Programmering 2: övningsuppgift 1

Version: 1.4
Uppdaterad: 2021-03-25, se dokumenthistoriken sist i dokumentet för info.
Författare: Patrick Wentzel

Detta är den första övningsuppgiften på kursen PROG2 VT2021.

Uppgiften är giltig under kursomgången fram till uppsamlingstillfället i augusti 2021 och upphör därefter att gälla och kan komma att ersättas kursomgången 2022.

Innehållsförteckning

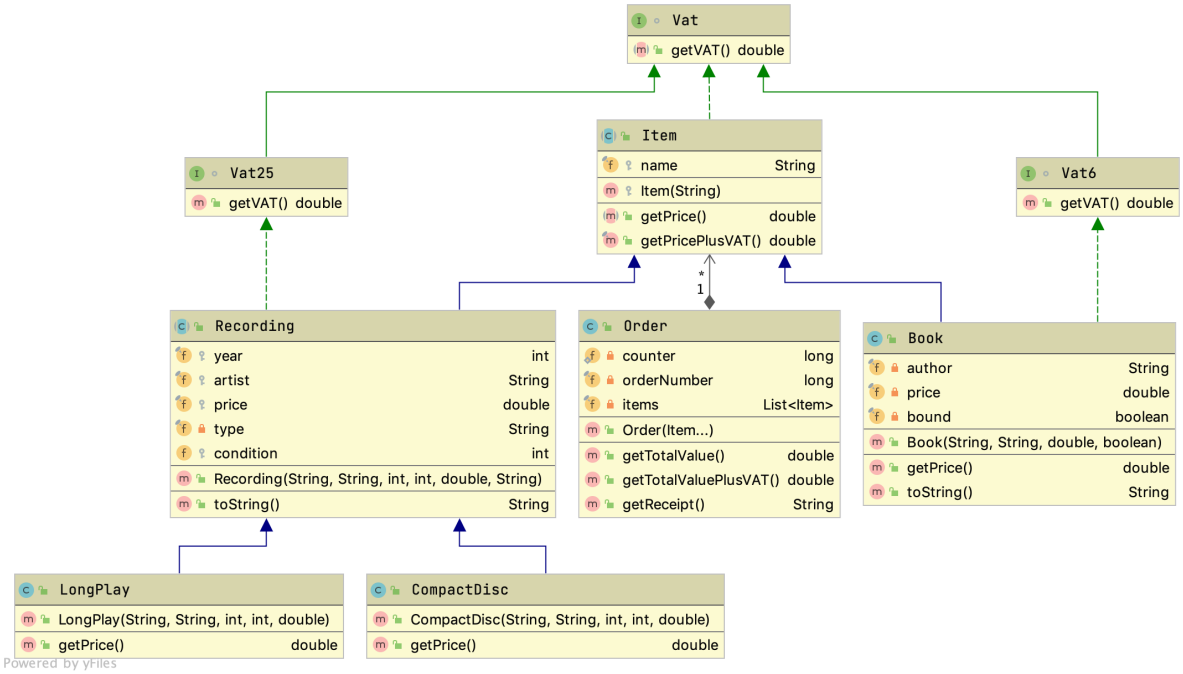
Inledning

Efter åratal av studier har du äntligen fått ditt första uppdrag efter examen, en praktikplats på företaget Music Mania som driver butiken Johnny Rockers Music Emporium (döpt efter ägaren Johnny Rocker).

Företaget importerar och säljer diverse musikrelaterade produkter, men har aldrig satsat på IT så det kommer finnas mycket att göra, och det kommer vara din uppgift att realisera olika förslag till förbättringar av företagets processer som Johnnys konsultkompis Patrick har tagit fram. (Patricks idéer är inte alltid så genomtänkta men det är de som gäller).

I ett första steg ska det skapas ett system där man kan registrera olika artiklar som säljs i butiken, och sedan skapa ordrar och skriva ut snygga kvitton till kunderna.

