

1. Angekündigter Test im Fach Informatik  
am 02.12.2020  
Klassen und deren Implementierung

Szenario: In einem Computerspiel lässt man in einer virtuellen Welt verschiedene Arten von Katzen verschiedene nützliche Dinge tun. Unter anderem gibt es Mausekatzen, die für das Jagen von Mäusen zuständig sind (so müssen Getreidevorräte geschützt werden). Für jede Mausekatze muss gespeichert werden, wie gut sie bei Kräften ist und wie viel Erfahrung im Mäusefangen sie hat. Um Mäusefangerfahrung zu sammeln, muss die Katze jagen. Allerdings verbraucht sie beim Jagen natürlich Energie; diese gewinnt sie wieder, wenn sie eine bestimmte Menge Katzenfutter frisst – je mehr sie davon bekommt, desto besser kommt sie wieder zu Kräften.

Aufgabe 1: Zeichne ein Klassendiagramm für die Klasse `Mausekatze`.

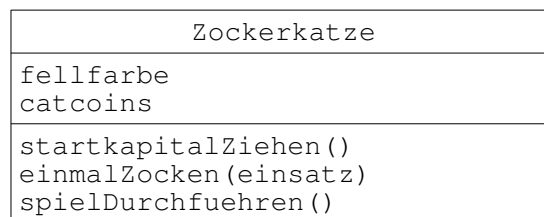
6 BE

Szenario: Es gibt nicht nur Mausekatzen, die durch die Jagd auf Mäuse Getreidevorräte schützen, es gibt auch Zockerkatzen, die man ins Casino schicken kann. Dort können sie im Glücksspiel erhebliche Geldbeträge gewinnen – oder verlieren. Das Glücksspiel, das im Casino gespielt wird, funktioniert so:

a) Zu Beginn erhält eine Zockerkatze sogenannte Catcoins, die als Einsatz verwendet werden können, als Startkapital. Sie lässt sich nämlich so lange einzelne zufällig ausgewählte Münzen im Wert von einem, fünf oder zehn Catcoins geben, bis der Wert von 30 Catcoins überschritten ist.

b) Jetzt kann die Zockerkatze zocken (solange sie noch Catcoins hat und noch weiterzocken möchte). Einmal zocken bedeutet, dass sie eine Spielkarte von einem Kartenstapel zieht. Dabei muss angegeben werden, wie viele ihrer Catcoins sie darauf setzen will, dass sie keine Hundekarte zieht. Auf jeder Karte ist nämlich entweder eine Maus oder ein Hund abgebildet. Zieht die Zockerkatze eine Mauskarte, wird ihr Catcoin-Guthaben um das Dreifache ihres Einsatzes erhöht. Zieht sie jedoch eine Hundekarte, sinkt ihr Catcoin-Guthaben auf 0. (Das Spiel ist dann vorbei.)

Das folgende Klassendiagramm wurde für das Spiel entworfen:



Die Fellfarbe, eine Zeichenkette, soll bei der Erzeugung eines Objekts der Klasse `Zockerkatze` – als Parameter des Konstruktors – angegeben werden. Das Catcoin-Guthaben hingegen muss bei der Erzeugung eines Objekts zunächst auf 0 gesetzt werden – es darf also kein Parameter des Konstruktors sein.

Die `spielDurchfuehren`-Methode führt das gesamte Spiel durch; unter anderem ruft sie bei Bedarf die anderen beiden Methoden auf. Wir betrachten sie aber hier nicht genauer!

Die `startkapitalZiehen`-Methode erledigt das oben unter a) Genannte. (Wie das „Münze-Geben-Lassen“ im Einzelnen funktioniert ist hier nicht von Bedeutung.) Wichtig ist dabei, dass – um es spannend zu machen – bei jeder neuen Münze angezeigt wird, was es für eine Münze war. Am Schluss (also wenn die 30 Catcoins überschritten sind) wird angezeigt, wie viele Catcoins die Zockerkatze jetzt insgesamt hat.

Die `einmalZocken`-Methode erledigt das oben unter b) Angegebene: Das Ziehen der Karte und das Aktualisieren des Catcoin-Guthabens. (Wir gehen davon aus, dass die `spielDurchfuehren`-Methode den als aktuellen Parameter übergebenen Einsatz vom Spieler erfragt hat.) Das Ziehen der Karte geschieht, indem von einem Objekt mit dem Bezeichner `Kartenstapel` die Methode `karteGeben()` aufgerufen wird. Diese Methode gibt einen Wahrheitswert zurück: `true`, wenn es eine Mauskarte ist, `false`, wenn es eine Hundekarte ist. (Wir gehen davon aus, dass das `Kartenstapel`-Objekt mit einer entsprechenden `import`-Anweisung bereitgestellt worden ist.)

Aufgabe 2: Zeichne ein Aktivitätsdiagramm, das darstellt, wie die `startkapitalZiehen`-Methode arbeiten muss.

7,5 BE

Aufgabe 3: Gib eine Implementierung der Klasse `Zockerkatze` in der Programmiersprache Kotlin ohne die `startkapitalZiehen`-Methode und ohne die `spielDurchfuehren`-Methode an. Berücksichtige alle Anforderungen aus dem Szenario, insbesondere hinsichtlich der Deklaration der Attribute.

8,5 BE