

Modular and Object Oriented Programming
Computer Science Engineering
UPV/EHU

Project

Serengeti Natural Park

Iyán Álvarez

erman ta zabal zazu



Universidad Euskal Herriko
del País Vasco Unibertsitatea

April 20, 2020

Contents

1	Introduction	1
2	Part 1	2
2.1	Summary	2
2.2	Designing	2
2.3	Coding	2
2.4	New attributes and methods	3
2.5	Name changes	3
2.6	Testing	4
2.7	Simulating	5
2.8	Project design	6
2.8.1	Classes design	6
2.8.2	Package design	6
3	Part 2	7

1. Introduction

In this document there is an explanation of the project that we have done on the MOOP subject on first grade of Computer Science Engineering at EHU/UPV university.

The programming language that we are using is Java. We have learn some other interesting tools such as JavaDoc or JUnit 5.

To develop this project I have used IntelliJIDEA as IDE and Git for VCS.

This projects purpose is to work on concepts that we have been working on the subject.

- **Part 1**

- Attributes
- Constructors
- Methods
- Packages
- Encapsulation
- Singleton
- ArrayLists
- Exceptions
- Files
- Junit 5 tests

- **Part 2**

2. Part 1

2.1 Summary

Este proyecto se ha realizado con el objetivo de poner en práctica diferentes conocimientos sobre el subsistema de entrada/salida de la consola Nintendo DS. Para ello, se han ido adquiriendo dichos conocimientos mediante las prácticas y laboratorios realizados durante el curso.

2.2 Designing

I have made a project diagram using the tool StarUML, organizing how we were going to implement all the project and what are the attributes, constructors, methods... that have to be implemented in each class.

I have found some aesthetic problems at the time of showing the parent package name (pack::class) or showing the Exception that throw a method, although I have added it on the specific part as was required. Despite these issues I have organized the diagram with the more dependent classes at the top (the ones that need other classes to function), and the more independent classes at the bottom (the ones that only need primitive types and default Java classes).

After finishing with the diagram, using an extension on StarUML I generated all the project “code skeleton”. With this last step I finished the part of the basic design of the project.

2.3 Coding

Later, I have implemented and documented all the attributes, constructors and methods shown in the StarUML diagram for GPS, Drone, Specimen and SerengetiMonitor classes. The class TrackingDevice was given by the MOOP teachers.

Exceptions were created in another package; this is the first contact that we are having with Java hierarchy. These exceptions extend the Exception class. This means, Exception is the super-class of the classes that we are creating, these are the sub-classes. In my opinion, it is a good practice to implement a main program in each of the classes to test the classes little by little, and if an error is found, it can be easily fixed.

After finishing with the implementation and little class simulations I had changed some parts of the diagram to work better or for being clearer. *

2.4 New attributes and methods

In my opinion, I think that some attributes, methods were important to implement or necessary to made testing or software maintenance easier. These are the methods that I have personally added to the first part of the project:

- **GPS class**
 - equals()
- **Drone class**
 - private final int IMAGE_WIDTH = 1000
 - private final int IMAGE_HEIGHT = 1000
- **Specimen class**
 - public GPS initialLocation() ...
 - public void clearDayRoute() ...
 - equals();
- **SerengetiMonitor class**
 - private final int COLLECT_IMAGE_ATTEMPTS = 5

2.5 Name changes

I have changed some methods names for better comprehension.

- **Specimen class**
 - lastPosition() → lastLocation()
 - public void clearDayRoute() ...
 - equals();
- **SerengetiMonitor class**
 - closestSpecimen() → closestSpecimenList()
 - captureWithAttempts() → captureWithAttempts()
 - collectImage() → collectImagesClosest()
 - updatePositions() → updatePositionsSpecimens()

2.6 Testing

After finishing the implementation of the basic Classes, we have developed some tests for the methods that were asked to be tested. For this task we are using the tool JUnit 5, the purpose of the tests is to verify the functioning of some methods. In some cases, the expected case is manually created and we compare it with the real one, this means, the one returned by the method.

I have implemented 5 classes to make the tests.

- **GPSTest:** 2 cases for `distanceTo()`
- **SpecimenTest:** 3 cases for `kmsTraveled()`
- **SerengetiMonitorTestDefault:** 2 cases for `closestSpecimenList()`
- **SerengetiMonitorTestDrones:** 2 cases for `captureWithAttempts()` and other cases for `collectImage()`, with Drones at the park.
- **SerengetiMonitorTestNoDrones:** 2 cases for `captureWithAttempts()` and other 2 cases for `collectImage()`, with no Drones at the park.

