Создайте пакет с именем by.gsu.epamlab и определите класс Material, который представляет собой однородный материал.

Поля класса:

• имя,

• плотность.

Конструкторы:

• конструктор no-arg;

• параметризованный конструктор.

Методы:

• геттеры;

• toString () - возвращает строковое представление однородного материала в формате csv: каждое поле, разделенное символом ";" символ.

Пример:

сталь; 7850,0

В том же пакете определите класс Subject, который представляет предмет, который состоит из однородного материала.

Поля класса:

•имя,

• материал,

• объем.

Конструкторы:

• конструктор no-arg;

• параметризованный конструктор.

Методы:

• геттеры / сеттеры;

• getMass () - возвращает массу объекта (= плотность \* объем);

• toString () - возвращает строковое представление объекта в формате csv: каждое поле и массу, разделенные символом ";" символ.

Пример:

проволоки; сталь; 7850,0; 0,03; 235,5

Определите класс Runner в пакете по умолчанию, где:

1. Создайте объект, который представляет собой стальную проволоку, имеющую объем 0,03 кубических метров.

2. Выведите содержимое объекта на консоль, используя метод toString ().

3. Обновите материал проволоки до меди (плотность = 8500,0) и выведите массу.

Пример:

Масса проволоки составляет 255,0 кг.

Create the package named by.gsu.epamlab and define a class called Material that represents uniform material.

**Class fields**:

* name,
* density.

**Constructors**:

* no-arg constructor;
* parameterized constructor.

**Methods**:

* getters;
* toString( ) – returns a string representation of uniform material in the csv–format: each field, separated by the ";" symbol.

Example:

steel;7850.0

In the same package define a class called Subject that represents a subject which consists of uniform material.

**Class fields**:

* name,
* material,
* volume.

**Constructors**:

* no-arg constructor;
* parameterized constructor.

**Methods**:

* getters/setters;
* getMass( ) – returns the subject mass (= density \* volume);
* toString( ) – returns a string representation of a subject in the csv–format: each field and the mass, separated by the ";" symbol.

Example:

wire;steel;7850.0;0.03;235.5

Define the Runner class in the **default** package, where:

1. Create the object which represents the steel wire having the volume 0.03 cubic meters.

2. Output the object content to the console, using toString( ) method.

3. Update the wire material to copper (density = 8500.0) and output its mass.

Example:

The wire mass is 255.0 kg.

#### Замечания и ограничения

– В этой задаче **нет** отношения **наследования** между сущностями (материал и предмет). Наследование возникает, когда одна сущность является частным случаем другой. Например, металл (или другое твердое вещество) и материал. Другими словами, металл является материалом. Предмет состоит из материала, а не является материалом. Поэтому предмет не может быть наследником от материала.

Такое отношение между сущностями называется композицией. Оно реализуется через ссылку на объект класса.

private Material material;

– Обратите внимание, что у конкретного материала плотность является константой, что нужно отразить при создании класса. Например, у стали плотность 7850.0 и никакая другая.

Еще являются константами объекты, соответствующие конкретным материалам (стали и меди).

– Не экономьте память под вещественные значения в java se. Вообще не используйте тип float, всюду только double.

– Этапы выполнения задачи 2

Этап 1. Реализация авторского решения.

Создать архив с решением **classes2.jar** и выложить в свою папку jars.

Этап 2. Анализ правильного решения.

После появления jar-файла скрипт через некоторое время откроет доступ к текстовому документу, где детально описан процесс реализации задачи.

Если считаете, что Ваш алгоритм лучше или имеете какие-либо замечания/вопросы, то выходите на устную консультацию.

Обратите внимание, что в последнем абзаце предлагается выполнить другую версию данной задачи.