

## Dokumentation av SubjektAssignment

### Vad gör programmet?

Programmet är en butik som säljer blacksmith produkter. Det börjar med att kontrollera om det finns ett namn till butiken i BlacksmithData.txt. Om namn ej finnes, ombes användaren att skriva in ett namn till smedjan som sedan sparas i BlacksmithData.txt. Därefter gör den fler kontroller om det finns filer av kunder, anställda och produkter och läggs in i respektive listor om det finns.

När kontrollen med filerna är klart, kommer användaren till huvudmenyn där man får tre alternativ, om man är en kund, anställd eller chef. Anställd och chef är låst bakom en ID kontroll som den anställda respektive chefen får ut när de skapas.

När du kör kundmenyn kommer den att fråga om du är ny kund och då får du alternativet att mata in ett namn. Därefter läggs du in i kundlistan och en unik id returnas till dej när du ska sedan gå tillbaka till ditt konto. I kundmenynalternativet kommer du till en annan meny där du får alternativ att se alla produkter och lägga in till din "shoppingcart", se din "shoppingcart" eller logga ut.

För att köra employee-menyalternativet krävs det att det finns några anställda som chefen har skapat i ens menyalternativ. Att den anställda finns kontrolleras med en unik ID som den anställda blir tilldelas när den "skapas" eller anställs av chefen. I employee-menyalternativet kan du lägga till produkt, skriva ut alla produkter och ta bort produkt, se alla anställda, se egen personlig information, spara alla produkter som txtfil och logga ut.

I chef-menyalternativet loggar du in med din ID som skrivs ut när du startar programmet för första gången och ingen chef finns redan. I chef-menyalternativet kommer du till en undermeny där du kan göra samma saker som hos den anställda plus att du kan anställa, avskeda anställda, se alla kunder, spara alla anställda och kunder som ser-fil och läsa från fil.

Kunden, den anställda och chefen ärver från samma abstrakta superklass, Account. Fast hos den anställda och chefen ärver de från en annan abstrakt klass som ärver från Account, alltså StaffAccount. Varför den anställda och chefen ärver från StaffAccount och StaffAccount ärver från Account och inte att den anställda och chefen ärver direkt från Account har med att den anställda och chefen har variabler och metoder som kunden ej behöver få in i sin subklass.

Varför jag sedan delade upp det i två underklasser, den anställda och chefen är att jag inte vill ha mer än tre fält som deklarerats i konstruktorn.

## Övrigt

Programmet använder sig av ett ID-baserat och typbaserat för att hitta i arraylistan. Det typbaserade systemet används när jag ska upptäcka om det är en anställd eller chef. Jag har försökt att hålla koden lös för att tillåta framtida utveckling.

## Sammanfattning

CustomerAccount och StaffAccount ärver från samma klass, Account och EmployeeAccount och EmployerAccount ärver från StaffAccount. Programmet använder ett ID-baserat och typbaserat system för att hitta i arraylistorna.

