# Projektbeskrivning

Projektnamn Bussen

Projektägare Jonas S

Projektledare Jonas S

Projektperiod 2019-05-28 – 2019-06-02

## Sammanfattning

Projektet går ut på att skapa ett program som kallas bussen. Detta program kommer kunna hantera information om passagerarna på bussen. Syftet är att föra statistik över vilka som resor med bussbolaget.

# Bakgrund

Projektet kom fram efter att VD:n på ”Buss i Väst” tröttnat på alla åker med bussbolaget utan att betala för resan. Nu vill man ta fram statistik över alla som reser med bussbolaget. Därefter skall man sätta in en utredning som skall använda denna statistik som underlag för att hitta en lösning på problemet.

## Syfte

Ta fram data över alla som åker med på bussen. Statistiken kan användas i arbetet med att minska antalet fuskåkare.

## Mål

Målet med programmet är att få en överblick över vilka som reser med bussbolaget.

# Strategi

Man kan beskriva strategin med följande steg:

* Vid körning ska ett objekt *bussen* skapas och metoden *Run* ska köras.
* I metoden *run* presenteras en meny med möjligheter för användare att göra val
* Varje val ska representeras av metoder med egen funktionalitet

Metoder som kommer användas är:

* Chauffören skall via metoden *add\_passenger* kunna lägga till passagerare i bussen genom att fylla i ålder på passagerare. Information skall sparas i en vektor som har X – antal element. X styrs utav hur många platser det finns i bussen.
* I metoden print\_buss kan man när som skriva ut alla passagerare som sitter i bussen just nu.
* Olika metoder kommer användas för att ta fram information om passagerarnas ålder.

## Problem

En meny skall skapas och här gäller det att den är användarvänlig och funktionell. Efter att ett val gjort skall användaren få välja på nytt.

Har tidigare inte jobbat så mycket med klasser och metoder. Metoderna som skapas måste användas på rätt sätt för att programmet skall fungera.

# Tidsplan

Tidsplaneringen för detta projekt är följande:

2019-05-29: Skapa metoder och klasser till programmet. Menyn skall vara klar.

2019-05-30 – 2019-05-31: Finslipa det som ev inte blev klart 29/5. Utveckla programmets kärna.

2019-06-01: Gör klart det som återstår och testa program.

2019-06-02: Säkerställ så att program inte har några buggar i sig. Leverera till slutkund.

# Kravspecifikation

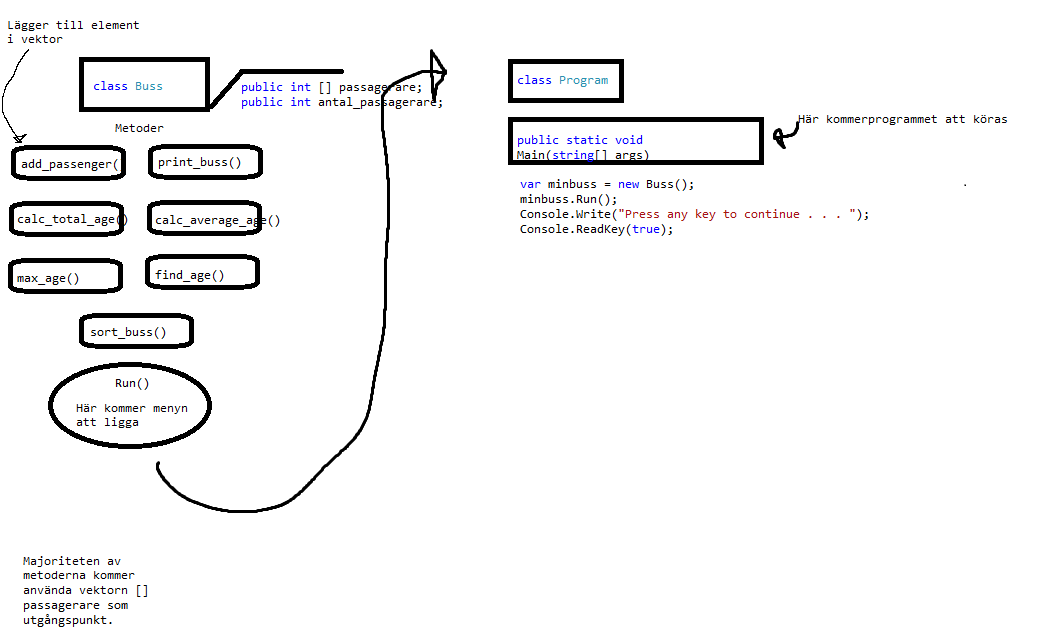
### Funktionella krav

* Programmet ska kunna lagra passagerare i en vektor och där dessa passagerare lagras som heltal (ålder)
* Passagerare ska struktureras som en klass med värden och metoder.
* Passagerare har en metod som körs när programmet exekveras och där denna metod styr en meny.
* Via menyn ska användaren kunna lägga till en passagerare
* Via menyn ska användaren kunna skriva ut alla värden i vektorn
* Via menyn ska användaren kunna räkna ut den totala åldern i vektorn
* Via menyn ska användaren kunna hitta den äldsta passageraren i vektorn.
* Via menyn ska användaren kunna hitta passagerare efter ålder och ett åldersspann.
* Via menyn ska användaren kunna sortera bussen efter ålder.
* Via menyn ska användaren kunna avsluta programmetin.

### Icke-funktionella krav

* Koden är konsekvent och kommenterad på ett enkelt sätt
* Programmet ska gå att exekvera och koden är robust
* Programmet ska upplevas rimligt att interagera med
* Programmet ska inte ha överdrivet enkelt för att krascha

# Bilaga 1: Diagram



# Bilaga 2: Pseudokod

Följande är kraven vad avser diagram:

Skriva ut alla tal:

START

Skriv ut : ” Alla passagerares ålder är..”

Loppa igenom alla element i vektor [] passagerare

Och

Skriv ut samtliga passagerare och ålder på enskild individ.

STOP

START

Int maxVärde;

Int maxVärde = passagerare.Max();

Returnera maxVärde

Skriv ut maxVärde(Äldsta passagerare)