QUIZ CHAPTER 3

{2h00}

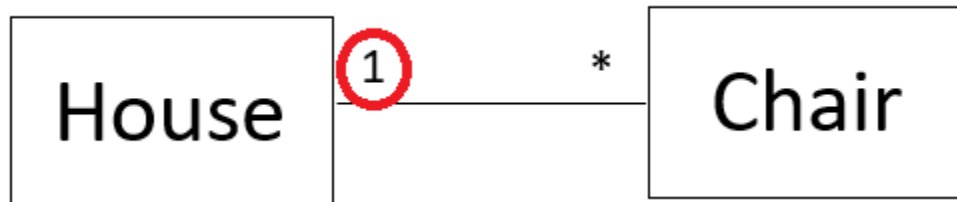| EXERCICES	POINTS
EXERCICE-1	20
EXERCICE-2	80

You need to submit on google classroom:

1. Your UML diagrams
2. Your Typescript project

EXERCICE-1

A house has many chairs

We also want to keep for each chair, to which house it belong to: you need to make the connection between the Chair and the house!

Q1   Complete the method addChair to:

- Add a new chair to the House
- Set the **House reference** on the new created chair

```
/**
 * Add a chair of given ID
 * @param chairID the chair ID
 * @return the created chair
 */
public addChair(chairID: number) : Chair {


    // TODO YOUR CODE


}
}
```

EXERCICE-2

We describe a library system as follows:

LIBRARY has :
- a name
- An address
- Books in the library

BOOK has :
- a title
- year of published
- authors, a book can be written by one or many authors
- publisher, a book can have a publisher or not

PUBLISHER has :
- a name
- a address

+ **AUTHOR** has :
- name

Q1 Open file **UML-STUDENT.pttx** and create the class with relationship including the following requirements:

- Visibilities (public/private)
- Multiplicities (has 1, 2, many etc.)

Q2 From the **UML Diagram** implement the classes and they relationship in **Typescript**

Q3 Inside class **Book,** create the following methods:

```
addAuthor(author:Author)          // Add an author to the book
```

```
setPublisher(publisher:Publisher)    // Set publisher of book
```

Inside class **Library,** create the following methods:

```
addBook(book:Book)                      // Add a book to the library
```

Q4   In the Main.ts **:** create the following **instances**:

Author

name
Ronan
Him

Publisher

name	address
Sipar	Phnom Penh
IBC	Siem Reap

Book

Title	Year	Author	Publisher
OOP is the best	2018	ronan	sipar
Best team ?	2015	ronan	sipar
The Why not book	2020	him	

Library

Name	Address	Books
PNC LIBRARY	Phnom Penh	OOP is the best
		Best team?
		The Why not book

Q5   Create a method to get all books from a given author

```
getBooksFrom(author:Author) : Book[]
```

Q6   Create a method to get all books from a given publisher

```
getBooksFrom(publisher:Publisher) : Book[]
```