Java07 − Schachbrett − Vorübung: Schachbrett-Reihe

Projekt-Beschreibung: Schachbrett

Wir wollen als BlueJ-Projekt ein Schachbrett zeichnen lassen.

Als Vorlage verwenden wir dazu das BlueJ-Projekt zum Zeichnen von Objekten (aus Jgst. 9).

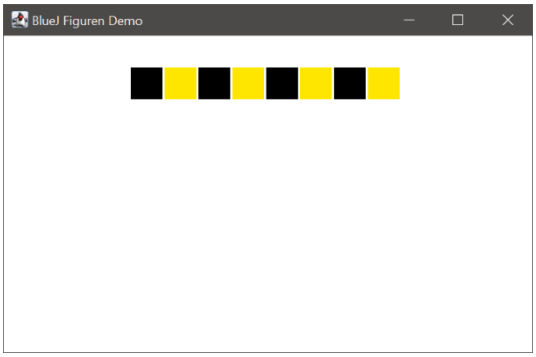
Als Vorübung erstellen wir eine Klasse „Reihe“ als eindimensionales Array von Quadraten, mit der eine (einzige) Schachbrett-Reihe gezeichnet werden kann.

Dann erstellen wir eine Klasse „Schachbrett“ als zweidimensionales Array von Quadraten, mit der das zweidimensionale Schachbrett gezeichnet werden kann.

Schritt 0 − BlueJ-Vorlage „Java_04_Schachbrett“ in das eigene Home-Laufwerk kopieren

Kopiere (d. h. KEIN Doppelklick) den gesamten Ordner „Java_04_Schachbrett“ in Dein Home-Laufwerk. Du findet diesen Ordner (diese BlueJ-Vorlage) unter „Tausch“ → „Klassen“ → „10d“ → „10doraInfBAU“.

Öffne dann in Deinem Home-Laufwerk diesen Ordner und starte mit einem Doppelklick auf die Datei „package.bluej“ das BlueJ-Projekt „Java_04_Schachbrett“.



Die im Klassen-Diagramm erscheinenden Klassen „Leinwand“, „Kreis“, „Quadrat“, „Dreieck“ und „Person“ kommen Dir bestimmt bekannt vor (Jgst. 9).

Schritt 1 − Klasse „Reihe“ erstellen

Eine Schachbrett-Reihe modellieren wir als Feld (Array) aus 8 Quadraten.

Lege im BlueJ-Projekt „Java_04_Schachbrett“ eine neue Klasse „Reihe“ an, öffne den Quelltext und lösche den gesamten „automatisch“ erzeugten Quelltext.

Kopiere in den (nun leeren) Quellcode den folgenden Code:

```
public class Reihe {
    private Quadrat[] reihe;    // Deklaration „reihe“ als Array v. Quadraten

    public Reihe() {    // Konstruktor
        reihe = new Quadrat[8];    // Initialisierung als Feld m. 8 Quadraten
            for (int j = 0; j < 8; j++){
                reihe[j] = new Quadrat();   // jedes Feld-Element ein Quadrat
            }
    }

    public void bildMalen() {    // Methode zum Malen des Bildes (Leinwand)
            for (int j = 0; j < 8; j++){
                reihe[j].sichtbarMachen();
            }
    }
}
```

Übersetze die Klasse, erzeuge ein neues Objekt „reihe1“ dieser Klasse „Reihe“ und rufe die Methode „`bildMalen()`“ auf.

Was sieht man auf der „Leinwand“? Warum ist das so?

Inspiziere das gerade erstellte Objekt „reihe1“ so weit (bzw. tief), bis keine „Zeiger“ (gebogene Pfeile) mehr zu sehen sind.

Inspiziere verschiede Elemente des Feldes „Quadrat[]“!

Nun ist klar, warum man auf der „Leinwand“ nur ein einziges rotes Quadrat sieht.

