

# Поправителен изпит по ООП

18.07.2018

## Художествена галерия

При изпълнение на задачата използвайте базовият проект, който се намира в `exams/july`.

В тази задача трябва да се имплементират класовете за скулптури и галерии.

Основните класове са:

- `org.elsys.gallery.Sculpture` - клас за скулптура, която се характеризира с автор, заглавие, година, в която е създадена скулптурата, тегло на скулптурата и цена.
- `org.elsys.gallery.Gallery` - клас за художествена галерия. Всяка галерия има месечен бюджет който харчи за поддръжка на дейността си и за закупуване на нови произведения на изкуството.
- `org.elsys.gallery.ModernGallery` - клас за художествена галерия за модерно изкуство. В такава галерия се купуват и излагат само произведения на изкуството, които са създадени след 1800 година.

## Оценяване

- 60т - Среден 3
- 80т - Добър 4
- 100т - Мн. добър 5
- 120т - Отличен 6

## Задачи

### 1. Sculpture, Gallery и ModernGallery

Имплементирайте методите на трите класа. За да сте сигурни, че методите са имплементирани коректно, трябва да изпълните JUnit тестовете, които се намират в папката `tests`.

## 1.1 Sculpture (10т)

1. (5т) Трябва да има конструктор с 5 аргумента - автор, заглавие на творбата, година в която е създадена, тегло и цена. Класът `Sculpture` трябва да имплементира интерфейса `PieceOfArt`.
2. (5т) Конструкторът трябва да проверява параметрите за коректност и ако стойностите на параметрите не са адекватни, трябва да генерира `RuntimeException`. Изискванията към параметрите са:
  - a. годината трябва да бъде число между -1000 и 2018;
  - b. теглото трябва да бъде положително;
  - c. цената трябва бъде положително число.

## 1.2 Gallery (25т)

За имплементиране на `Gallery` използвайте подходяща колекция от стандартната библиотека. За всеки метод в кода е добавено описание в `JavaDoc` с изискванията към него. Обърнете внимание, че класът `Gallery` трябва да работи като използва интерфейса `PieceOfArt`.

1. (5т) Месечните разходи на галерията за застраховки, охрана, климатизация и т.н. се изчисляват като процент от общата цена на картините в колекцията на галерията. За класа `Gallery` месечните разходи са 10% от общата цена на картините в галерията.
2. (5т) Методите `add` и `addAll` са предназначени за добавяне на произведения на изкуството към колекцията на галерията. Ако към колекцията се добавя произведение, чието име, автор и година на изработване съвпада с друго, вече добавено към колекцията произведение, методите трябва да генерират `RuntimeException`.
3. (5т) Методът `monthlyPurchase` е предназначен за закупуване на нови картини и трябва да провери наличието на достатъчно средства за закупуване. Ако наличният бюджет не е достатъчен за закупуване на всички картини, методът трябва да генерира `RuntimeException`.
4. (5т) Методът `getWorksByWeight` връща колекция от творби, чието тегло е между `minWeight` и `maxWeight`.
5. (5т) Колекцията, върната от `getWorksByWeight` трябва да бъде подредена според теглото на картините от най-леката към най-тежката (използвайте вграденият метод `sort` в колекцията, която използвате).

## 1.3 ModernGallery (30т)

В галерията за модерно изкуство се купуват само творби, създадени след 1800 година. Изискванията за климатизация на такива творби са занижени, поради което месечните разходи са по-ниски - 8% от общата цена на творбите в колекцията.

1. (5т) Направете `ModernGallery` наследник на `Gallery`

2. (5т) Дефинирайте абстрактен клас `AbstractGallery` и направете `Gallery` и `ModernGallery` наследници на абстрактния клас.
3. (5т) Методите `add` и `monthlyPurchase` трябва да генерират изключение `RuntimeException` ако годината на създаване на творбата е преди 1800 година.
4. (5т) Дефинирайте метод `getWorksOf`, който връща списък от всички творби на даден автор от колекцията на галерията.
5. (5т) Списъкът от творби върнат от `getWorksOf` трябва да бъде сортирани от най-скъпата към най-евтината творба на даденият автор.
6. (5т) Напишете JUnit тест за този метод

## 2. main (19т)

Дефинирайте `main` метод в класа `org.elsys.gallery.MainClass`, който прави следното:

1. (5т) прочита ред от стандартния вход, който съдържа репрезентация на множество скулптури. Картините са разделени с точка и запетая и интервал, а автора, заглавието, годината, теглото и цената на скулптурата са разделени със запетайка и спейс. Например: "Artist1, Title1, 1920, 20, 1000; Artist1, Title2, 1921, 2, 1100, Artist1, Title3, 1901, 500, 910"
2. Създайте списък (`java.util.List`), в който да добавяте всички прочетени скулптури.
3. (3т) Ако годината на създаване на скулптурата е преди 1000 година, я пропуснете
4. (3т) Всички картини се добавят в инстанция на `ModernGallery` с месечен бюджет 5000 чрез метода `monthlyPurchase`
5. (3т) Взема всички произведения на скулптура "Artist1" от галерията чрез метода `getWorksOf`
6. (5т) Оградете закупуването на картини в галерията в `try catch` блок, който обработва изключение от типа `RuntimeException`

## 3. Поддръжка за картини - йерархия от творби на изкуството (32т)

1. (8т) Добавете клас `Painting`, който представя картини в галериите. Класът `Painting` трябва да имплементира интерфейса `PieceOfArt`. Освен атрибутите на `PieceOfArt`, класът `Painting` трябва да има допълнителни атрибути за размерите на картината (ширина и височина) и техника на изпълнение (масло, акварел, темпера, графика, и т.н.).
2. (8т) Добавете абстрактен клас `AbstractArtWork`, в който да бъдат изведени общите части на класовете `Sculpture` и `Painting`.
3. (8т) Добавете методи в `Gallery` които да връщат творби от галерията, които са картини, и такъв който връща творби, които са скулптури.
4. (8т) Добавете методи, които могат да филтрират само картините от колекцията на галерията според техниката на изпълнение и според размерите им.

## 4. Използване на Java Streams API (32т)

1. (8т) Имплементирайте всички методи за филтриране на творбите на изкуството в класовете Gallery и ModernGallery като използвате Java Streams API.
2. (4т) Добавете метод, който изчислява средната цена на творбите в колекцията `getAverageWorkPrice()`.
3. (4т) Добавете метод, който изчислява средната цена на картините в колекцията `getAveragePaintingPrice()`.
4. (8т) Добавете метод, който изчислява средната цена на картините в колекцията с година на създаване след дадена година `getAveragePaintingPriceAfterYear(int year)`.
5. (8т) Добавете метод, който изчислява средната цена на скулптурите с тегло над даденото `getAverageSculpturePriceAboveWeight(double weight)`