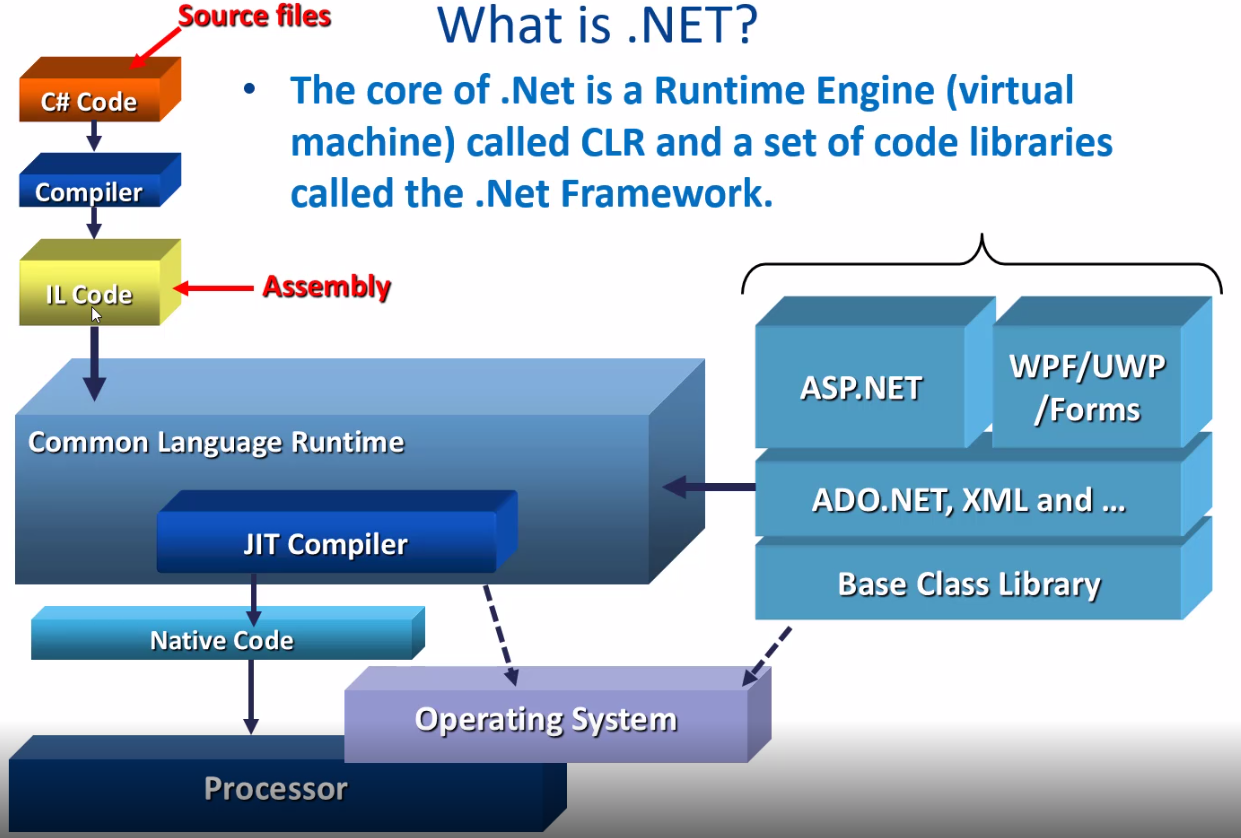
# GUI eksamen – Jacob Nyvang Hansen

## WPF

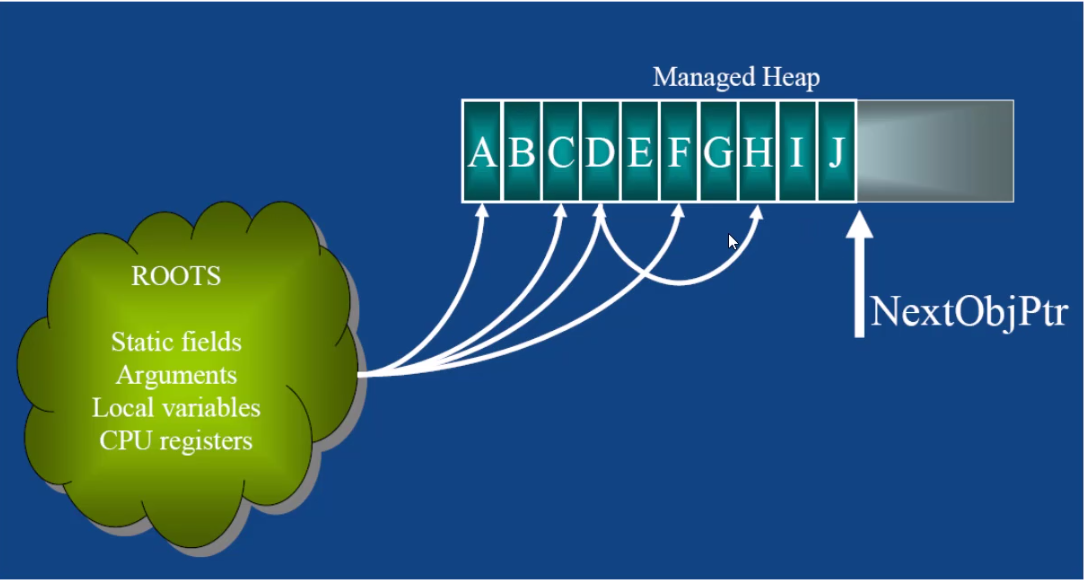
**Redegøre** for principperne i .Net frameworket og dets overordnede arkitektur samt beskrive og anvende programmeringssproget C#



* Compiler – comiler
* IL code – Laver .exe og .dll filer
* CLR – Virtuel maskine der kompilerer (loader ved startup, og laver en jit (just in time) kompilering)  
  garbage collector er i CLR’eren også.
* En masse biblioteker ASP.NET osv.

