точка находиться в пределах одной из граней прямоугольника и в пределах отрезка, то данные геометрические тела пересекаются. *СЛОЖНОСТЬ ПО ПАМЯТИ И ВРЕМЕНИ БУДЕТ РАВНА О(количество элементов в списке)*

**public GraphicPic hasSquareBiggerThanS(double s)**

Данный метод создаёт новый список GraphicPic из тех фигур, у которых площадь больше , чем переданное в аргумент число. *СЛОЖНОСТЬ ПО ПАМЯТИ БУДЕТ РАВНО В ХУЖШЕМ СЛУЧАЕ(когда все элементы в списке удовлетворяют условию): О(количество элементов в списке); И В ЛУЧШЕМ(когда либо один, либо ни один элемент не удовлетворят условию): О(1). ПО ВРЕМЕНИ ЖЕ ВСЕГДА БУДЕТ О(количество элементов в списке)*

**public GraphicPic getByColor(Figure.Color[] colors)**

Данный метод создаёт новый список GraphicPic из тех фигур, у которых значение цвета такое же, как переданное в аргумент значение. *СЛОЖНОСТЬ ПО ПАМЯТИ БУДЕТ РАВНО В ХУЖШЕМ СЛУЧАЕ(когда все элементы в списке удовлетворяют условию): О(количество элементов в списке); И В ЛУЧШЕМ(когда либо один, либо ни один элемент не удовлетворят условию): О(1). ПО ВРЕМЕНИ ЖЕ ВСЕГДА БУДЕТ О(количество элементов в списке)*

**public static int getRandomInteger(int a, int b)**

Возвращает случайное целое значение в диапазоне от а до b. Эта функция нужна для реализации метода instert(), а если быть точнее, то случая, когда нам требуется изменить параметры фигуры на случайное значение. *СЛОЖНОСТЬ ПО ВРЕМЕНИ И ПО ПАМЯТИ ВСЕГДА О(1)*

**private void add(GraphicPicElement nGPEtemp)**

Данный метод добавляет новый элемент в список. Конкретно тестирования этого метода нет, но он тестируется вместе с методами: insert(), getByColor(), hasSquareBiggerThanS(); т.к. там создаются новые списки и в них добавляются элементы с помощью этого метода. *СЛОЖНОСТЬ ПО ВРЕМЕНИ И ПО ПАМЯТИ ВСЕГДА О(1)*