Kassasystem

Denna uppgift är lämplig för grupper på tre till fem personer, den är betydligt renare än den föregående, men kan upplevas som lite tråkigare eftersom den är betydligt mer styrd av hur denna typ av system fungerar i verkligheten.

Uppgiften går ut på att implementera klasser som skulle kunna användas i ett kassasystem med varor, kvitton, rabatter, kunder, etc. Till skillnad från rougelike-förslaget kan det här eventuellt vara intressant att utveckla en liten del av ett användarinterface, eller åtminstone något som kopplar ihop de olika delarna, dock absolut inte hela kassasystemet.

Förslag på uppdelning

Nedanstående lista innehåller förslag på funktioner som kan implementeras. Några av dessa, som pengar och varor är absolut nödvändiga, och är lämpliga att utveckla tillsammans för att komma igång i projektet. Därefter kan varje person välja ett antal funktioner och gruppen jobba mer distribuerat. Det går inte att säga exakt hur många funktioner man ska ta per person eftersom detta beror på deras komplexitet. Tänkt dock på att varje person måste ha tillräckligt mycket intressant funktionalitet, och också möjlighet att interagera med andra deltagares kod. Det är också viktigt att arbeta objektorienterat. Rabatter till exempel ska inte räknas ut av kvittoklassen, även om den säkert ansvarar för att samla ihop dem.

Pengar

I princip det som gjordes när vi gick igenom TDD. En klass som representerar pengar.

Valutor

En möjlig utökning av ovanstående så att systemet stödjer flera valutor. Kan ge intressanta effekter om man tänker på växelkurser, och vad som ska hända om kunden inte betalar med ett exakt belopp. I vilken valuta ska kunden då få växel?

Varor

Som ett minimum ett namn och ett pris. Antagligen också förmågan att presentera sig själv på något lämpligt sätt.

Varugrupper

Samlingar av varor med något gemensamt, ofta använt i samband med rabatter, eller för att generera statistik. Man bör kunna lägga till och ta bort varor ur gruppen, samt iterera över den[1].

Moms-satser

Olika varutyper har olika momssats som behöver räknas ut och redovisas.

Pant

Vissa varor, till exempel många dryckesförpackningar, har en separat pant som läggs på utöver priset, och som man (rimligen) inte betalar moms eller får rabatt på.

Åldersgränser

Vissa varor, som alkohol och tobak, får bara köpas av personer över en viss ålder.

Leverantörer

Egentligen samma sak som varugrupper, men kopplade till en viss leverantör.

Kvitton

En samling med varor som automatiskt kan räkna ut pris. Funktioner man kan tänka sig är lägga till och ta bort varor, gruppera varor, sortera dem i bokstavsordning, samt naturligtvis skriva ut kvittot. Tänk på att samma vara kan läggas till och tas bort vid olika tillfällen.

---

[1] https://docs.oracle.com/en/java/javase/16/docs/api/java.base/java/lang/Iterable.html

Java har ett ganska enkelt interface för att skriva ut på en skrivare[2] som kan implementeras, och med fördel kan testas med hjälp av, till exempel, mock-objekt. Kvitton har normalt också datum och tid som kan testas med hjälp av test doubles.

Många av de andra funktionerna kan påverka kvittot, till exempel personen som sitter i kassan, kundens eventuella medlemskap, momssatser, etc.

Kassan

En driver för många av de andra funktionerna. Exakt vad denna ska innehålla beror på vad man väljer att implementera, och det finns inga krav på att man överhuvudtaget gör denna.

Växel

Om kunden betalar kontakt ska systemet lämna växel med minsta antal sedlar och mynt. Idag hanteras detta ofta av automtiska system, vad händer med ett sådant om det inte finns tillräckligt mycket av en valör? I vissa fall kan man ge fler av en lägre valör, i andra inte.

EAN/GTIN[3]

En naturlig utökning av varuklassen skulle vara en klass som representerar streckkodens GTIN (Global Trade Item Number). Validering av kontrollsiffran är en funktion här.

Generering av den associerade streckkoden skulle kunna vara en intressant utökning, och kan göras antingen till en bild eller till en skrivare.

Scanning

Scanning av varorna är det vanliga sättet att registrera dem idag. Ni har inte tillgång till en scanner, så att försöka lägga till en sådan funktionalitet skulle kunna vara ett intressant exempel på hur man testar interaktion med något som ännu inte är implementerat. (Interface och test doubles, mock-objekt är ett alternativ även här.)

Tänk på att scanning inte alltid lyckas.

Våg

En variant av ovanstående är att kunna väga varor. Detta skulle också få in en ny typ av varor där priset är per viktenhet istället för per styck.

Enkla rabatter

Enkla rabatter är sådana som ger ett visst belopp eller en viss procentsats i rabatt. De kan vara kopplade till hela köpet, en viss vara eller en varugrupp. Nästan alltid har de en giltighetstid som kan testas med hjälp av dependency injection.

Ett designmönster som kan vara intressant här är Decorator[4].

Mer komplicerade rabatter

Mer komplicerade rabatter är till exempel tag 3 betala för 2, rabatter som bara gäller max 2 köp per kund, rabatter som kräver att kunden handlat över ett visst belopp, eller att det är en speciell tid på dagen och bara gäller vissa kunder så som pensionärsrabatter dagtid vissa dagar, eller happy hour.

