

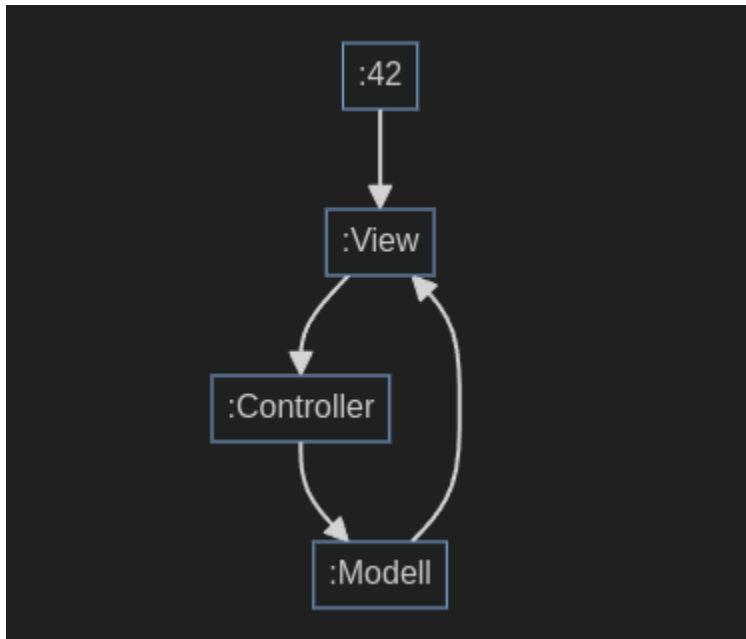
%Overview

- **A1:** Vilka uppgifter har M, V respektive C i vårt program?
 - M = Modellen, huvudfokus, innehåller algoritmer som ska hantera alla funktioner som ska kunna utföras i XL.
 - V = View, egentligen given i koden, vi ska komplettera så att man kan integrera med vyn.

Ska innehålla actions som skickas till *CONTROLLER* när vi vill ändra på något.
 - C = Controller, ska koordinera alla ändringar i modellen, vart vi klickar.

Koden som gör det vi vill göra
- **A2:** Vilka klasser syns i figuren ovan?
 - Klassen som "syns" är view klassen som är GUI'n
- **A3:** Förklara kortfattat vad var och en av klasserna `SlotLabel`, `SlotLabels`, `Editor`, `StatusLabel`, `CurrentLabel` och `XL` i den utdelade koden gör.
 - `SlotLabel` Kodar för en ruta utseendet av rutan.
 - `SlotLabels` Läger alla rutor vi vill ha i en lista och itererar över vyn.
 - `Editor` Är själva textboxen där man skriver in värden och figurer
 - `StatusLabel` Checkar statusen på cellen, om den är tom, innehåller något eller är intryckt.
 - `CurrentLabel` Väljer dem rutorna som man har skrivit in i "textbaren"
 - `XL` XL är main klassen som knyter ihop alla metoderna för att skapa själva vyn.

- **A4:** Användningsfall: Någon skriver in 42 i **Editor** – vad skall hända för att vi skall se värdet 42 i rutnätet (dvs vilken väg skall värdet gå igenom M, V och C)?
 - Vi skickar 42 i view som skickas sen till controllern och stämmer av ändringen



Modell

- **Tips:** Paketet **expr** innehåller i princip allt som behövs för att tolka och beräkna uttryck.
- **B1:** Hur representeras adresser (det är redan bestämt i den givna koden)? Kan ni komma på något alternativ?
 - Adresser representeras i cellerna
- **B2:** Hur modellerar vi att en ruta (cell) kan innehålla antingen en kommentar eller ett uttryck?
 - Vi borde kunna ändra på **StatusLabel** och ange om cellen ska innehålla en kommentar eller uttryck
- **B3:** Vilket slags cell får vi om vi matar in värdet 42 i editorn?
 - En numerisk cell
- **B4:** Vilka klasser behövs i modellen (utöver dem i expr-paketet)?
 - Det saknas klasser för att beräkna bombceller.
- **B5:** Vilken klass håller reda på alla celler?
 - **SlotLabels** håller reda på alla celler

