

Software Security A.A. 2021-2022
Individual Project 2
Stanco Donato Francesco Pio 2027523

1- Considerations on the project

For this project I used the tool OpenJML to detect possible violations, and these are reported as “warnings”. One of the first difficulties was understanding how the tool worked, i.e. how the code and the invariants were analyzed.

Initially I corrected the warnings given by the tool and the first one concerned the attribute *DEFAULT_SUBSIDY* which was not assigned, so I decided to declare it in the class.

One of the warnings that the tool produced was related to the *spouse* object, which could not be null, but in Java objects can be null for this reason the *@nullable* annotation was added and then I put some invariants to describe the other properties (i.e. the requests of the project and other considerations made by me).

As mentioned in the lesson the attributes in Java must be private, to implement this I used the keyword */*@ spec_public */* which allows the attributes to have different visibility, as for Java they are private but for JML they are public, obviously also the invariants for logical reasons must be public.

The second part of the project concerns the rules for modifying the allowances based on marriage and age. To solve this scenario I added some lines of code to the methods *marry()*, *divorce()* and *haveBirthday()*, for the verification and calculation of the subsidy.

2 - Person.java

```
/*
This assignment illustrates how specifications such as invariants and
preconditions written in a formal language can help in removing
errors in code.

The assignment concerns a class "Person" that is used for Persons.
*/
class Person {

    /* isFemale is true iff the person is female */
    private /*@ spec_public @*/ boolean isFemale;
    /* isMale is true iff the person is male */
    private /*@ spec_public @*/ boolean isMale;
    /* NOTE: 1. Persons are either male or female. */
    //@ public invariant isFemale == true <==> isMale == false;
    //@ public invariant isFemale == false <==> isMale == true;

    //@ non_null
    private /*@ spec_public @*/ Person father, mother; // These fields won't really be used
    //@ public invariant father.isMale == true;
    //@ public invariant mother.isFemale == true;

    /* Age in years */
    public int age;
    //@ public invariant age >= 0;
    //@ public invariant age < Integer.MAX_VALUE;

    public boolean isMarried;

    /* Reference to spouse if person is married, null otherwise */
    //@ nullable;
    private /*@ spec_public @*/ Person spouse;
    //@ public invariant isMarried == true <==> spouse != null;
    /* NOTE: 2. Persons can only marry people of the opposite sex. */
    //@ public invariant isMarried == true && isFemale == true ==> spouse.isMale == true;
    //@ public invariant isMarried == true && isMale == true ==> spouse.isFemale == true;
    //@ public invariant spouse != null ==> age >= 18;

    /* NOTE: 3. If person x is married to person y, then person y should of course also be married and to person
x. */
    //@ public invariant spouse != null ==> this == spouse.spouse ==> (spouse.isMale != this.isMale ||
spouse.isFemale != this.isFemale);

    /* welfare subsidy */
    private /*@ spec_public @*/ int state_subsidy;

    public static final int DEFAULT_SUBSIDY = 500;
    public static final int NEW_SUBSIDY = DEFAULT_SUBSIDY + 100;
    //@ public invariant state_subsidy > 0;
    //@ public invariant state_subsidy <= NEW_SUBSIDY;

    //@ public invariant age > 65 && isMarried == false ==> state_subsidy == NEW_SUBSIDY;
    //@ public invariant age > 65 && isMarried == true ==> state_subsidy == NEW_SUBSIDY - (NEW_SUBSIDY * 30 /
100);
}
```

```

//@ public invariant age <= 65 && isMarried == false ==> state_subsidy == DEFAULT_SUBSIDY;
//@ public invariant age <= 65 && isMarried == true ==> state_subsidy == DEFAULT_SUBSIDY - (DEFAULT_SUBSIDY *
30 / 100);

/* CONSTRUCTOR */
//@ requires ma != null;
//@ requires ma.isFemale == true;
//@ requires pa != null;
//@ requires pa.isMale == true;
Person(boolean s, Person ma, Person pa) {
    age = 0;
    isMarried = false;
    this.isMale = s;
    this.isFemale = !s;
    mother = ma;
    father = pa;
    spouse = null;
    state_subsidy = DEFAULT_SUBSIDY;
}

/* METHODS */

/* Marry to new_spouse */
//@ requires new_spouse != null;
//@ requires new_spouse.spouse == null;
//@ requires new_spouse.isMarried == false;
//@ requires new_spouse.age >= 18;

//@ requires spouse == null;
//@ requires isMarried == false;
//@ requires age >= 18;
//@ requires isMale == true <==> new_spouse.isFemale == true;
//@ requires isFemale == true <==> new_spouse.isMale == true;

//@ ensures state_subsidy < \old(state_subsidy);
void marry(Person new_spouse) {
    spouse = new_spouse;
    isMarried = true;
    if(age > 65){
        state_subsidy = NEW_SUBSIDY - (NEW_SUBSIDY * 30 / 100);
    } else {
        //@ assert age <= 65;
        state_subsidy = DEFAULT_SUBSIDY - (DEFAULT_SUBSIDY * 30 / 100);
    }
}

/* Divorce from current spouse */
//@ requires isMarried == true;
//@ requires spouse != null;
//@ ensures state_subsidy >= \old(state_subsidy);
void divorce() {
    if( age > 65){
        state_subsidy = NEW_SUBSIDY;
    } else{

