

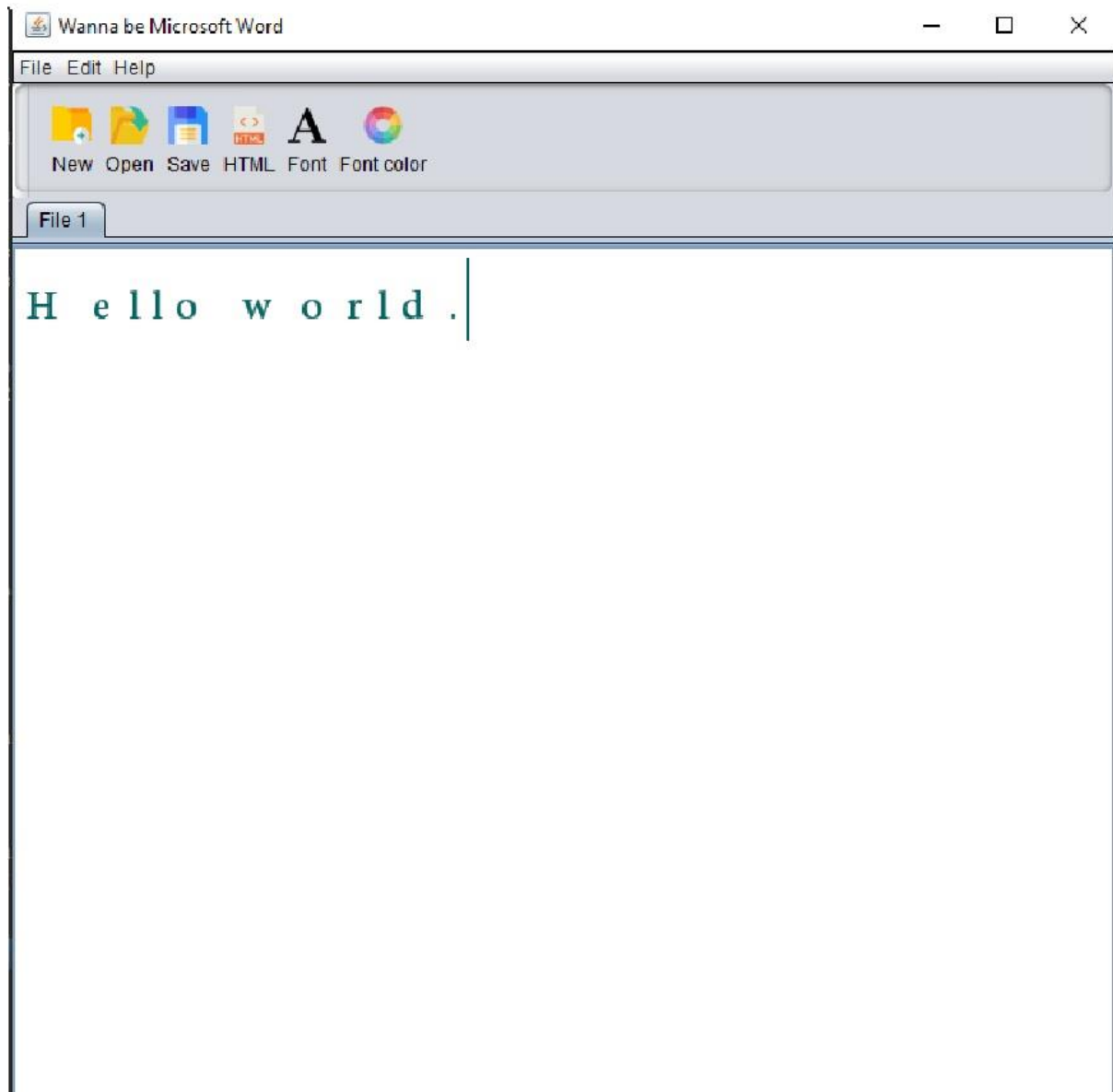
## Dokumentation av Text Editor.

Detta dokument kommer innehålla information om min text editor som jag byggt i NetBeans. Jag kommer beskriva uppbyggnaden av programmet samt vilka funktioner den har och hur jag gått tillväga med detta projekt.

## Uppbyggnad.

En JMenuBar med en JMenu som består av 3 st menyval "File", "Edit" och "Help".

Menyvalet "File" består utav 5 JMenuItem:



### New

Ska ge användaren möjlighet att skapa en ny tab där man kan skriva in ny text. Har man skrivit text i befintlig tab och inte sparat filen frågar programmet när man öppnar en ny tab om man vill spara texten som är i den befintliga taben.

## **Open**

Ger användaren möjligheten att öppna befintliga filer. När man trycker Open får användaren en förfrågan om man vill öppna filen i befintlig tab eller skapa en ny tab med den nya filen.

## **Save**

Ger användaren möjligheten att spara sin text till en txt fil. Har man i programmet öppnat en redan existerande fil skall programmet komma ihåg sökvägen till den filen och automatiskt spara ändringarna till samma fil.