- **B6:** Vilken datastruktur är mest naturlig för att hålla reda på alla celler?
 - Matris är den mest naturlig för att hålla reda på alla celler
 - **B7:** Uppgiftstexten ovan innehåller formuleringen: "Det är ett krav från uppdragsgivaren att minnesbehovet för modellen av kalkylarket ej skall bero på arkets storlek utan bara på den mängd information som matats in i arket." Hur gör vi för att inte använda mer minne än vad som verkligen behövs?
 - Det finns några olika lösningar, till exempel kan vi implementera en sparse datastruktur där vi endast lagrar celler som innehåller data i minnet.
-

Modell Environment

- **C1:** När ett uttryck som innehåller en adress skall beräknas använder vi **Environment** – varför?
 - Environment lagrar värdet av adressen och därför använder vi environment för att beräkna uttryck.
 - **C2:** Vilken klass skall implementera **Environment**?
 - Expr implementerar environment för att kunna hantera aritmetiska uttryck
 - **C3:** Vilken praktisk nytta har vi av **Environment** interfacet innan vi börjar skriva vår kod?
 - Innan vi börjar skriva kod kan vi se att Environment klassen innehåller en metod:

public double value(String name); Som tar emot en sträng namn som ska vara en identifierare eller variable som en input och returnerar en double.
 - **C4:** Vilken princip dikterar att vi skall använda ett **Environment** interface?
 - **Environment** följer "Dependency Inversion" principen från SOLID.

Abstraktion: **Environment** är en icke konkret klass som tar emot ett värde av en variabel.

Dependency: Andra klasser som vill komma åt variabel värden behöver inte direkt luta på en konkret implementation men istället lutar man på det abstrakta gränssnittet **Environment**.

Inversion: Högnivå komponenter såsom **ExprParser** är beroende på abstrakta interfacet **Environment**.
-

Koppling mellan M och V/C

- **D1:** Vilka klasser i modellen behöver vårt GUI känna till?
 - Num, Variable, Add, Sub, Mul och Div
 - **D2:** När vårt GUI hämtar värden att skriva ut i `SlotLabel` eller `SlotLabels`, vilka värden, och vilken typ vill vi få tillbaka?

Vi bör få en expression som har antingen ett numerisk värde eller en sträng
 - **D3:** När användaren klickar i en cell, och vår `Editor` skall uppdateras, vilket värde, och vilken typ vill vi få tillbaka och lägga i editorn?
 - Editor extendar `TextField` som kan endast lagra strängar i cellerna men eftersom strängarna kan vara av typen `int` eller `double` skulle vi behöva parse-a strängen för att få tillbaka rätt värde
-

Koppling mellan M och V/C: felhantering

- **E1:** Användningsfall: Användaren matar in en felaktig formel – vad skall hända?
 - Vi borde få en `IllegalFormatException` i koden och sen hantera det genom att visa ett error meddelande i vyn
 - **E2:** Användningsfall: Användaren ändrar en cell så att vi någon annanstans får division med 0, vad skall hända?
 - Vi borde få en `ArithmeticException` i kode och då skulle vi kunna hantera genom att kasta en error till användaren.
 - **E3:** Användningsfall: Användaren matar in en cell så att vi får en cirkulär referens, vad skall hända?
 - Vi ska få en referens error `#REF` i statusfältet
 - **E4:** Allmänt: i vilket paket upptäcker vi felen?
 - I `Expr`-paketet i klassen `ExprParser` upptäcks skriv fel med `Exceptions`
 - **E5:** Allmänt: i vilket paket hanterar vi felen, och hur?
 - I `util` paketet hanterar vi felen genom att signalera användargränssnittet med ett error meddelande att något fel har hänt.
-

Övrigt

- **F1:** Hur upptäcker vi cirkulära referenser?
 - Vi måste iterera genom hela arket för att upptäcka om vi har cirkulära referenser.
- **Tips:** Gör inga optimeringar, åtminstone inte förrän allting fungerar!