

Inlämningsuppgift 3 – Små Stålers AB (del 3 av 4)

I tidigare inlämningsuppgifter har du skapat ett system åt Små Stålers AB där kunder kan göra uttag och insättningar med mera. Allt eftersom både systemet och kundunderlaget nu växer har det visat sig att fler och fler fel uppstår i systemet. Systemet kan krascha vid felinmatningar och tillåter att kunder med felaktiga personnummer skapas. Små Stålers AB har fått systemet analyserat och följande är vad som nu måste åtgärdas av dig:

- Personnummer

Det behövs en felkontroll av personnummer. Ett personnummer för en kund ska alltid bestå av 10 siffror (det ska inte finnas med något bindestreck). Skapa en undantagsklass (klass som ärver Exception) som ska kastas om felaktigt personnummer ges till kundklassen. Ange lämpligt felmeddelande som du kan förmedla till användaren.

Förutom att validera att det är 10 tecken ska du även validera att personnumrets (ååmmddnnnn) månader och dagar är giltiga tal. Det vill säga att mm måste ligga mellan 1-12 och dagar mellan 1-31. Du behöver inte ta hänsyn till skottår eller att månader har olika antal dagar. Kasta samma undantag som tidigare koll med personnummer, men kasta med ett annorlunda felmeddelande än ovan. Meddela användaren om problemet.

Förutom ogiltiga tecken kan du försöka att konvertera personnumret till ett heltal och därigenom avgöra om siffror har matats in eller inte. Här bör kanske NumberFormatException kastas om personnumret inte är enbart siffror.

- Postnummer

Det behövs även en felkontroll av postnummer. Ett postnummer för en kund ska alltid bestå av 5 siffror. Skapa en undantagsklass som ska kastas om felaktigt postnummer ges till kundklassen. Ange lämpligt felmeddelande som du kan förmedla till användaren.

Skapa en testklass som på lämpligt sätt visar att de nya funktionerna du implementerat fungerar som de ska. Det kan till exempel ske genom att skapa en kund med korrekta uppgifter samt en kund med icke korrekta uppgifter för personnummer och/eller postnummer. På den korrekt skapade kunden kan du sen prova att sätta ett felaktigt personnummer och/eller postnummer. Testklassen ska hela tiden meddela användaren vad som sker (om kund kunde skapas eller inte och om personnummer/postnummer kunde ändras eller inte och vad som i så fall är fel).

Viktigt! Inga in- eller utmatningar från/till kommandofönstret får ske i någon annan klass än "testklassen". Använd i stället lämpliga returtyper i dina övriga klassers metoder.

Tänk även på att det inte ska vara din testklass som ska kasta undantagen utan det är din Kund-klass. I testklassen ska du fånga upp eventuella undantag som kastas och vidta lämpliga åtgärder. Det vill säga att skapa en kund eller modifiera en befintlig på till exempel följande sätt ska inte vara möjligt:

```
try {  
    Customer c = new Customer("800101-0101", "Test", "Kund");  
    // Undantag kastas då personnummer innehåller tecknet - (samt har fler än 10  
tecken)  
}  
catch (CustomerExcpetion e) {  
    System.out.println("Kunde inte skapa kund: " + e.getMessage());  
}
```

eller:

```
try {  
    Customer c = bank.getCustomer(1);  
    c.setPostalNumber("123456");  
    // Undantag kastas då postnummer innehåller 6 siffror  
}  
catch (CustomerException e) {  
    System.out.println("Kunde inte ändra postnummer: " + e.getMessage());  
}
```