**FINAL Exam: OOP**

**WEP 2024**

**{3h00}**

11th April 2024

* All documents allowed during practice time.
  + Search google allowed.
  + Artificial Intelligence (AI) are forbidden.
* Chatting and talking to other students are forbidden.

|  |  |
| --- | --- |
| EXERCICES | POINTS |
| THEORY | 20 |
| EXERCICE 1 | 30 |
| EXERCICE 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 - 20 POINTS

Students, please ensure you have completed the theoretical aspects before proceeding to practical exercises.

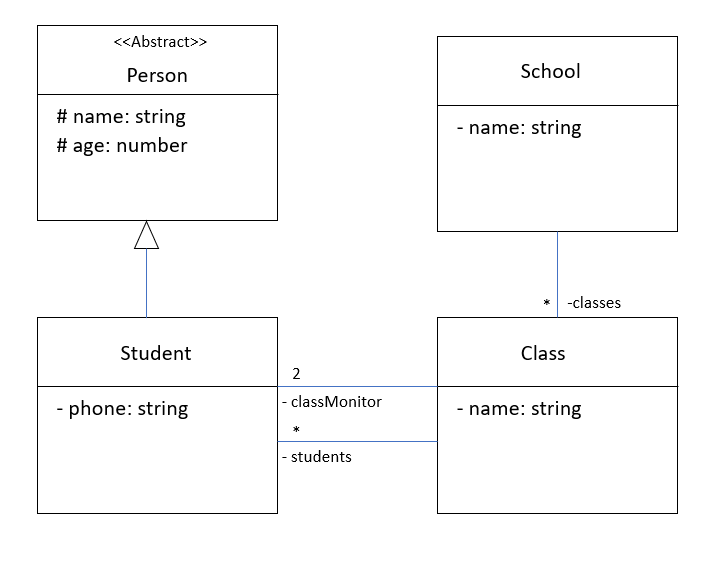
# EXERCICE1 - 30 POINTS

Open the START CODE / EXERCICE-1

Convert the bellow diagram below to **TypeScript** code, including:

* Attributes (visibility, type)
* Constructors …

*This model represents a school system, consisting of schools, classes within each school, and students assigned to each class.*



# EXERCICE2 - 50 POINTS



**"We're setting up a system to run our restaurant smoothly. Customers can come to our restaurant and pick what they want to eat, like pizza or a meal, and order it easily."**

1. **MenuItem**

* In a restaurant, this class represents the different dishes available on the menu.
* Each MenuItem object has a name, description, and price.
* Customers can view the menu, which includes all MenuItem objects.

For example,

|  |  |  |
| --- | --- | --- |
| name | description | price |
| Big Meal | A delicious burger with fries and a drink. | $9.99 |
| Crispy Snack | Crunchy chips served with a dip. | $7.99 |

1. **Customer**

* Each Customer object has an ID, name, and a list of orders they've placed.
* Customers can place orders, selecting items from the menu.
* One customer can make multiple orders, but each order is placed by only one customer.

For example,

|  |  |  |
| --- | --- | --- |
| ID | name | Orders |
| 101 | John | [Order 1] |
| 102 | Alice | [Order 2, Order 3] |

1. **Order**

* Each Order object contains a list of MenuItem objects and calculates the total price of the order

For example,

|  |  |  |
| --- | --- | --- |
| Order Items | Total Price | customer |
| [Big Meal, Crispy Snack] | $17.98 | John |
| [Big Meal, Big Meal, Crispy Snack] | $29.97 | Alice |

1. **Restaurant**
   * In the restaurant, we store items and orders, such as MenuItem and Order objects.

* We can let people order menu items directly, without needing a customer.

**Q1 – 10 POINTS**

On DIAGRAMS, write the UML diagram corresponding to your solution

**Q2 – 10 POINTS**

On exercise 2 folder, write classes attributes and constructors corresponding to your solution for this problem

**Q4** – 05 POINTS

In the restaurant, write a method to add item to restaurant.

addItem(item: MenuItem): void

**Q5** – 10 POINTS

In the restaurant, write a method to order item by customer.

placeOrder(customer: Customer, items: MenuItem[]): void

**Q6**– 10 POINTS

In the restaurant, write a method to get order by customer

getOrdersByCustomer(customer: Customer): Order[]

**Q7– 05 POINTS**

Let's create an instance of the Restaurant class and test.