

Beleg 2: A - Bauernhof

Aufgabe

In einem Prototyp für Bauernhofspiel gibt es drei Arten von Gegenständen: Gebäude, Fahrzeuge und Grundstücke. Die Gebäude unterteilen sich weiterhin in Wohngebäude, Ställe und Wirtschaftsgebäude. Wohngebäude und Wirtschaftsgebäude haben Blitzableiter. An Fahrzeugen gibt es Traktoren, Mähdrescher und Transporter. Grundstücke werden unterteilt in Wald-, Wiesen- und Feldgrundstücke. Jeder Gegenstand besitzt eine eindeutige ID beginnend bei 5000 für Gebäude, beginnend bei 8000 für Fahrzeuge und beginnend bei 1000 für Grundstücke. Bleiben Sie offen für Erweiterungen, da im Verlauf der Entwicklung des Spiels garantiert weitere Gegenstände hinzukommen.

Diese Gegenstände können in spezialisierten Fachabteilungen erworben werden. Fahrzeuge gibt es im Landmaschinenhandel, Gebäude bei der Bauinnung und Grundstücke beim Landwirtschaftsverband. Nutzen Sie das Factory-Method Pattern für die Erzeugung der Gegenstände.

Alle Gegenstände besitzen eine Position, an der sie sich gerade befinden. Das sind in dieser Variante x,y Koordinaten. Gehen Sie davon aus, dass sich die Art und Weise, wie die Position gespeichert wird, ändert. Bleiben Sie bei der Implementation der Position also flexibel und nutzen Sie OO-Techniken zu Ihrem Vorteil.

Erstellen Sie ein 5x5 Felder großes Spielraster, auf dem Sie die Gegenstände platzieren können. Das Raster besitzt in x-Richtung Buchstaben zur Kennung und in y-Richtung Zahlen, beginnend bei 1 (z.B. A1,C4 usw.) Jeder Gegenstand nimmt genau ein Feld ein. Auf jedem Feld darf nur ein Gebäude stehen, aber nicht auf Grundstücken wie Wiese, Wohngebäuden und Ställen.

In der Mitte des Spielfelds steht ein großer Felsen, dieser ist unverrückbar, nicht befahrbar für Fahrzeuge und nicht überbaubar. Allerdings kann er durch Regen nass werden und vom Blitz getroffen werden.

Stellen Sie programmtechnisch sicher, dass alle Vorgaben für Gegenstände und Spielraster durch den Compiler überprüft werden können und eingehalten werden.

Führen Sie einen Programmablauf durch, der alle Felder des Spielrasters zufällig belegt und geben Sie das Ergebnis im Terminal aus. Lassen sie es regnen, alle Fahrzeuge, die nicht in einem Gebäude stehen werden nass. Geben Sie die betroffenen Fahrzeuge im Terminal durch eine geeignete Ausschrift aus.

Simulieren Sie zu dem Regen ein Gewitter. Mit einer Wahrscheinlichkeit von 5% werden Gebäude ohne Blitzableiter vom Blitz getroffen (auch der Felsen!). Lassen Sie es solange blitzen, bis ein Gebäude vom Blitz getroffen wurde. Geben Sie das Ergebnis im Terminal aus.

Abnahmekriterien:

- Erzeugen Sie zur Laufzeit, geeignete Ausgaben im Terminal, um die Korrektheit und Vollständigkeit des Programms zu unterstreichen. Bereiten Sie sich auf Nachfragen zum source-Code vor. Erklären Sie anhand Ihres Klassendiagramms die Implementation.
- Klassendiagramm (*.png,*.jpg,*.pdf) -> in Opal hochladen (1 Datei)
- zwingend: Objektorientiertes Programm, keine GUI. Prozedurale Programme werden nicht mehr anerkannt. (Tipp: Es gibt mehr als 5 Klassen)
- Clean Code(I): kurze Methoden (20 - 30 Zeilen (Es gibt Ausnahmen!))
- Clean Code(II): max. Einrückung pro Methode: 2 (3 - begründen)
- Clean Code(III): aussagekräftige Variablen- und Methodennamen