Code Exercise

Create the package by.epam.lab for entity classes.

Define a class Purchase with two fields: item and quantity of items, which represents a purchase of some item. The quantity of purchased items can be both integer and fractional value. An item is considered to be any entity possessing specific or calculated price in BYN. The purchase cost is the rounded result of multiplying the item price by the quantity of items purchased.

Define three classes as representatives of an item:

1. Product with two fields: name and price;
2. DiscountProduct with the price discount field – the subclass of Product.
3. Service with three fields: name, total cost of service, number of service users. The price of a service per one user is the rounded-up result of dividing the total cost of a service by the number of service users.

Define a class PurchaseUtils with the only field - purchase and 5 methods:

- getPurchase() – getter;
- printPurchase() – outputs a purchase in the csv-format;
- printCost() – outputs a purchase cost in the format: cost = value BYN;
- printCostDiff(p) – outputs a cost difference for purchases this and p in the format: xxx diff = value BYN, where
  xxx is positive if the this cost is greater than the p cost,
  xxx is negative if the this cost is less than the p cost,
  xxx is empty string if the this cost is equal to the p cost,
- printIsSameCost(purchases) – outputs whether there is some purchase among purchases with the same cost like this purchase.

Define the Runner class in the default package, where:

1. Create p1 instance - the purchase of 20 packages of milk with the price 1.70 BYN.
2. Output p1 and its cost with the PurchaseUtils instance pu1.
3. Create p2 instance - the purchase of 12.5 kg of sugar with the price 3.00 BYN.
4. Output the cost of p2 and the cost difference of p2 and p1.
5. Create p3 instance - the purchase of 60 kg of sugar with the price 2.80 BYN and the price discount 0.10 BYN.
6. Without a Purchase instance create the PurchaseUtils instance pu4 for the gym workout for 2.25 months with the total cost 75.60 BYN and 5 participants.
7. Get an **item** instance from the last purchase with the PurchaseUtils instance pu4 and output it.
8. Output the cost of the last purchase with a PurchaseUtils instance.

9. Using a PurchaseUtils instance, output whether someone of p1, p3 or the last purchase has the same cost as the p2 purchase.

Замечания к задаче

– Это вспомогательная задача и в приведенной выше ее постановке параметризованные типы не требуются.
Поэтому реализуйте решение **без параметризованных типов**.
– Обратите внимание, в пункте 6 запрещено создавать экземпляр покупки.
А в пункте 7 требуется выполнить два действия (оператора):

1. объявить ссылку и инициализировать ее (оператор присваивания),
2. вывести сущность на консоль.