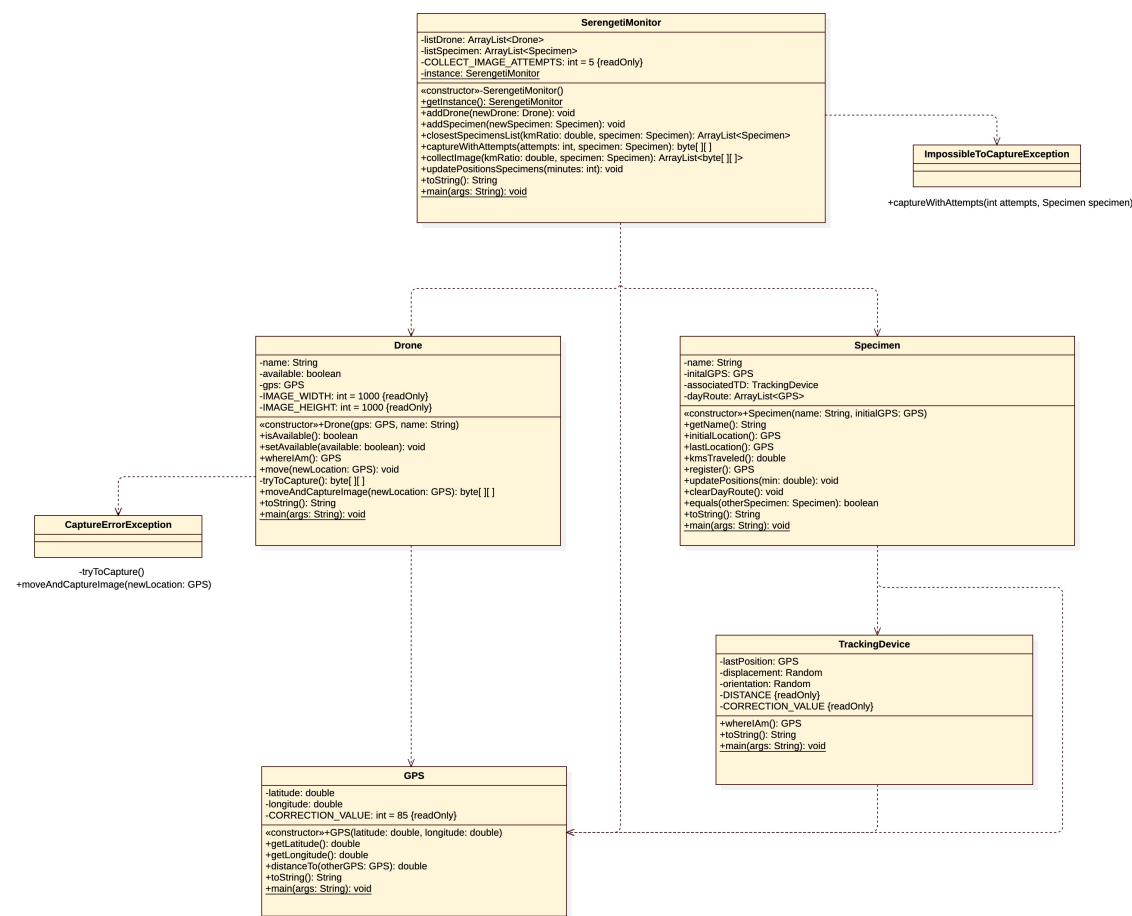
2.7 Simulating

To finish with this part of the project we have created a `SerengetiSimulator` class. The purpose of this class is to simulate a situation where some Drones and Specimens are on the Serengeti National Park and we can monitor what are they doing and send them actions, such as simulating the movement of all the specimens at the park, a drone moving and taking photos of a concrete specimen or knowing what are the specimens nearby a specimen in an specific ratio. Each simulation is different, some methods (the ones that simulate movement are random). Here is one of the simulations done:

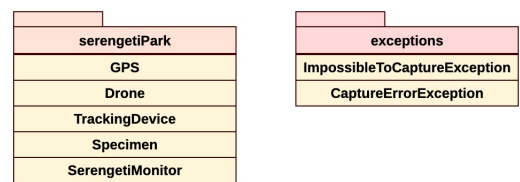
<p>Trying to create <code>SerengetiMonitor</code> instance. Created successfully.</p> <p>Trying to add 10 Specimens to the park on the given coordinates. Showing <code>SerengetiMonitor</code> status after adding 10 Specimens. <code>SerengetiMonitor</code>: - Drones: () - Specimens: (Specimen1 -2.949865 35.062473 Specimen2 -3.049865 34.962472999999999 Specimen3 -2.749865 35.262473 Specimen4 -3.249865 34.762473 Specimen5 -2.549865 35.462472999999999 Specimen6 -3.449865 34.562473 Specimen7 -2.349865 35.662473 Specimen8 -3.649865 34.362472999999994 Specimen9 -2.149865 35.862472999999994 Specimen10 -3.849865 34.162473)</p> <p>Simulating 6 hours of movement registers on all the Specimens in the park. Showing <code>SerengetiMonitor</code> status after 6 hours of movement in the park. <code>SerengetiMonitor</code>: - Drones: () - Specimens: (Specimen1 -2.3473567242871005 34.979283966919226 Specimen2 -2.8490289080957005 35.26077110658032 Specimen3 -2.818781051485518 35.6641451838086 Specimen4 -3.4162430661615404 34.145691742691845 Specimen5 -2.598596007338011 35.44228795585249 Specimen6 -3.5187810514855187 34.32717888235294 Specimen7 -2.63389012498507 35.99522913232308 Specimen8 -3.7960580220140336 34.127178882352936 Specimen9 -2.6691842426321295 35.69609493383845 Specimen10 -3.8498649999999994 34.39776711764706)</p> <p>Showing the data of the Specimens in the ratio of 300km to the first Specimen added. Specimen2 -2.8490289080957005 35.26077110658032 48.89605992632275km. Specimen3 -2.818781051485518 35.6641451838086 70.67154681919236km. Specimen4 -3.4162430661615404 34.145691742691845 115.21793136833166km. Specimen5 -2.598596007338011 35.44228795585249 44.77603397816152km. Specimen6 -3.5187810514855187 34.32717888235294 113.95948432979121km. Specimen7 -2.63389012498507 35.99522913232308 89.72417246621036km. Specimen8 -3.7960580220140336 34.127178882352936 142.861698272884km. Specimen9 -2.6691842426321295 35.69609493383845 66.7880928956429km. Specimen10 -3.8498649999999994 34.39776711764706 136.9448124940891km.</p>	<p>Adding 3 drones to the park, the first one is not available. Showing <code>SerengetiMonitor</code> status after adding the Drones. <code>SerengetiMonitor</code>: - Drones: (Drone1, false, -2.949865 35.062473 Drone2, true, -2.949865 35.062473 Drone3, true, -2.949865 35.062473) - Specimens: (Specimen1 -2.3473567242871005 34.979283966919226 Specimen2 -2.8490289080957005 35.26077110658032 Specimen3 -2.818781051485518 35.6641451838086 Specimen4 -3.4162430661615404 34.145691742691845 Specimen5 -2.598596007338011 35.44228795585249 Specimen6 -3.5187810514855187 34.32717888235294 Specimen7 -2.63389012498507 35.99522913232308 Specimen8 -3.7960580220140336 34.127178882352936 Specimen9 -2.6691842426321295 35.69609493383845 Specimen10 -3.8498649999999994 34.39776711764706)</p> <p>Trying to take photos of the Specimens nearby the first added Specimen in the ratio of 90km. Drone2 moving to -2.8490289080957005 35.26077110658032 ... arrived. Trying to take capture ... Trying to take capture ... image was taken successfully. Drone2 moving to -2.8490289080957005 35.26077110658032 ... arrived.</p> <p>Drone2 moving to -2.818781051485518 35.6641451838086 ... arrived. Trying to take capture ... Trying to take capture ... Trying to take capture ... image was taken successfully. Drone2 moving to -2.818781051485518 35.6641451838086 ... arrived.</p> <p>Drone2 moving to -2.598596007338011 35.44228795585249 ... arrived. Trying to take capture ... Trying to take capture ... image was taken successfully. Drone2 moving to -2.598596007338011 35.44228795585249 ... arrived.</p> <p>Drone2 moving to -2.63389012498507 35.99522913232308 ... arrived. Trying to take capture ... image was taken successfully. Drone2 moving to -2.598596007338011 35.44228795585249 ... arrived.</p> <p>Drone2 moving to -2.6691842426321295 35.69609493383845 ... arrived. Trying to take capture ... Trying to take capture ... Trying to take capture ... Trying to take capture ... Trying to take capture ... Specimen9 couldn't be photographed. Drone2 moving to -2.598596007338011 35.44228795585249 ... arrived.</p> <p>1 of 5 Specimens couldn't be photographed. 5 specimens are in the ratio of 90km to the first added specimen. 4 images have been taken.</p> <p>Process finished with exit code 0</p>
--	---

2.8 Project design

2.8.1 Classes design



2.8.2 Package design



3. Part 2

Not done yet.