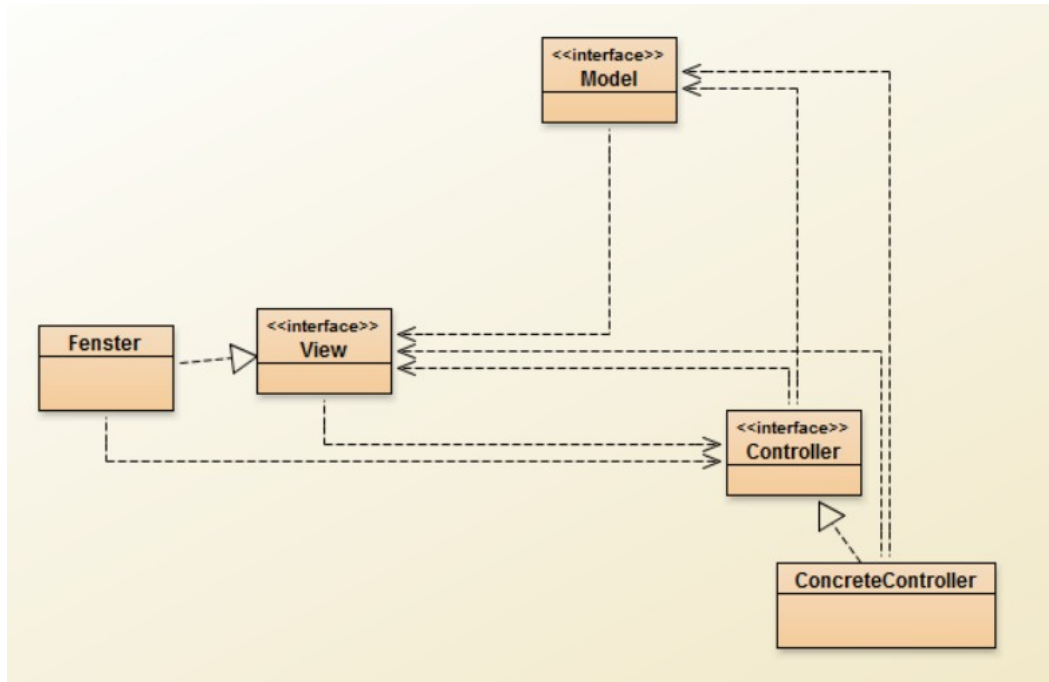


Tic-Tac-Toe



<<interface>> View
void controllerSetzen(Controller) void bildZeichnen(int, String) void textZeichnen(int, String)

Das Interface View legt die Darstellung fest. Es gibt eine Methode zum Setzen des Controller-Referenzattributs, und je eine Methode, um an Feld/Zelle/Knopf mit einer bestimmten Nummer (angefangen bei 0) ein Bild zu zeichnen oder einen Text darzustellen. Die Klasse Fenster ist bereits spielfertig, man muss nichts an ihr ändern. Folgende Bilder-Strings sind möglich: „bilder/0.png“ bis „bilder/16.png“ sowie „bilder/x.png“ und „bilder/o.png“.

<<interface>> Controller
void nachrichtEmpfangen(int) void modelSetzen(Model) void viewSetzen(View)

Das Interface Controller legt die Steuerung fest. Der View ruft die Methode „nachrichtEmpfangen“ auf mit der Nummer des angeklickten Feldes als Parameter. Der Controller fragt dann beim Model nach, ob das Feld bereits besetzt ist oder noch leer, und wenn leer, dann – abhängig davon, wer gerade am Zug ist – schickt er ein „x“ oder ein „o“ an das Model und weist den View an, den entsprechenden Text zu zeichnen oder ein Bild darzustellen (falls das nicht das Model übernimmt).

<<interface>> Model
void viewSetzen(View) char belegungGeben(int) void feldBelegen(int, char)

Das Interface Model enthält den aktuellen Spielzustand. Der kann für Tic-Tac-Toe mit einem eindimensionalen char-Feld der Länge 9 implementiert werden, dessen Zellen alle mit ' ' initialisiert werden. Mit der Methode belegungGeben(int) erhält man die aktuelle Belegung der entsprechenden Zelle zurück, mit der Methode feldBelegen(int, char) kann man eine beliebige Zelle mit einem beliebigen Zeichen füllen.

Aufgabe:

- Schreiben Sie eine Klasse, die das Model implementiert.
- Oder schreiben Sie eine Klasse, die den Controller implementiert.
- Die Änderungen am View können dabei Model oder Controller übernehmen.)