Programmierzertifikat Objekt-Orientierung mit Java, Blatt Nr. 9
http://proglang.informatik.uni-freiburg.de/teaching/java/2009/

9.1 Aufgabe, Die Blockliste*

Verwenden Sie die Klassen AWorld, Field und World aus der Übung von letzter Woche und erstellen Sie die folgenden zwei Klassen: Block und BlockList. Erstellen Sie auch ein Interface IBlock, das von der Klasse Block implementiert wird. Das Inteface soll folgende Methoden fordern:

- **void** draw(Graphics g, int blockwidth, int blockheight)
- int getX()
- int getY()

Instanzen von Klassen, die dieses Interface implementieren (wie z.B. die Block Klasse, aber später eventuell auch noch die Stone Klasse, deshalb ein Interface!), sollen bei der draw Methode sich selber auf den Graphik Kontext malen. Hierbei ist die Position den Objekten selber bekannt (z.B. die Koordinaten des Blocks). Die Information wie breit bzw. hoch ein Block ist ermöglicht den Blöcken dann ihre Rechtecke passend zu positionieren.

Hinweis:
Die Blockgröße an die Methode zu übergeben ermöglicht später recht einfach, eine verkleinerte Vorschau des nächsten Steins zu erstellen. Dabei wird dieselbe draw Methode verwendet, und es muss nur die Blockgröße verändert an die Methode übergeben werden. Alternativ könnte die Blockgröße auch an den Konstruktor übergeben werden, oder als static final Wert in einer passenden Klasse gespeichert werden. Die hier gewählt Version ist, in Bezug auf Erweiterbarkeit für später, am flexibelsten. Bei den anderen Versionen sollte das Interface angepasst werden, und die blockwidth und blockheight Werte müssen nicht übergeben werden.

Die Blockliste (Blocklist) soll eine Liste von IBlocks enthalten und mindestens folgende öffentliche Methoden bereitstellen:

- boolean add(IBlock b)
- boolean isFree(int x, int y)
- **void** remove(int x, int y)
- **void** draw(Graphics g, int width, int height)
- boolean isFull()
- **void** clear()

Die Methode add soll einen Block zu der Liste hinzufügen. Hierbei soll geprüft werden, ob das Feld, auf das der Block gelegt werden soll, frei ist.

Wie Sie sehen soll es möglich sein, Blöcke aus der Liste zu entfernen, indem die Koordinaten des Blocks angegeben werden. An dieser Stelle sollten Sie mit dem `extended-For`-Statement aufpassen (dieses ist zum löschen aus einer Liste nicht geeignet). Verwenden Sie deshalb an dieser Stelle eine `while`-Schleife, und entfernen Sie den Block über den `Iterrator`.

Binden Sie die Blockliste in die Field Klasse ein, so dass Sie ein Spielfeld sehen. Ein Feld sollte hierzu eine Methode newGame() erhalten, dass ein neues Spielfeld erzeugt, und leer zeichnet. Erstellen Sie per Programmcode an der richtigen Stelle (an welcher Stelle in der World?) einige Blöcke, fügen Sie diese der Blockliste hinzu und testen Sie auf diese Art und weise, ob Ihre Blöcke korrekt gezeichnet werden.

Hinweis:
Die Klasse Graphics besitzt die Methode create(int x, int y, int width, int height). Diese erzeugt ein neues Objekt der Klasse Graphics, dass allerdings eine anderes Koordinatensystem besitzt. So können Sie in der Field Klasse einen Rahmen für Ihr Spielfeld zeichnen, und dann ein neues Graphics Objekt erzeugen, dass Sie dann an die Methode draw der Blocklisten-Klasse übergeben. Auf diese Art und Weise sparen Sie sich die Übergabe eines Offsets für die Koordinaten beim Zeichnen der Blöcke.
Zwei Methoden von Graphic Contexten, die Sie nützlich finden könnten sind: create, fill3DRect.

9.2 Aufgabe, Block - Abwandlung und Steine*

Wandeln Sie nun die Blockklasse in eine abstrakte Klasse um, und entfernen Sie die draw Methode aus dieser Klasse. Erzeugen Sie nun eine Reihe von speziellen Blockklassen, so dass die unterschiedlichen Blöcke unterschiedlich aussehen (sie sollen aber immer quadratisch sein).

Schreiben Sie eine abstrakte Klasse AStone, die eine Liste von Blöcken enthält. Spezielle Unterklassen der abstrakten Klasse sollen dann die unterschiedlichen Steinformen (L, T, usw.) repräsentieren. Überlegen Sie, welchen Code sie in der abstrakten Klasse schreiben können, damit Sie diesen nicht mehrmals in den Unterklassen schreiben müssen.

Fügen Sie zu ihrem Feld einen Stein hinzu. Erweitern Sie nun den Stein, so dass er sich auf dem Feld nach unten bewegt, wenn die onTick Methode der Weltklasse aufgerufen wird. Achten Sie hierbei darauf, dass der Stein nur dann bewegt werden kann, wenn die passenden Felder im Spielfeld frei sind.