**FINAL Exam: OOP**

**WEP 2021**

**{3h00}**

NOV 2021

* All documents allowed
* Chatting and talking to other students are forbidden

|  |  |
| --- | --- |
| EXERCICES | POINTS |
| THEORY | 25 |
| PROBLEM 1 | 25 |
| PROBLEM 2 | 50 |
| **TOTAL** | **100** |

**You need to return on google classroom:**

* **A ZIP file containing all necessary files**
* **The ZIP file should be named :**

**<YOU FIRST NAME>-<YOU LAST NAME>.ZIP**

## THEORY – 25 POINTS

**Q1** 5 POINTS - What is the advantage of a statically typed language such as TypeScript?

*(The fact that types are declared in the code)*

**Q2** 5 POINTS - Give one reason why encapsulation is useful

*The reason why encapsulation is useful. Because encapsulation is used to hide the values or state of a structured data object inside a class. In an encapsulation there are three types (Public, Protected, Private).*

**Q3** 5 POINTS - Give three differences between an interface and an abstract class

*The differences between an interface and an abstract class:*

*Abstract class is a class that have abstract methods. Abstract class allows us to create function and can implement or override.*

*Interface allows us to create function but can’t implement.*

**Q4** 5 POINTS - If I have an attribute X with a protected visibility, what does that imply?

*If you have an attribute X with a protected visibility. It means that any class inside can access and use this attribute X. Like if you are a parent class and have protected attribute and I’m a child class I can access attribute from you.*

**Q4** 5 POINTS -

Draw the class diagram for this code:

* Write your UML on the **DIAGRAM.pptx** file
* interface Drivable
* interface FastDrive implement Drivable
* abstract class Maintenance{}
* Car extends Maintenance implements Drivable
* Truck extends Maintenance implements FastDrivable

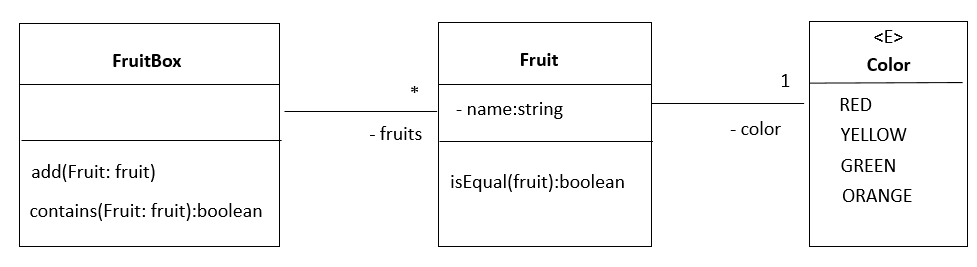
## PROBLEM 1 – 25 POINTS



A fruit box contains fruits. But all fruits in this fruit box **must be different.**

A fruit is defined by a color and a name.

* So for example, the fruit box **cannot** contain 2 banana YELLOW.



The fruit box contain a method to add a fruit.

If no fruit similar exists in the fruit box, the fruit is added

**// EXAMPLE**

Let banana1 = new Fruit(“Banana”, Color.YELLOW);

Let apple = new Fruit(“Apple”, Color.GREEN);

Let banana2 = new Fruit(“Banana”, Color.YELLOW);

fruitBox.add(banana1);

fruitBox.add(apple);

fruitBox.add(banana2);

**// Only banana1 and apple add added because the box contains already banana2**

*Note: The code of the classes is already given.*

Q1 – 5 POINTS - Implement the following method in the class Fruit

isEqual(fruit):boolean

Q2 – 10 POINTS - Implement the following method in the class FruitBox

contains(Fruit: fruit):boolean

Q3 – 10 POINTS - Implement the following method in the class FruitBox

add(Fruit: fruit)

## PROBLEM 2 – 50 POINTS



Our application requires to manage **student’s results** in a school. A school has a name and an address.

The school manage many students and each student is defined by his/her first name and last name.

The school manage many topics. Each topic has a category (JAVA, HTML, ANDROID, OOP, and VUEJS) and a year (number).

The school system keep tract of **the student result of each topic**

* The result the final exam score, from 0 to 100

Your system shall be able to do the following operations:

* Add a student, topic, score
* Compute the score average for a given students on every topic he/she passed

getAverageFor(student : Student) : number

* Compute the score average of all students on a given topic

getAverageFor( topic : Topic) : number

**Q1** – 15 POINTS - Design the UML class diagram corresponding to your solution for this problem

* + Write your UML on the **DIAGRAM.pptx** file

**Q2** – 10 POINTS - Implement UML class on TS

* + Implement : Add a student, topic, score

**Q3** – 20 POINTS - Implement the following operations on School class:

* + getAverageFor(student)
  + getAverageFor(topic)

**Q4** – 05 POINTS – Write a main.ts with appropriate tests of the system operations