# Laboration – Text inserter

*Denna laboration kan lösas individuellt eller i grupp (max 3 personer). Stickprov där du ombedes redogöra för din lösning muntligt kan förekomma.*

## Bakgrund

Att kunna spara ner meddelanden i text är vitalt för ett bra liv idag. Applikationen vi arbetar med i denna laboration gör just detta.

Men det har visat sig att risken att någon tjuvläser meddelanden är vanligt. Vi har därför bestämt att anlita dig för att utöka funktionaliteten i applikationen.

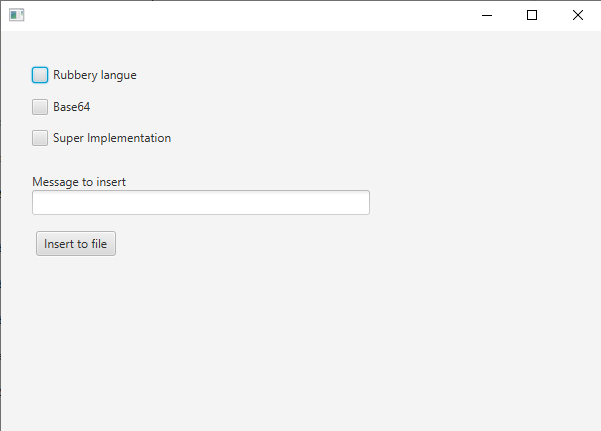
Designprincipen: Open-closed (*[Open for extension but closed for modification](https://en.wikipedia.org/wiki/Open%E2%80%93closed_principle)*) har vi förstått är av stor vikt vid utveckling av system. Vi önskar därför att utbyggnaden av applikationen skall göras med hjälp av designmönstret [**Decorator**](https://sourcemaking.com/design_patterns/decorator)**.**

## Projektet

Kod att utgå ifrån hittas här: <https://bitbucket.org/margob/exam_text-insert_a-d/src/master/>  
( *git clone https://margob@bitbucket.org/margob/exam\_text-insert\_a-d.git* ).

## Att göra

## När du startar applikationen så möts du av följande:



I applikationen är det förnärvarande möjligt att skriva in ett meddelande och trycka på ”Insert to file” det som då sker är att meddelandet sparas i en .txt fil i ditt projekt.

Vi vill nu utöka funktionaliteten så man via checkboxar (redan implementerade i vyn) kan välja om texten skall modifieras (encodas) innan den sparas i text-filen.

Det skall gå att encoda med: Base64, Rövarspråket och Super Implementation (du väljer vad detta skall vara).

### Konkret:

* Få önskat resultat i applikationen genom att implementera Decorator pattern.
* Bifoga i din inlämning en .pdf som innehåller följande:   
    
  1. Beskriv och exemplifiera hur din implementation är kopplad till Open-closed principen?

## Förberedelser

* Föreläsningar och laborationer på sal vecka 1 och 2.
* Kursbok, kap 1 och 3.

## Inlämning

Du laddar upp en .zip innehållandes:

1. Koden för en fungerande lösning av applikationen där Decorator pattern är implementerat.
2. En .pdf med efterfrågat innehåll.

Om du har jobbat i grupp så skickar ni alla in samma fil och inkluderar då även:

1. En .txt som innehåller *”Vi (”namn1”, ”namn2” och ”namn3”) har arbetat i grupp med denna lösning. Och försäkrar här med att vi samtliga har aktivt deltagit samt nu anser att vi kan redogöra för lösningen om så skulle bli aktuellt”.*

## Betyg

Inlämningen kan ge något av betygen komplettera (U) eller godkänd (G).

### Betygskriterier

När vi betygsätter inlämningen tittar vi på om:

* Efterfrågat mönster finns implementerat.
* Frågan besvaras korrekt givet din/er implementation.
* Kvalitet på lösningen.

## Exempellösningar för rövarspråket och Base64:

Base64.getEncoder().encodeToString(“STRING”.getBytes());

public String rubberLang(String message) {

String vowels = "aeyuoåäöi";

String outString = "";

int length = message.length();

for (int i = 0; i < length; i++) {

if (vowels.contains(message.toLowerCase().charAt(i) + "")) {

char c = message.charAt(i);

outString += c + "o" + c;

} else {

outString += message.charAt(i);

}

}

return outString;

}