GarbageCollector er automatisk, allokere på heap, holder selv styr på om noget bruges.

Marking (Løber igennem det hele, og rykker ned til frie pladser) arbejder rekursivt.



Figur Garbage collector (FØR)

Der laves generationer for gb collectoren, hvor man kun kigger på generation 0 (De nyeste).

Kunne **anvende** MVVM arkitekturen til at implementere applikationer med en grafisk brugergrænseflade til Microsoft Windows platformen ved brug af .Net frameworket og programmeringssprogene C# og XAML.

* Model – View – Viewmodels. Hovedargument = synkroniseret og testbar uden UI, lavere kobling.

**Anvende** kontroller til opbygning af både modale og modeless dialoger, samt kunne anvende de forskellige layout panels.

* Modale = Har ”låst” prioritet (showDialog()) 🡨 Den jeg bruger
* Modeless = Man kan skifte mellem vinduerne (show())

Historik kunne godt have været en modeless

**Anvende** WPF’s faciliteter til tegning af 2D grafik samt visning af billeder.

* Kig på forsiden af WPF’appen både billede og ellipse

**Anvende** styles og ressourcer.

* Kig i app.xaml (Grid, Label, button og TextBox)

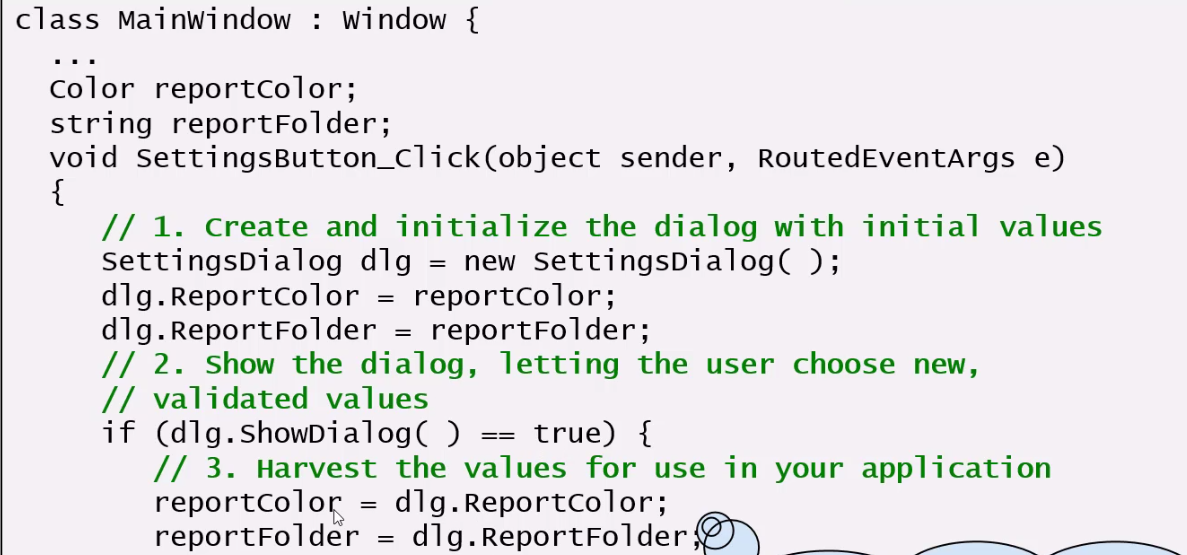
**Redegøre** for WPF’s faciliteter til kommunikation mellem bagrundstråde og GUI-tråden i flertrådede programmer.

* Der er en gui tråd, som er den eneste der kan bruge brugergrænsefladen, andre tråde giver en exception (der er fire måder)
* Dispatcher for at komme til UI tråden (kan altid bruges)
* BackgroundWorker for at indkapsle baggrundstråden (ikke så relevant)
* Data binding til at opdatere UI’et
* Await (når muligt) (alfa omega).

Generelle ting ved WPF

Setting mangler et closing event, som spørger om man vil gemme ens settings, eller hvis ens data er ugyldig, skal den ikke lukke og sætte de gamle værdier.

Alt er sat til at åbne center, ud over main vinduet.



Settings burde være lavet på den her måde.

Save og load burde have common dialogs, nåede jeg ikke (Save og load vinduer)

## ASP.NET

**Redegøre** for arkitekturen for en Webapplikation.

* Request-response
* Client -> det der viser ting (HTML)
* Server -> Det der gør ting (C# og CSS)

**Anvende** data binding til at sammenknytte data i modellaget med deres præsentation i viewlaget.

* Inputtypes
* Mellem Javascript og Views (ved brug af HTML ID)

**Designe og implementere** Webapplikationer med en grafisk brugergrænseflade med brug af HTML5, CSS og javascript.

* Præsenter dine ting

**Anvende** et server side MVC framework til udvikling af Webapplikationer

* Kun V af MVC bliver vist på klienten

**Anvende** et clients side MVC framework til at bygge Web applikationer.

* Hele MVC ligger på client
* Vue
* Bruger web api