Bara en rabatt

Oavsett vilka rabatter man har brukar det i de flesta fall vara så att bara en rabatt gäller, och då är det viktigt att det är den som ger högst rabatt. I vissa fall kan detta bli ganska komplicerat, speciellt om det finns vissa varor som ingår i flera rabatterbjudanden, eller om det finns rabatter som gäller hela kvittot samtidigt som det finns rabatter på vissa varor.

Lagersaldo

En naturlig utökning av ett kassasystem är att också hålla reda på lagersaldot för varorna i

[2] https://docs.oracle.com/javase/tutorial/2d/printing/index.html

[3] https://sv.wikipedia.org/wiki/European_Article_Number

[4] https://en.wikipedia.org/wiki/Decorator_pattern

butiken, och automatiskt uppdatera detta när någonting köps.

Kunder

Namn, adress, personnummer, telefonnummer, e-postadress. Ju mer information vi har om kunderna desto bättre. Självklart vill vi också hålla reda på vad varje kund handlar så att vi kan generera statistik och riktad reklam.

Ett tips är dock att inte försöka få in all information i klassen som ni kan tänka på. Det är lätt att det blir för stort fokus på kvantitet istället för kvalitet. Välj ut delar som blir intressanta ur testsynvinkel, eller som är nödvändiga för andra funktioner.

Medlemskap

Många affärskedjor använder sig av någon typ av medlemskap för sina kunder. Dessa ger ofta någon typ av fördel för kunden, samtidigt som de tillåter affären att samla in betydligt mer information om kunden.

Medlemskap kan också vara tidsbegränsade, kosta pengar, ha en minimiålder, etc.

Återbäring

En fördel som kan finnas är direkt återbäring, där medlemmen får tillbaka en del av köpesumman. Detta kan ske när kunden handlat för tillräckligt mycket inom en viss tid, eller med viss periodicitet.

Ibland finns det också olika nivåer på återbäring beroende på hur mycket man har handlat för eller vilken typ av medlemskap man har.

Vissa kategorier av varor är ibland uteslutna ur återbäringsprogram, till exempel lotter och presentkort.

Poäng

En variant på återbäring är att man samlar poäng som sen kan bytas in mot varor eller presentkort.

En typ av erbjudande som ibland förekommer är att medlemmar får dubbla poäng under en viss period, eller på vissa varor eller kategorier.

Bonuscheckar

En sak som ofta kan köpas för poäng är bonuscheckar som sen kan användas för att helt eller delvis betala ett köp. Dessa räknas normalt inte som rabatter, så man kan betala ett köp med flera bonuscheckar, och eventuella rabatter ska appliceras. En möjlig begränsning är att man inte får ut något av värdet för en bonuscheck i kontanter, så att de alltså inte ger någon växel om man betalar för mycket.

Presentkort

En annan form av alternativt betalningsmedel är presentkort. Dessa är varor, men räknas aldrig med i rabatter eller bonussystem, och måste ibland aktiveras för att vara giltiga. De har normalt en tidsgräns för hur länge de får användas.

Riktad reklam

En viktig anledning till medlemskap är möjligheten att generera riktad reklam som är anpassad efter vilka varor eller varukategorier som kunden brukar köpa, ålder, kön, adress, etc.

Personal

Kassasystem har normalt information om vem som satt i kassan vid en viss tidpunkt. Detta är absolut relevant, men behöver kompletteras med funktionalitet för att bli riktigt intressant i detta sammanhang. Annars blir det bara en klass till och ett fält i kvitto-klassen.

En möjlighet skulle vara att hålla reda på hur många varor varje medlem av kassapersonalen kan hantera per tidsenhet. En ofta kritiserad funktion i denna typ av system.

En annan skulle vara att införa någon typ av bonussystem som är beroende av hur mycket personen har sålt under månaden eller året.

Återköp

Ibland är inte kunder nöjda med ett köp utan vill göra ett återköp av hela eller delar av det. Detta påverkar många av de tidigare uppräknade funktionerna, till exempel lagerhållning, återbäring och poäng. Om varan ingått i en

grupp av varor som kunden fått rabatt för måste detta också tas hänsyn till.

Uppdelat köp

Det finns många sätt att betala på: kontanter, kort, Swish, poäng, bonuscheckar, present-kort, etc. Speciellt med de sistnämnda är det vanligt att dessa bara täcker delar av ett köp, och att kunden vill använda andra betalnings-medel för resten.

Parkera köp

Ett köp kan parkeras om kunden glömt något och vill gå tillbaka in i butiken eller hem och hämta plånboken. Detta är bara relevant om man faktiskt bygger själva kassan.

Loggning

Väldigt mycket av vad som sker i ett system av denna typ loggas, antingen för att det finns lagkrav på det, eller för att kunna få ut information om hur systemet används. Detta kan göras i databaser eller filer, som kan testas med hjälp av test doubles.

Rapporter

All den information som finns i systemet ger möjlighet att generera rapporter till olika delar av verksamheten. Dessa kan till exempel vara i html eller skrivas ut på skrivare. Alternativt kan man testa att generera diagram i html eller bildformat.