## **Save as**

Ger användaren möjligheten till att skapa och spara en ny version av filen med ett nytt namn och plats.

## **Exit**

Ger användaren möjligheten att avsluta programmet, när man trycker på på exit via menyvalet eller kryssrutan uppe till höger skall programmet visa ett pop-up meddelande som frågar om användaren är säker sitt val.

Menyval "Edit" har en JMenuItem:

## **Clear**

Ger användaren möjligheten att snabbt rensa hela textfältet i befintlig tab.

Menyval "Help" innehåller en JMenuItem:

## **About**

Ger användaren möjligheten att få upp en JFrame med information om programmet som namn på programmet, författare samt vem programmet är licensierat till.

En JToolBar som består av 6st JButtons med funktionerna "New", "Open", "Save", "HTML", "Font" samt "Font color"

Funktionerna New, Open och Save har exakt samma funktionalitet som de JMenuItem:s under menyn "File". Anledningen till att jag valt att lägga dom i min JToolBar är för att detta är funktionerna som används mest i en text editor och jag vill dessa funktionerna ska vara lättillgängliga på ett smidigt sätt till användaren som arbetar med editorn.

## **HTML**

Ger användaren en möjlighet att generera en färdig HTML mall i befintlig tab.

## **Font**

Ger användaren möjlighet att ändra fonten på sin text i den tab användaren befinner sig i. Användaren är inte begränsad till att använda endast en font i alla tabbarna utan man kan variera med fonter i de olika filerna om man arbetar med flera filer.

## **Font color**

Ger användaren möjlighet att ändra färgen på fonten och sin text i den tab användaren befinner sig i. Precis som i "Font" funktionen är inte användaren begränsad till att använda en och samma färg i alla tabar och textfiler utan det är fritt för användaren att kombinera färger.

Metoder som jag byggt i detta projekt är följande:

**Public void createTabs();**

Syftet med denna metod är att skapa nya tabbar, denna metod ligger inåt i knapparna "Open" samt "New" i programmet och är den metod som instansierar nya tabbar med textfält.

**Public void validateBeforeNew();**

Syftet med denna metod är att varna användaren ifall denne har osparade ändringar innan de väljer att skapa en ny tab eller öppna ett nytt dokument. Metoden lagrar två strängar som jämförs med varandra för att sedan kunna se om det har skett en ändring, har det skett en ändring varnas användaren om ingen ändring har skett säger programmet ingenting.

**Public void validateBeforeClosing();**

Precis som funktionen ovan om man väljer ett nytt dokument så varnar den här metoden innan man stänger ner programmet. Har man inte gjort någon ändring i dokumentet ställer programmet frågan "Are you sure you want to exit?" har man en osparad ändring i filen varnar programmet för detta genom att säga "You have unsaved changes! Are you sure you want to exit?".

**Private void saveFile(StringName);**

Den här metoder ger användaren möjlighet till att spara den filen användaren arbetar med. Denna metoden ligger under knapparna "Save" och "Save as". Om användaren laddar upp en befintlig fil i text editorn och gör en ändring kommer programmet att spara filen till samma sökväg. Skapar användaren ett nytt dokument från grunden kommer programmet att be användaren välja vart på datorn filen ska sparas samt välja ett namn för filen.

**Public void openMultipleFiles();**

Denna metoden ger användaren möjlighet att öppna filer i den tab som användaren befinner sig i. Den använder sig utav en "JFileChooser" för att öppna filer, med hjälp av en scanner kan jag även scanna det som innehållet i filen och hämta det till min JTextArea för att sen presentera txt dokumentet för användaren.

**Private void initFirstTab();**

Denna metod initierar en första tab med en JTextArea samt en JScrollPane till TextArean. Den skapar även upp en indexlista som håller reda på hur många antal tabbar som öppnats under sessionen.

Jag har även byggt några enstaka metoder som är väldigt små som jag väljer att kommentera i koden då dessa är endast på 2-3 rader. Den metod som kan vara värd att nämna är:

```
int sel = jTabbedPane1.getSelectedIndex();
```

```
JTextArea multipleTabsTextArea = (JTextArea) (((JScrollPane)  
jTabbedPane1.getComponentAt(sel)).getViewport()).getComponent(0);
```

Dessa tre rader är kritiskt viktiga för programmet då det är genom dessa jag kan hjälpa programmet känna av vilken tabb användaren befinner sig på. Om användaren har fyra st dokument öppnade och befinner sig på dokument nr 3 känner text editorn igen detta och de ändringar som görs sker endast i detta dokument. Vill användaren ändra färgen texten i detta dokument så skall inte resterande tre dokument påverkas.

## Tillvägagångssätt och resultat.

Efter mycket blod, svett och tårar samt väldig många timmar på sidor som YouTube, Google, Stackoverflow m.m så skrapade jag ihop en text editor som jag är väldigt nöjd med. Detta har varit en väldigt svår men rolig uppgift och den har gett mig mycket. Jag har inte bara lärt mig ny kod men uppgiften har även gjort mig bättre på att söka efter problem på nätet och förstå andra människors kod samt hur de tänkt.