**JAVA Aufgabenblatt 6**

**A6.1)**

Implementiere eine virtuelle Welt, in der verschiedene Objekte vorkommen und durch bestimmte Ereignisse Veränderung erfahren.

**Teil 1)**

In der virtuellen Welt gibt es folgenden Klassen:

|  |  |  |
| --- | --- | --- |
| **Klasse** | **Attribute** | **Methoden** |
| Blume | bezeichnung (String)  farbe (String)  wohlfuehltemperatur (int)  wasserstand (int)  wasserstandMax (int, maximal erlaubter Wasserstand)  wasserstandMin (int, minimal erlaubter Wasserstand) | void verliereBlaetter()  nur Ausgabe |
| Baum | bezeichnung (String)  wohlfuehltemperatur (int)  wasserstand (int)  wasserstandMax (int, maximal erlaubter Wasserstand)  wasserstandMin (int, minimal erlaubter Wasserstand) | void verliereBlaetter()  nur Ausgabe |
| Feuerwehrmann | vorname (String)  nachname (String)  aufgabenbereich (String) | void wechsleKleidung()  nur Ausgabe |
| Urlauber | vorname (String)  nachname (String)  heimatland (String)  badewettertemp (int) | void wechsleKleidung()  void geheBaden()  void fahreHeim()  nur Ausgabe |
| Polizist | vorname (String)  nachname (String)  dienstgrad (String) | void wechsleKleidung()  nur Ausgabe |

* Überlege welche der Klassen du jeweils sinnvoll in einer Klassenhierarchie (über Superklassen) zusammenfassen kannst.
* Implementiere anschließend das Klassengerüst unter Berücksichtigung der folgenden Kriterien:
  + sämtliche Klassen kommen in ein gemeinsames Package
  + Es dürfen nur von den hier angeführten Klassen Objektinstanzen erzeugt werden (also nicht von den Superklassen)!
  + keine der Klassen/Methoden darf außerhalb des Projekts sichtbar sein!
  + keines der Attribute darf außerhalb seiner eigenen Klasse sichtbar sein!
  + Keines der Attribute (ausgenommen Wasserstand) darf nach dem Initialisieren verändert werden! (setze eine entsprechende Maßnahme!)
  + Konstruktoren müssen vorhanden sein / Getter & Setter nur sofern notwendig

**Teil 2)**

Implementiere nun das Verhalten der einzelnen Klassen bei folgenden Events

|  |  |  |
| --- | --- | --- |
| **Klasse / Event** | **Regen mit bestimmter Menge** | **Temperaturaenderung in Grad** |
| Blume | verändert den Wasserstand der Blume nach oben. Wasserstand zu hoch/niedrig 🡪 Blume verliert Blätter\* | ist die Temperaturänderung zu hoch (Wohlfühltemperatur wird überschritten), dann sinkt der Wasserstand der Blume entsprechend. Wasserstand zu niedrig 🡪 Blume verliert Blätter!\* |
| Baum | verändert den Wasserstand des Baumes nach oben. Wasserstand zu hoch/niedrig 🡪 Baum verliert Blätter\* | ist die Temperaturänderung zu hoch (Wohlfühltemperatur wird überschritten), dann sinkt der Wasserstand des Baumes entsprechend. Wasserstand zu niedrig 🡪 Baum verliert Blätter!\* |
| Feuerwehrmann | wechselt Kleidung | wechselt Kleidung |
| Urlauber | fährt nach Hause | wenn Temperatur hoch genug (was immer das heißt für einen Urlauber), dann geht ein Urlauber baden |
| Polizist | wechselt Kleidung | wechselt Kleidung |

* Jede Methode muss auf der Konsole eine sinnvolle Ausgabe erzeugen.
* Überlege dir außerdem einen eigenen kleinen Algorithmus, der für die mit \* markierten Methoden den Wasserstand sinnvoll anpasst. Der Wasserstand kann nie <0 sein!
* Lege um dein Programm testen zu können eine neue Klasse mit folgenden Kriterien an:
  + diese Klasse ist public
  + enthält die main (rufe in der main die Events auf – entweder direkt oder indirekt)
* Teste alle Klassen einzeln ob sie entsprechend der Angabe funktionieren!
* Kommentare nicht vergessen!