

## Design

Vi började med att designa GUI-t i ett Word dokument så att vi skulle ha klart för oss hur vi på ett ungefär ville att det skulle se ut när det var klart.

Själva programmet har vi valt att göra i imperativ kodstil eftersom vi tycker att det för mindre program, som för vårt, passar bättre med en linjär stil än vad objektorienterad kodstil ger. Vi har endast en klass för produkter så därför behövs inte några instansmetoder. Objektorienterad programmering hade sannolikt varit bättre om man behövt göra fler utökningar i programmet t.ex. flera användare med konton, flera kassor, beräkna lagerstatus, anställdas lön och liknande.

Vi har inte använt oss av inkapsling då inte det är nödvändigt i vårt program då användaren i Gui-t inte kan ändra datan i programmet. Det enda användaren kan mata in är rabattkoder och detta kollar programmet på olika sätt om den är riktig eller inte. Dessutom begränsas hur många tecken som kan skrivas i rutan för rabattkoder. Rent allmänt så har vi kodat så att texten i dom olika fönstren inte kan ändras av användaren genom att hen direkt klickar och ändrar innehållet i rutorna.

Koden till GUI-t har vi lagt upptill i programmet och vi har utformat den så att fönstret maximeras vid start och att knapparna är stora så att det skall vara enkelt att använda även på datorer med pekskärm (eller för den delen på en mobiltelefon). Vi har lagt produktbilden i en central position i GUI-t och gjort den relativt stor. Vi anser att det för en restaurang är viktigt med produktens utseende när man skall ta beslut om köp.

Det finns kod i programmet som behöver upprepas flera gånger. Denna kod har vi lagt i separata klassmetoder som vi anropar från flera ställen i programmet. Detta har vi gjort för att slippa upprepa samma kod flera gånger och därmed minska kodmängden samt risken för fel i koden.

Vi har valt att använda funktionell programmering med LINQ på två ställen där vi anser att den var lämplig då det förenklar koden.

Novak och Tomas