```

```

    //@ assert age <= 65;

    state_subsidy = DEFAULT_SUBSIDY;
}
spouse = null;
isMarried = false;
}

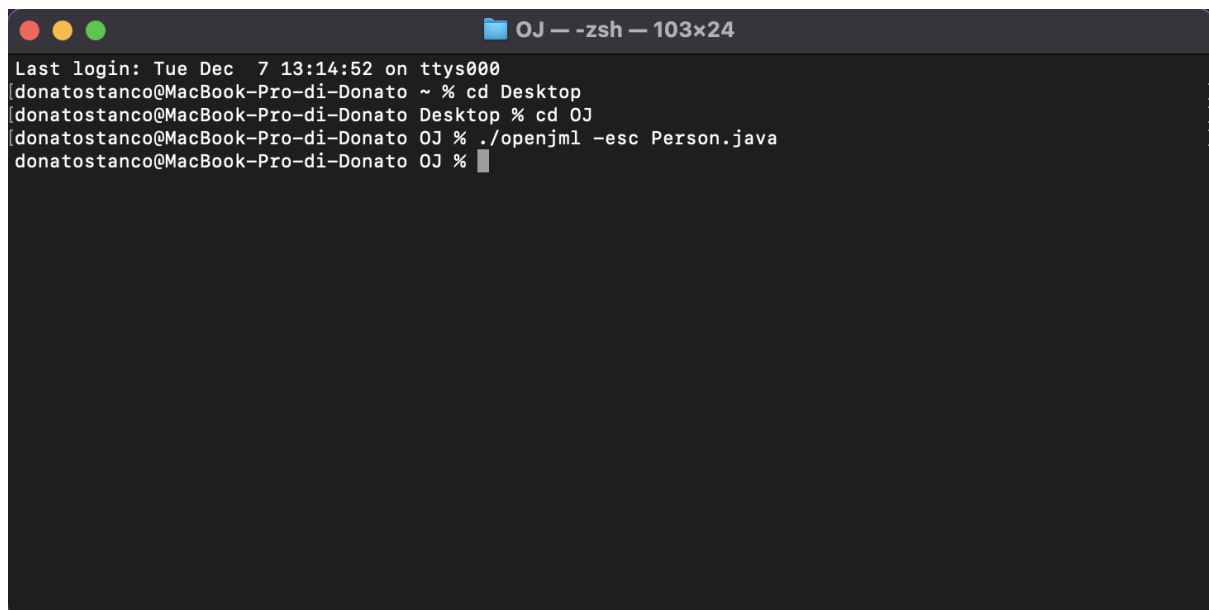
/* Person has a birthday and the age increases by one */
//@ requires age + 1 < Integer.MAX_VALUE;
void haveBirthday() {
    age++;
    if(age > 65){
        if(isMarried == false){
            state_subsidy = NEW_SUBSIDY;
        } else {
            //@ assert isMarried == true;
            state_subsidy = NEW_SUBSIDY - (NEW_SUBSIDY * 30 / 100);
        }
    }
}
}
}

```

Pastebin of the code: <https://pastebin.com/w7i8Pw3m>

3 - Screen of the tool

Adding invariants and some lines of code the OpenJML tool does not produce errors as shown in the following screen.



```

OJ — -zsh — 103x24
Last login: Tue Dec  7 13:14:52 on ttys000
donatostanco@MacBook-Pro-di-Donato ~ % cd Desktop
donatostanco@MacBook-Pro-di-Donato Desktop % cd OJ
donatostanco@MacBook-Pro-di-Donato OJ % ./openjml -esc Person.java
donatostanco@MacBook-Pro-di-Donato OJ %

```