**OOP\_C# Exercises**

**Excercise 1**: Write a program to calculate the area, the perimeter of the rectangle.

* -  Write class **HinhChuNhat** includes:
  + - * 1. -**Attributes**: Length, width
        2. -Method:

+ Method setting (set), and getting (get) information for length and width.

* + - * 1. + Method of calculating area, perimeter.
        2. + The show() method consists of length, width, area, perimeter information.

-  Building the class contains the main function for the test section. Length, width can enter from the keyboard.

**Excercise 2**: We need to manage a football team, includes player and coach.

* Player: with the informations: code, name, address, shirt number , position, salary.
* Coach: with the informations: code, name, address, position,salary, years of experience.

1. Build the class as the above requirement (using the inheritance).

2. Build an interface, named IManger to design some methods:

* Input a list of players.
* Input a list of coaches.
* Show list of players.
* Show list of coaches.
* Update the information of players 🡪 void changePlayer(int playercode, int option, int shirtnumber, double salary);

+ If option = 0, change the shirt number of player

+ If option !=0, change the salary.

* Count the coaches that have years of experience >=3.
* Sum of the salary of the players that are the striker.
* Who have the max salary? 🡪 void showMaxLuong();
* Sort the list of players by ascending shirt number.
* Sort descending salaries of experienced coaches = 3.

3. Implement the above interface.

4. Build the main class:

* Show the menu.
* Depending on the option, call the processing functions in section 3 to display the results.

5. Build the class **CheckData**:

* Check null for String properties.
* Check number for number properties.
* Check…

**Excercise 3**: Develop an employee management program, specifically as follows:

1. Class **Employee**:

- Property: Code, name, date of birth, gender, number of children, salary

- Method:

+ Constructor: default and full parameters

+ Get/set method for properties

+ show() method

+ Calculate the income = salary + allowance.

In which, allowances are calculated as follows:

- If the number of children is not available, there is no allowance.

- If the number of children <= 2, the allowance = 1 million.

- The rest is allowance = 1.5 million.

2. Building an interface named **IManger** contains the following methods:

- Enter an employee list 🡪 ***void inputList(int size);***

- Show an employee list 🡪 ***void showList();***

- Count how many female employees are there without allowances. 🡪 ***int count();***

- Displays employees with child numbers <n, n entered from the keyboard

🡪 ***void showSocon(int n);***

- Sort the list of employees to increase salary according to the number of male employees. 🡪 ***void sortBySalary();***

- Remove from the list of male employees with the number of children> n, n entered from the keyboard. 🡪 ***void delete(int n);***

- Display the employees that name = given name from keyboard.

🡪 **void showByName(String name);**

- Update the salary of employees according to the following criteria: plus compared with the original salary

If the employee is <30 years old, add 5% of the salary.

If the employee has 30 <= age <40, add 10% of the salary

The remaining plus 15% of the salary

🡪 **void updateSalary();**

3. Xây dựng lớp có tên **Manger**:

- Property: ArrayList contains the employee list.

- Method:

+ Constructor has a parameter: ArrayList

+ Implement all methods of above interface.

4. Building a TestMain class to test the program, implementing the following requirements:

- Show the Menu:

**1. TC1- Enter the employee list**

**2. TC2- Display the employee list**

**3. TC3- Counts the number of female employees with no allowances**

**4. TC4- Displays employees with the number <given number.**

**5. TC5- Arranging an increasing number of male employees' salaries.**

**6. TC6- Removing male employees according to the number of entries entered from the keyboard.**

**7. TC7- Display employees by name.**

**8. TC8- Update salary**

**9. Exit.**

- Create a list of employees, including 5 employees who have information.

- Choose any option to display the corresponding results.

(*Use try ... catch to control data entry - Utilities*)

**Excercise 4**: In a school, there are 2 types of teachers:

* Fulltime teacher: salary = salary coefficient \* 2000000
* Parttime teacher: salary = slot \* 50000

Develop a payroll management and teacher information program.

Specifically:

1. Build the class **Teacher**: abstract class

* Property: code, name.
* Method:

+ Constructor: 2 types (no parameter and full parameters)

+ Get/set methods for all properties.

+ show() method.

+ Get salary – abstract method (only declare without body)

2. Build the class **FulltimeTeacher**: inherite from class Teacher

* Property: salary coefficient
* Method:

+ Constructor: 2 types (no parameter and full parameters)

+ Get/set methods for all properties.

+ Get salary = salary coefficient \* 2000000

3. Build the class **ParttimeTeacher**: inherite from class Teacher

* Property: slot
* Method:

+ Constructor: 2 types (no parameter and full parameters)

+ Get/set methods for all properties.

+ Get salary = slot \* 50000

4. Build an interface, named **IManager**, design the functions:

* Import and display a list of Teachers (including Fulltime teacher and PartTime teacher).
* Show list of teachers with the highest salary.
* How many parttime teachers have the slot > 10.
* Calculate the total number of parttime teacher's slots.
* Sort the list of teachers increased by salary.
* Is there a teacher whose name starts with ‘T’? If so, show that person's name and salary.
* Displays a list of fulltime teachers with the salary coefficient < 3, in which the full name will separate First and Last Names into 2 columns.

🡪 Build the class Manager implement the above interface.

5. Build the main class to excute the above functions. (Show menu and check input).