

Happy Customer

Vom realiza o „simulare” a unui magazin cu o promoție specială. În cadrul simulării vor exista următoarele feluri de obiecte.

ShoppingCart: data class (clasă ce conține doar date, nu comportament) ce conține un câmp întreg *value* aferent valorii coșului. Instanțele ei trebuie să fie imutabile (nu se pot modifica datele din interiorul său, fiind setate doar prin constructor).

Customer: caracterizat de nume, o sumă de bani de care clientul dispune respectiv o referință către un *ShoppingCart*. Clasa conține o metodă *boolean tryBuy()* care, dacă clientul are destui bani, scade din suma de bani de care clientul dispune valoarea coșului și returnează *true*, altfel, returnează *false* și nu face altceva în plus. Implementați de asemenea metoda *toString* pentru ca afișarea unei instanțe să fie human readable (într-o formă aleasă de voi).

SelfPayStation: reprezintă o casă a magazinului și este reprezentată de un identificator unic. Există o promoție prin care fiecare al 10-lea client ce reușește cu succes să achiziționeze ceva, primește un cadou, indiferent de ce casă folosește. Pe o casă se poate apela metoda *void use(Customer c)*, în care clientul încearcă să realizeze plata și se verifică dacă acest client primește sau nu un cadou.

Store: Contine PSVM (metoda *main*) a programului. În ea se vor crea instanțe ale claselor amintite mai sus la libera decizie a programatorului, dar scopul este de a reproduce efectul final.

Hint 1: Pentru id-ul stațiilor de plată, luați în calcul atât varianta cu contor static, dar să vă uitați și la [java.util.UUID](#).

Notă: Se vor afișa la ieșirea standard informațiile clienților ce primesc cadou, respectiv id-ul casei la care s-a realizat achiziția.