Java SE Task, Week 6

Classes and Objects. Inheritance.

Task 1

Our aim is to calculate the total energy gained from all workers in factory. To solve this problem, we have to create our hierarchy classes of workers. From physics (natural science), you know that work done is equal to changed energy (**Work = ΔEnergy**). It means that, any worker will have a potential energy, which he will use for doing some work. In addition, any worker class will include own power. Power is measured by kilowatt, and from physics we remember that **Power = Work/time**.

Create a class **Worker** with parameters:

- String fullName;
- double energy; // in **kilojoules - KJ**
- double power; // in **kilowatts - KW**
- double efficiency; // efficiency coefficient, take from 0.1 to 0.4

Constructors (default and parameterized)
Getters and Setters

+ String toString(); // This method returns string value of a worker

+ double toWork(int time); // work done by worker in **kilojoules – KJ, time** is measured by **hours.** This method returns **work** done by worker in **time** hours. Each worker will have a potential energy. If potential energy is 0, then worker will not work. It means that the method will return 0. You will use an efficiency coefficient for measuring work by this formula (convert to algorithm):

```
Work = (Power * time) * efficiency;
Energy = Energy – Work;
```

Be careful, with creating an algorithm. I suggest you to use loops. In every hour (every iteration), you must calculate work done by worker in one hour, and decrease this value from energy. If energy will be 0, then worker will stop working and return work done by worker. Finally, the **toWork(int time)** method will return the total work done in **time** hours.

Create a class called **Programmer** that extends **Worker** class, which has parameters and methods:

- double moralMotivation; // Moral motivation coefficient of a programmer from 0.5 to 2.

Constructors (default and parameterized)
Getters and Setters
+ String toString(); // This method returns a String value of programmer

Override the method **double toWork(int time)**. For this method, you will use **moralMotivation** coefficient value. For every hour, you will just multiply the work done by a programmer to **moralMotivation** coefficient. That's all.

Create a class called **Teacher** that extends **Worker** class, which has parameters and methods:

- double tiredness; // Tiredness coefficient of a teacher from 0.01 to 0.1

Constructors (default and parameterized)
Getters and Setters
+ String toString(); // This method returns a String value of teacher

Override the method **double toWork(int time)**. For this method, you will use **tiredness** value. For every hour, you will just decrease efficiency by tiredness coefficient. It means, after every hour, teacher will lose efficiency coefficient by **tiredness** value.

In main class, create 5 programmers and 5 teachers with different parameters. Put them into one-dimensional array. Calculate:

a. The total work done by all workers in 5 hours.

b. The total work done by all workers in 10 hours.

c. Print total work done by each workers in 10 hours.

d. The worker data (It may be teacher or programmer) with maximum power

e. The worker data (It may be teacher or programmer) with maximum work done in 10 hours