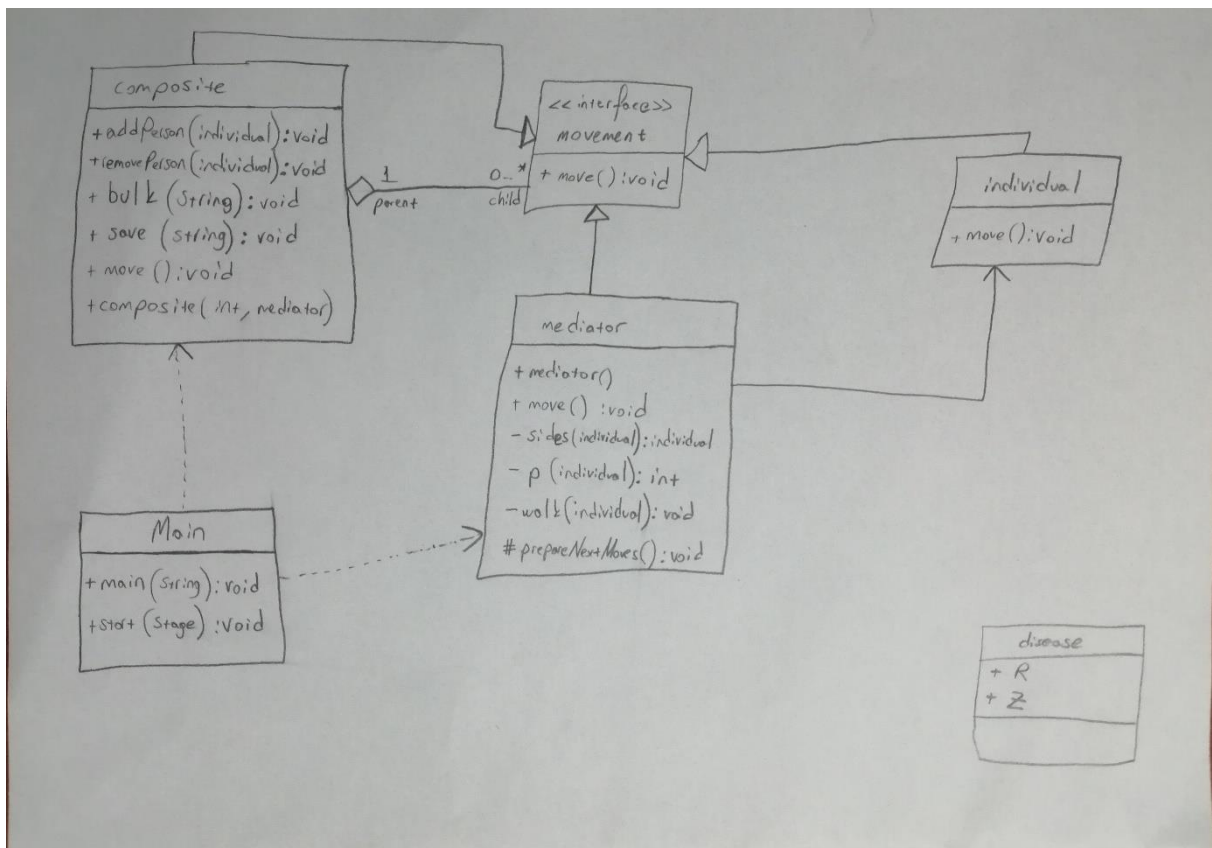


# OOAD Final Report

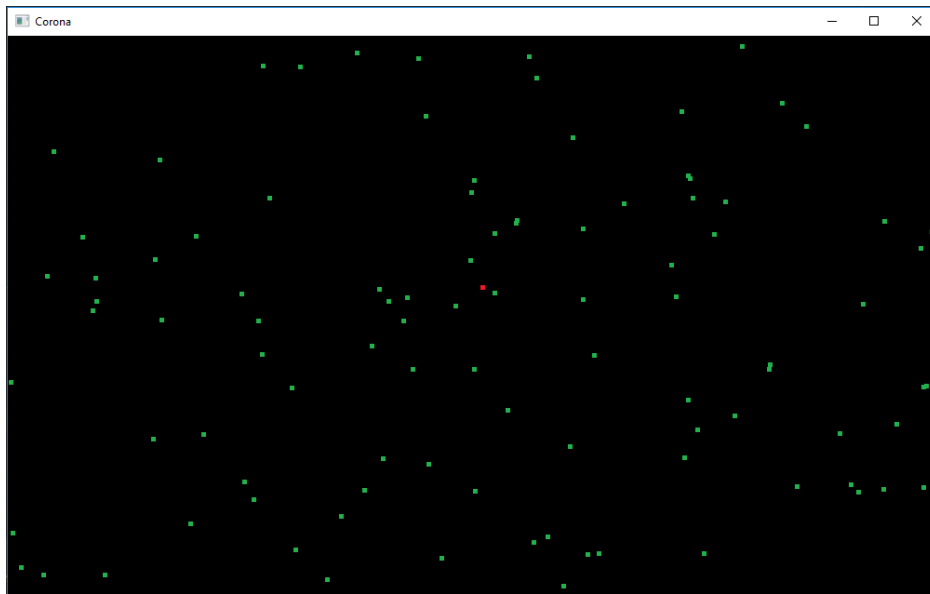
I preferred composite design pattern for creating individuals. Why?

- Every person in population differ from each other just by values of attributes. And there is only a notify method to communicate with mediator. So people tend to manage as a whole.
- Composite pattern easy my job from calling notify for all individuals separately to call notify of composite for once.
- Also factory design patterns are useful when there is type of or type of type of products i.e. specialized objects which is not our case.

## Diagram



Output of Main.main(String args):



This creates a file named save and save the objects to that file and retrieve them.

```
/**
 * Test for serializing and deserializing
 * @param args no need
 */
public static void main(String[] args){

    mediator m = new mediator();
    composite c = new composite(100,m);
    c.save("save");
    composite.people.clear();
    c.bulk("save");
}
```

Output of mediator.move(): This method performs required move on individual and javafx uptade it.