Som ett första proof-of-concept behöver du skriva ett antal klasser och gränssnitt som motsvarar följande klassdiagram:



Uppgiften

Uppgiften är alltså att realisera ovanstående klassdiagram i kod på ett sätt som uppfyller de olika kraven som specificeras i beskrivningen av klasserna och gränssnitten.

Som en hjälp finns det i iLearn ett enkelt testprogram (`Exercise1.java`) som visar hur objekt kan skapas och skrivas ut.

Inlämningsdatum (deadline)

Rekommenderat inlämningsdatum är den 31/3. Uppgiften behöver vara avklarad innan slutuppgiften med deadline 5/6 kan lämnas in.

Nyckelord

abstract, class, default, extends, implements, interface, @override

Klasserna

I projektets första fas kommer systemet enbart att hantera ett litet urval av produkter, och informationen om respektive produkt är ganska liten och ogenomtänkt (det har inte gjorts en ordentlig domänmodell).

Item

Klassen `Item` är supertyp till övriga klasser[1] och ska innehålla följande metoder:

- `getPrice` – en abstrakt metod som returnerar priset. Måste implementeras av ärvande konkreta klasser.
- `getPricePlusVAT` – en konkret metod som returnerar priset efter att moms (mervärdesskatt) har lagts på. Metoden ska inte kunna överskuggas.

Klassen ska implementera gränssnittet Vat eftersom den representerar en kategori av momspliktiga varor. Klassen ska inte kunna instansieras.

Alla saker som är `Item` ska ha ett namn (attributet name).

Konstruktor:

```
Item(String name)
```

Book

Klassen `Book` är en konkret subtyp till `Item` och representerar en bok som kan vara antingen inbunden eller inte (attributet bound). Böcker är belagda med 6% moms och ska därför implementera gränssnittet Vat6.

Priset på böcker påverkas av om boken är inbunden eller inte. Inbundna böcker är värda 25% mer.

Böcker har utöver namn också en författare (attributet author).

Konstruktor:

```
Book(String name, String author, double price, boolean bound)
```

[1] men inte till gränssnitten som utgör en egen hierarki

Recording

Klassen `Recording` är supertyp för de konkreta skivklasserna. Klassen innehåller inga särskilda metoder, men ska implementera gränssnittet Vat25 eftersom skivor säljs med 25% moms. Klassen ska inte kunna instansieras.

Priset på olika typer av skivor minskar med slitage (attributet condition) som är ett värde mellan 0 och 10, där 10 är perfekt skick och 0 representerar något osäljbart. Värdet minskar med 10 procentenheter för varje steg, så något med startpriset 100 och slitaget 8 har värdet 80.

Inga skivor kan dock få ett värde under 10 kronor före moms.

Se klassdiagrammet för detaljer om vilka attribut som förväntas.

Konstruktor:

```
Recording(String name, String artist, int year, int condition, double price, String type)
```

CompactDisc

Klassen `CompactDisc` är en konkret subtyp `Recording` och representerar en typ av inspelning.

Priset på CD beräknas på standardviset (med slitage).

Konstruktor:

```
CompactDisc(String name, String artist, int year, int condition, double price)
```

LongPlay

Klassen `LongPlay` är en konkret subtyp `Recording` och representerar en typ av inspelning som har blivit populär igen.

Priset på `LongPlay` beräknas på standardviset (med slitage), men en egenhet är att skivans värde också ökar över tid, med 5 kronor per år, vilket innebär att en skiva från förra året skulle vara värd 5 kronor extra. En LP från 2020 med priset 90 kr slitage (condition) 1 får priset 14 kronor innan moms (90.0 * 0.1 + (2021-2020)*5.0).

Konstruktor:

```
LongPlay(String name, String artist, int year, int condition, double price)
```

Order

Klassen `Order` är en klass som representerar ett inköp. Klassen ingår inte i någon arvshierarki och implementerar inte några gränssnitt.