Schritt 2 − Ausbau des Konstruktors

Beim Aufruf des Konstruktors (d. h. bei der Erzeugung eines neuen Objekts der Klasse „Reihe“) wird ein Feld aus 8 Quadraten erzeugt. Jedes Feld-Element (d. h. jedes Quadrat) wird mit den Standard-Werten für die Erzeugung eines Quadrats versehen (zu finden im Quellcode der Klasse „Quadrat“).

Was ist zu tun, damit eine Schachbrett-Reihe von Quadraten erzeugt wird?

a) Größe der Quadrate ändern (für alle Quadrate, unabhängig vom Feld-Index j)
b) Position der Quadrate ändern (in Abhängigkeit vom Feld-Index j)
c) Farbe der Quadrate ändern (in Abhängigkeit vom Feld-Index j)

zu a) Größe der Quadrate ändern

Hierzu rufen wir für jedes Feld-Element „`reihe[j]`“ (d. h. für jedes Quadrat der Reihe) die Methode „`groesseAendern(int neueGroesse)`“ (der Klasse Quadrat) auf und übergeben als „neueGroesse“ den Wert „30“, damit die Quadrate eine Seitenlänge von 30 Pixeln erhalten.

Im Quellcode des Konstruktors muss (nur) die blaue Zeile eingefügt werden (innerhalb der for-Schleife):

```
reihe[j] = new Quadrat();
reihe[j].groesseAendern(30);
```

zu b) Position der Quadrate ändern

Hierzu rufen wir für jedes Feld-Element „`reihe[j]`“ (d. h. für jedes Quadrat der Reihe) die Methode „`positionSetzen(int xPos, int yPos)`“ (der Klasse Quadrat) auf.

Die y-Position soll für alle Quadrate der Reihe den Wert „`30`“ bekommen (30 Pixel Abstand zum Rand oben).

Die x-Position muss natürlich unterschiedlich sein, damit eine Reihe entsteht. Sie hängt vom Feld-Index j ab und soll den Wert „`120 + 32*j`“ erhalten (2 Pixel Abstand zwischen den Quadraten).

Füge den hierfür notwendigen Code (unterhalb von **`reihe[j].groesseAendern(30);`**) in den Quellcode des Konstruktors ein.

Übersetze die Klasse, erzeuge ein neues Objekt „reihe1“ der Klasse „Reihe“ und rufe die Methode „`bildMalen()`“ auf.

Was sieht man nun auf der „Leinwand“?

zu c) Farbe der Quadrate ändern

Bei einem geraden Feld-Index (also bei j = 0, 2, 4, 6) soll das entsprechende Quadrat die Farbe „schwarz“ erhalten, bei einem ungeraden Feld-Index (also „sonst“ ≙ „else“) die Farbe „gelb“.

Testen, ob der Feld-Index j eine gerade Zahl ist, kann man mit der „modulo-Funktion“.
Die „modulo-Funktion“ „a % b“ berechnet den Rest bei der Division von „a“ durch „b“. Also:

„j % 2“ liefert den Wert „0“, wenn j eine gerade Zahl ist, und den Wert „1“, wenn j eine ungerade Zahl ist.

Nötig ist also eine bedingte Anweisung („`if (…)`“) mit „`j % 2 == 0`“ als Bedingung.

Wenn die Bedingung erfüllt ist, soll mit der Methode „`farbeAendern(String neueFarbe)`“ (der Klasse „Quadrat“) das entsprechende Quadrat die Farbe „schwarz“ erhalten, ansonsten die Farbe „gelb“.

Füge den hierfür notwendigen Code (unterhalb von **`reihe[j].positionsSetzen(…);`**) in den Quellcode des Konstruktors ein.

Übersetze die Klasse, erzeuge ein neues Objekt „reihe1“ der Klasse „Reihe“ und rufe die Methode „`bildMalen()`“ auf.

Was sieht man nun auf der „Leinwand“?

Wenn eine schwarz-gelbe Reihe aus 8 Quadraten zu sehen ist (wie auf der ersten Seite):

Herzlichen Glückwunsch!