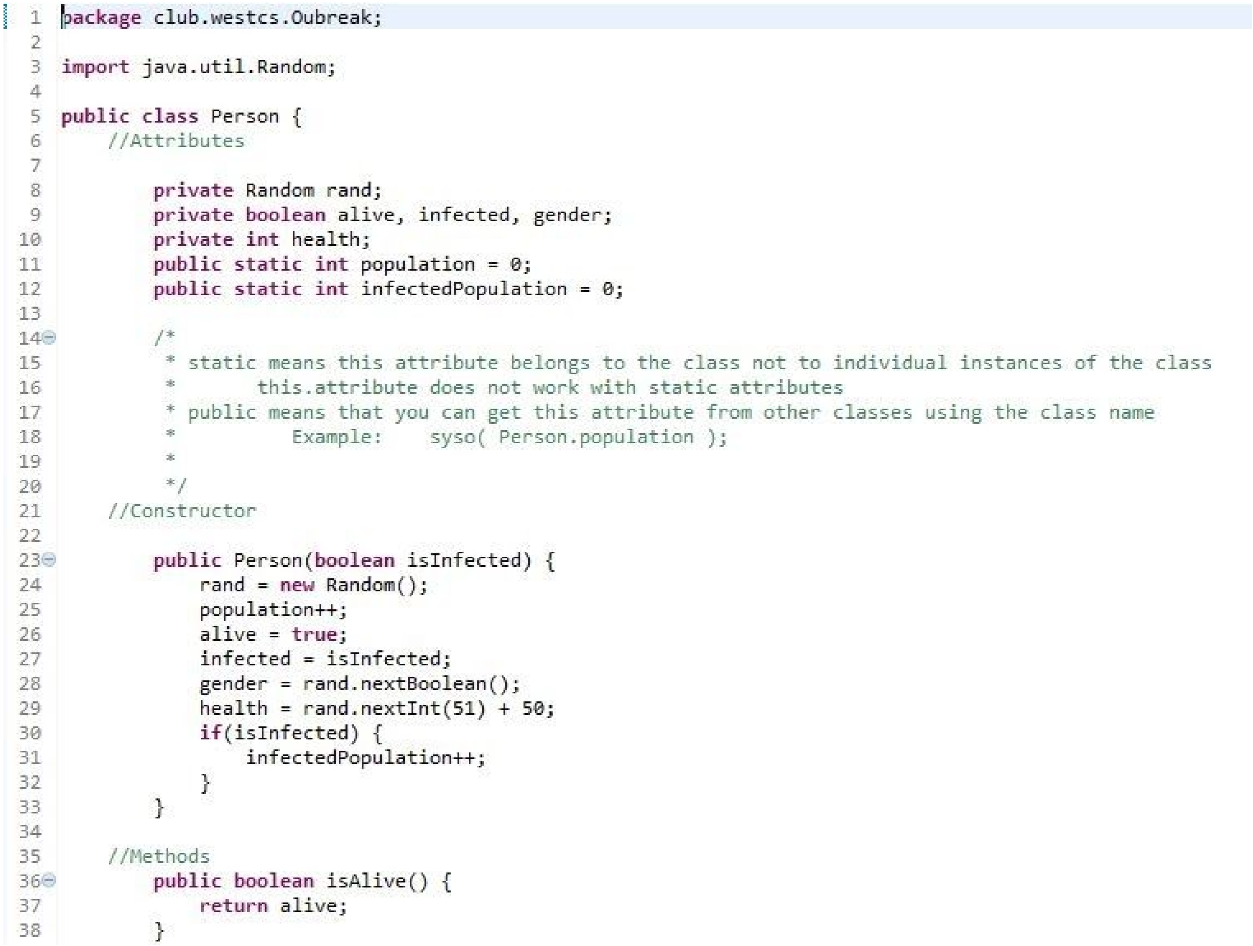
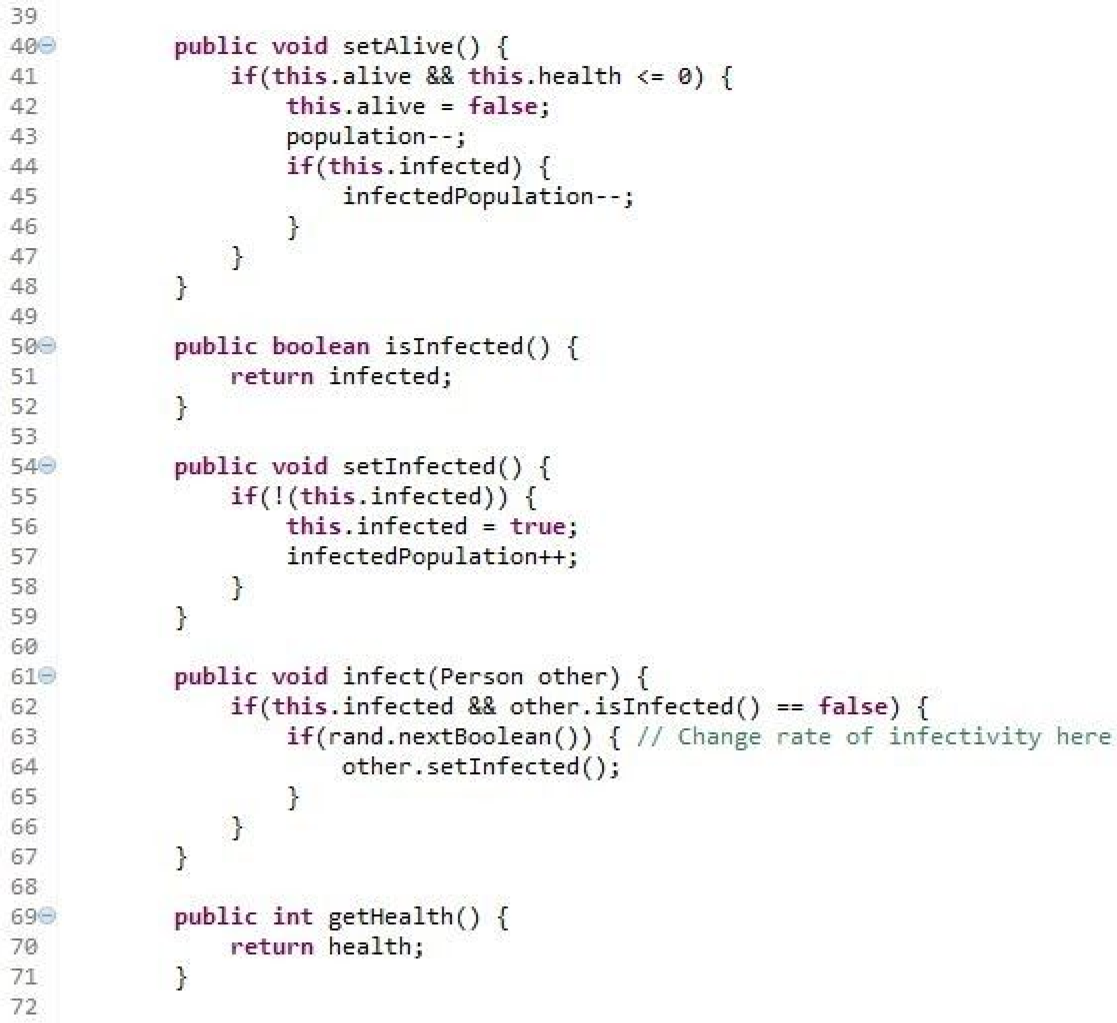
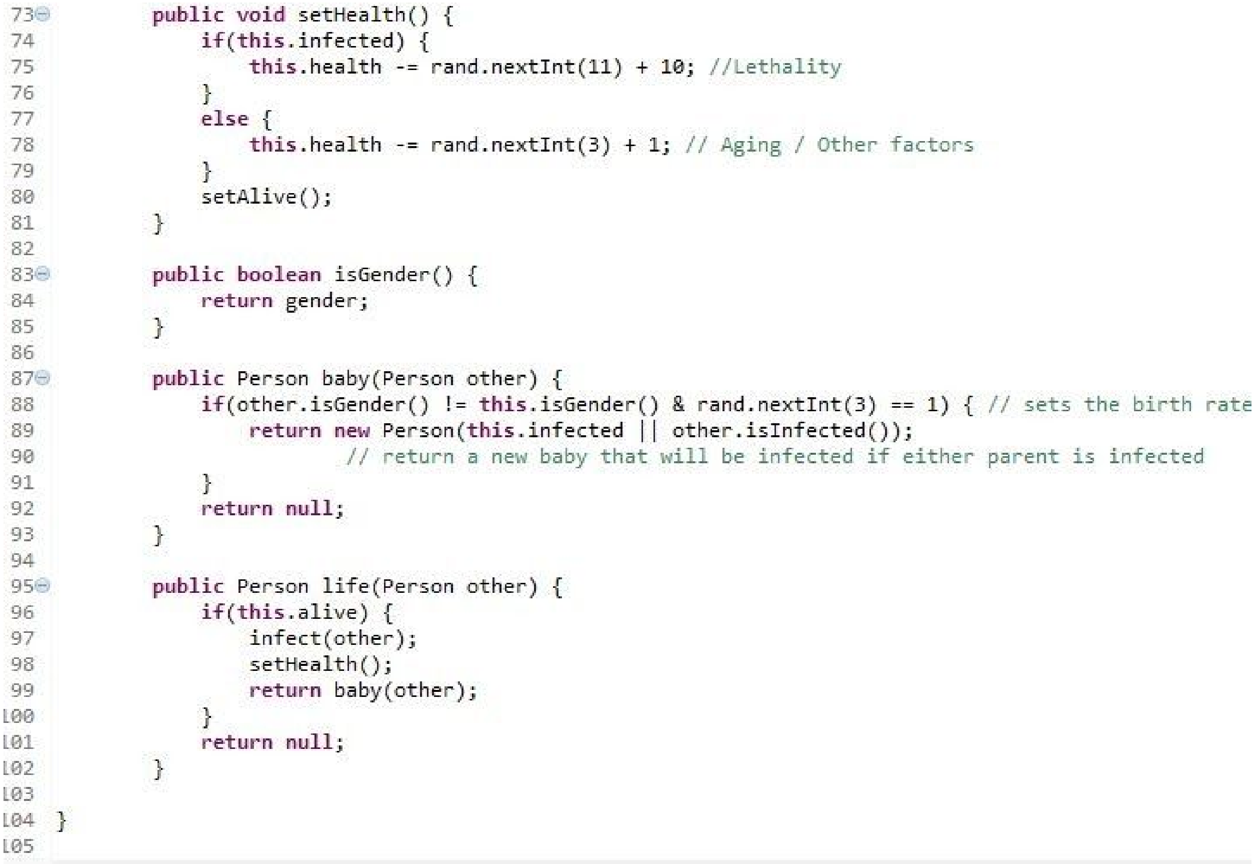


```java
package club.westcs.Oubreak;

import java.util.Random;

public class Person {
    //Attributes
    
        private Random rand;
        private boolean alive, infected, gender;
        private int health;
        public static int population = 0;
        public static int infectedPopulation = 0;
        
        /*
         * static means this attribute belongs to the class not to individual instances of the class
         *       this.attribute does not work with static attributes
         * public means that you can get this attribute from other classes using the class name
         *          Example:    syso( Person.population );
         * 
         */
    //Constructor
    
        public Person(boolean isInfected) {
            rand = new Random();
            population++;
            alive = true;
            infected = isInfected;
