Instituto Superior Técnico

Software Testing and Validation

School Year 2017/2018 — 2nd Semester

1st Test **A** - May 4, 2018 — Duration: 1:30h

- Clearly identify the first page of this test.
- Each answer should be given using solely the blank space that follows it.
- The back of each sheet can be used as scrap paper.

I. (1.0+0.75+0.75+1.0 = 3.5)

a) Draw the control flow graph for the following method, clearly identifying its several code segments.

```
public float doSomething(int x, int y) {
  String res = null;

  switch (x) {
    case 1:
      res = "1";
      break;

    case 2:
      res = "2";
      break;

    case 3:
      res = "3";
      break;
  }

  if (x > 3 && y >= 0)
    res = anotherFunc(x, y);
  else if (y <= 0)
    res = anotherFunc(x, 0);

  return res;
}
```

b) Show a minimal set of paths that achieves branch coverage for this method.

c) Show a minimal set of paths that achieves statement coverage for this method.

Number:________ Name:________

d) Which statements require branch coverage (can easily hide bugs even with a statement coverage of 100%)?

II. (7.0 val.)

Consider the following class *Habitat*

```
public class Habitat {
  ...
  Habitat(List<Animal> animals, int area, List<AnimalHandler> handlers) { /*... */}

  public List<Animal> getAnimals(); { /* ... */}
  public List<AnimalHandler> getHandlers() { /* ... */}
  public int getArea() { /* ... */}
  public boolean hasAnimal(Animal animal) { /* ... */}
  public int getNumberOfAnimals() { /* ... */}

  public void addAnimal(Animal animal) throws InvalidOperation { /* ... */}
  public boolean removeAnimal(Animal animal) throws InvalidOperation { /* ... */}
  public void addHandler(AnimalHandler handler) { /* ... */}
  public void removeHandler(AnimalHandler handler) throws InvalidOperation { /* ... */}
  public void changeArea(int newArea) throws InvalidOperation { /* ... */}

}
```

that represents the functionality of an habitat in an application that simulates a zoo. In this context, a Zoo is made of habitats, handlers and animals. Each habitat has a set of animals, an area and handlers. An habitat must always have at least one animal, and by security reasons cannot have more than 100 animals. Each habitat must have at least two handlers and a minimal area equal to 100 m^2. The amount of space per animal in an habitat must be greater than or equal to 10 m^2/animal.

Draw the test suite that checks the behaviour of this class using the appropriate class scope test pattern. Describe the various steps of the applied test pattern. Finally, implement two **successful** test cases of the designed test suite using the *TestNG* framework. For the implementation of these test cases, you can assume (if needed) any interface for the classes *Animal*, *AnimalHandler* and *Zoo*.

III. (4.5 val.)

a) Consider the following method of *Animal* class:

```
public int computeSatisfaction() throws InvalidOperation
```

that is responsible for computing the satisfaction level of the animal. The satisfaction level of an animal depends on several parameters: number of animals in the same habitat, state of health of the animal, area of the habitat of the animal and number of handlers assigned to the habitat of the animal. The computation of the satisfaction is performed according to following rules:

- If animal is ill, then the satisfaction level is -1;
- If the animal is in good health and it is alone in the habitat, then the satisfaction level is 0.5 if the number of handlers is less than 5 and 2 otherwise;
- If the animal is in good health and there are several animals in the habitat, then
  - the satisfaction level is 20 if the number of animals is lower than 20, the number of handlers is greater than or equal to 5 and the area of the habitat is less than 400 m^2;
  - the satisfaction level is 20 if the number of animals is lower than 20, the number of handlers is lower than 5 and the area of the habitat is less than 400 m^2;
  - the satisfaction level is 50 if the number of animals is lower than 20 and the area of the habitat is greater than 400 m^2;
  - the satisfaction level is 30 if the number of animals is greater than or equal to 20 and the area of the habitat is less than 400 m^2;
  - the satisfaction level is 30 if the number of animals is greater than or equal to 20 and the area of the habitat is greater than 400 m^2;

Using the method test pattern more appropriate, and describing the several steps of the test pattern applied, draw the test suite that checks the correct behaviour of this method.
If you apply the *Combinational Function Test* or the *Category Partition Test* to solve this question, then in the **last step of each pattern** you only need to design the domain matrix for the first two variants (*Combinational Function Test*), or show the first 10 test cases resulting from the application of the *Category Partition Test*.

IV. (2.5 + 2.5 = 5.0 val.)

a) Suppose you have defined a test suite for each method defined in a given class. Now consider a subclass of this class, For each method available in the subclass, show under what conditions you do not need to test the method, you can reuse the test suite of the super class, you need to add some new test cases or you need to implement a completely test suite.

b) Describe the method test pattern *Recursive Method Test*.