Klassen ska ha följande metoder:

- En konstruktor som accepterar ett valfritt antal `Item` som en variabel argumentlista (`Item...`) och lägger till de i en lämplig samling
- `getTotalValue` returnerar totalpriset för ordern innan momspåslag
- `getTotalValuePlusVAT` returnerar totalpriset för ordern efter momspåslag
- `getReceipt` returnerar en sträng med information om orderns innehåll och totalpris (med och utan moms) i formatet[2] enligt exemplen nedan.

Exempel 1:

```
Receipt for order #3
-----------
* Recording { name=Punisher, artist='Phoebe Bridgers', year=2020, type=CD, condition=10, original
price=200.0, price=200.0, price+vat=250.0 }
* Recording { name=What Kinda Music, artist='Tom Misch', year=2020, type=LP, condition=10, original
price=150.0, price=155.0, price+vat=193.75 }
* Recording { name=Little Oblivions, artist='Julien Baker', year=2021, type=LP, condition=10, original
price=120.0, price=120.0, price+vat=150.0 }

Total excl. VAT: 475.0
Total incl. VAT: 593.75
-----------
```

Exempel 2:

```
Receipt for order #2
-----------
Book { name='Beethoven: a biography', author='Holmqvist', bound=false, price=400.0, price+vat=424.0 }
Recording { name=Giant Steps, artist='John Coltrane', year=1959, type=LP, condition=10, original
price=100.0, price=410.0, price+vat=512.5 }
Recording { name=Kind of Blue, artist='Miles Davis', year=1959, type=CD, condition=5, original price=100.0,
price=50.0, price+vat=62.5 }
Total excl. VAT: 860.0
Total incl. VAT: 999.0
-----------
```

[2] fast utan radbrytningarna i raderna med Recording som inte fick plats…

Gränssnitten

En poäng med övningen är att visa att även gränssnitt (interface) kan ingå i arvshierarkier och att de kan erbjuda standardimplementationer av de metoder som definitionen kräver via så kallade default-metoder.

Vat

Gränssnittet Vat (eng. Value Added Tax) representerar att något är momspliktigt[3] och innehåller metoden getVat som returnerar momssatsen i de fall den är implementerad.

Vat6

Gränssnittet Vat6 används för varor som är belagda med 6% moms, t.ex. böcker. Gränssnittet ska ärva från **Vat** och implementera metoden getVat som en default-metod som returnerar värdet 0.06.

Vat25

Gränssnittet Vat25 används för varor som är belagda med 25% moms, t.ex. skivor. Gränssnittet ska ärva från **Vat** och implementera metoden getVat som en default-metod som returnerar värdet 0.25.

.

[3] Moms, eller mervärdesskatt (tidigare omsättningsskatt oms), är en skatt som läggs på priset vid försäljning av varor och tjänster (förutom för momsbefriade saker).

Instruktion för inlämning

För att kunna lämna in behöver man ha valt grupp för inlämningen.

I VPL för övningsuppgiften laddas klasserna och gränssnitten upp i separata filer (6 klasser och 3 gränssnitt). Testprogrammet ska inte lämnas in.

Koden testas automatiskt med hjälp och kommer att bli godkänd med automatik när alla tester uppfylls.

Överst i samtliga inlämnade filer ska det finnas en kommentar med namn och användarnamn för samtliga gruppmedlemmar. Exempelvis:

```
// PROG2 VT2021, Inlämningsuppgift, del 1
// Grupp 004
// Dewey Duck dedu1111
// Huey Duck hudu1234
```

Dokumenthistorik

Version	Datum	Ändringar
1.0	2021-03-23	Första version.
1.1	2021-03-23	La till information om klassen Book som hade fallit bort.
1.2	2021-03-24	La till information om konstruktorer och förtydligade exempel i klassen Order.
1.3	2021-03-24	Uppdaterade klassdiagrammet.
1.4	2021-03-25	Förtydligade hur värde beräknas för Recording och subtyper.