Object-Oriented Programming

Assignment 2

Teacher: Mansur Zhanbolat

Groups: SE-2214 - SE-2222

This assignment will help you to dive into the concepts of polymorphism and inheritance.

Create classes Person, Employee and Student.

Class Person

- id (int) – field (should be automatically set. **Hint**: use helper static variable)
- name (string) – field
- surname (string) - field
- toString() – method (e.g. "1. John Lennon")
- getters and setters (the field id needs only getter)
- 2 constructors (default and parametrized which takes both name and surname)
- string getPosition() – method (should return either the appropriate position or text "Student")

Class Employee

- id (int) – field (should be automatically set. **Hint**: use helper static variable)
- name (string) – field
- surname (string) – field
- position (string) – field
- salary (double) – field
- toString() – method (e.g. "Employee: 1. John Lennon")
- getters and setters (the field id needs only getter)
- 2 constructors (default and parametrized which takes 4 arguments name, surname, position, and salary)

Class Student

- id (int) – field (should be automatically set. **Hint**: use helper static variable)
- name (string) – field
- surname (string) – field
- gpa (double) – field
- toString() – method (e.g. "Student: 1. John Lennon")
- getters and setters (the field id needs only getter)
- 2 constructors (default and parametrized which takes 3 arguments name, surname, and gpa)

Interface Payable

- double getPaymentAmount() – will return the amount of money one could get in form of either salary or stipend (**Hint**: The stipend is given in case student is having gpa > 2.67 and it is constant: 36660.00 tenge)

Make your person to be Payable (Person should implement Payable interface)

Create Main class with **main()** and **printData(Iterable<Person>)** methods.

main() method should include:

- Several employees and students (including those who are not taking stipend because of gpa)
- ArrayList<Person> and with all employees and students inserted to current List.
- A call to printData method.

printData(Iterable<Person>) method:

Should iterate through every Payable object using foreach structure and print the data as following example output:

Student: 3. Ringo Starr earns 0.00 tenge

Employee: 1. John Lennon earns 27045.78 tenge

Student: 4. Paul McCartney earns 36660.00 tenge

Employee 2: George Harrison earns 50000.00 tenge

Last point: Try to make your Person to implement Comparable interface.

(**HINT**: *class Person implements Comparable<Person>*)

Implement compareTo() method for both Employee and Student classes. That comparison should be based on the amount of money they make.

In the main method call the Collections.sort() method and send your ArrayList there before the call to the **printData** method.

Hint: one of the Collections.sort() methods accepts List<T> where T is generic Comparable type. Since your Person is comparable, that should work.