            gender = rand.nextBoolean();
            health = rand.nextInt(51) + 50;
            if(isInfected) {
                infectedPopulation++;
            }
        }
        
    //Methods
        public boolean isAlive() {
            return alive;
        }
```



```java

    public void setAlive() {
        if(this.alive && this.health <= 0) {
            this.alive = false;
            population--;
            if(this.infected) {
                infectedPopulation--;
            }
        }
    }

    public boolean isInfected() {
        return infected;
    }

    public void setInfected() {
        if(!(this.infected)) {
            this.infected = true;
            infectedPopulation++;
        }
    }

    public void infect(Person other) {
        if(this.infected && other.isInfected() == false) {
            if(rand.nextBoolean()) { // Change rate of infectivity here
                other.setInfected();
            }
        }
    }

    public int getHealth() {
        return health;
    }

```



```
        public void setHealth() {
            if(this.infected) {
                this.health -= rand.nextInt(11) + 10; //Lethality
            }
            else {
                this.health -= rand.nextInt(3) + 1; // Aging / Other factors
            }
            setAlive();
        }

        public boolean isGender() {
            return gender;
        }

        public Person baby(Person other) {
            if(other.isGender() != this.isGender() & rand.nextInt(3) == 1) { // sets the birth rate
                return new Person(this.infected || other.isInfected());
                        // return a new baby that will be infected if either parent is infected
            }
            return null;
        }

        public Person life(Person other) {
            if(this.alive) {
                infect(other);
                setHealth();
                return baby(other);
            }
            return null;
        }

}

```