OOP Small Assignment 1

Description: An alchemist manages a collection of magical potions, each with its own unique properties. Every potion has a name, an effect type (such as healing, poison, or stamina boost), a potency level that determines its strength, a rarity classification (common, rare, or legendary), and an expiration status which indicates whether the potion is still usable or expired (potions are given usable or expired, you don't need to model them expiring). The alchemist needs to find and organize potions efficiently. Some of their tasks include searching for the strongest potion by identifying the one with the highest potency level. Alchemists also regularly check how many healing potions they have in stock, as these are commonly used in their craft. Additionally, the alchemist might want to find out which rare or legendary potions are available in their collection.

Beyond inventory management, the alchemist can brew new potions by combining multiple potions of a similar type to create a stronger version, where

- effect type is the effect type that occurs most often in ingredient potions
- potency level is the average of the potency of the ingredient potions added to the maximum potency level among them
 - rarity classification is legendary
- if one of the ingredient potions is expired then the new potion is also going to be expired

Occasionally, the alchemist must ensure that expired potions are removed from the inventory. Alchemists also keep track of how many potions they have used and created over time.

```
searchStrongestPotion():

if(collection.isEmpty())
return null
strongest := collection(0)
for potion in collection(1;1;
if(potion.getPotencyLevel() > strongest.getPotencyLevel()
return strongest.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       countHealingPotions():
cnt := 0
for potion in collection:
if potion.getType() = "Healing"
cnt++
return cnt
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       findRareOrLegendary():

newList := new Potion[]

for potion in collection:

(flpoting neRarityClassification = "Rare" or

potion getRarityClassification() = "Legendary")

newList add(potion)

return newList
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    brewNewPotion(ingredients):

if ingredients is empty:

throw error (illegalArgumentException)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              counter := new empty map
for each potion in ingredients:
if potion type is not in counter:
add potion type to counter with count 1
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      else:
increase potion type count by 1
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              maxEntry := null
for each entry in counter:
if maxEntry is null or entry value > maxEntry value;
maxEntry := entry
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              effType := maxEntry key
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          maxPotLevel := first potion's potency level 
sum := 0 
for each potion in ingredients: 
sum += potion's potency level 
if potion's potency level > maxPotLevel: 
maxPotLevel := potion's potency level
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            avg := sum / number of ingredients
potLevel := maxPotLevel + avg
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              expired := false
for each potion in ingredients:
if potion is expired:
expired := true
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              rarity := "Legendary"
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              return new Potion(name, effType, potLevel, rarity, expired)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    removeExpireds():
unExpired := new empty list
for each potion in collection:
if not potion getExpired():
unExpired.add(potion)
                                                                                                     Alchemist
              - collection: Potion[]
         - getCollection(): Potion[]
- searchStrongestPotion(): Potion
- countHealingPotions(): Int
- IndRareOrt.opendary: Potion[]
- brewNewPotion[Potion[] ingredie
- fryee, pott.evgl, rarity, expired)
- removeExpireds(): Potion[]
- cntlinventory(): Int
- addPotion(Potion p): Potion[]
                                                                                                                                                                                                                                                                           .....
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          cntinventory():
return countBrews
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              addPotion(Potion p):
collection.add(p)
                                                                                                                                                                                                                                                                                                                                                this.name := name
this.effectType := effectType
this.potencyLevel := potencyLevel
this.rartlyClassification := raritlyClassification
this.expired := expired
      - name: String

- effectType: String

- potencyLevel: double

- rarityClassification: String

- expired: boolean
                                                                                                                                                                                                                                                                                                                                                     return this name

- expred. bouwarn
- Potion(name: String, effectType: String, potencyLevel: String, rarrlyClassification String, expired. booleans)
- geName(): String;
- seNsme(): St
                                                                                                                                                                                                                                                                                                                                                     return this.effectType
                                                                                                                                                                                                                                                                                                                                                     this.effectType := effectType
                                                                                                                                                                                                                                                                                                                                                     return this potencyLevel
                                                                                                                                                                                                                                                                                                                                                     this.potencyLevel := potencyLevel
                                                                                                                                                                                                                                                                                                                                                     return this.rarityClassification
                                                                                                                                                                                                                                                                                                                                                    this.rarityClassification := rarityClassification
   void
+ getExpired(): boolean
+ setExpired(expired: boolean): void
                                                                                                                                                                                                                                                                                                                                                    return this.expired
                                                                                                                                                                                                                                                                                                                                                    this.expired := expired
```

getCollection(): return this.collection