

Übungen zur Vererbung

ERWARTUNGSHORIZONT

Szenario

Ergänzen Sie in jedem Fall fehlende Getter, Setter.

1 Bienen

... sind Tiere, Arbeiter-/Honig-/Malerbienen sind Bienen. Wenden Sie das Prinzip der Spezialisierung an und ändern Sie die Klassen entsprechend.

- Biene extends Tier
- Arbeiter/Honig/Maler extends Biene
- Arbeiter/Honig/Maler bekommen Konstruktor mit Parameter spezies und geben diesen per super nach oben
- Attribut spezies:String in Biene muss protected sein (damit es nach unten vererbt werden kann)
- Konstruktor von Biene: Textausgabe „Biene wurde erzeugt“ (muss nicht in jeder Unterklasse einzeln stehen; wenn Sie 100 Unterklassen hätten, müssten Sie 100 Mal die gleiche Textausgabe schreiben und bei Fehlern 100 Mal ändern)

Entfernen Sie dabei auch alle überflüssigen Attribute, Methoden etc.

- Arbeiter/Honig/Maler – Textausgabe aus dem Konstruktor entfernen (weil schon in der Klasse Biene vorhanden)
- Arbeiter/Honig/Maler – Getter/Setter für spezies entfernen; summen() entfernen (weil schon in der Klasse Biene vorhanden, wird auf die Unterklassen vererbt)

2 Bluete

Reparieren Sie die Fehler in den Blüten.

- Bluete: Getter/Setter für anzahlPollen
- Mohn: Braucht einen Konstruktor mit den Parametern name:String, anzahlPollen:int und gibt diese beiden Werte per super nach oben zur Bluete.
- Butterblume: Braucht als zusätzliche Konstruktorparameter name:String, anzahlPollen:int und gibt diese beiden Werte per super nach oben zur Bluete
- Butterblume: Getter/Setter für buttrigkeit
- Achtung: buttrigkeit in Butterblume BLEIBT private

3 Blumen

Alle Blumen haben ein Attribut geruch:float.

- in Blume Attribut als protected einfügen, Getter+Setter erstellen

Alle Rosen haben ein Attribut romantikfaktor:int.

Alle Rosen haben ein Attribut meinGiesser:Gaertner

- in Rose Attribute als protected einfügen, Getter und Setter erstellen (NUR in Klasse Rose; Unterklassen erben das)

Diese Attribute werden jeweils über einen parametrisierten Konstruktor initialisiert.

- Blume: Konstruktor mit Parameter geruch einfügen; geruch im Konstruktor initialisieren
- Rose: Konstruktor mit Parametern geruch, meinGiesser, romantikfaktor einfügen; geruch wird per super nach oben gegeben, meinGiesser/geruch wird im Konstruktor initialisiert
- Rose-Kindklassen: Konstruktor mit Parameter geruch, meinGiesser, romantikfaktor einfügen; alle drei Werte werden per super nach oben gegeben.

Alle Blumen haben eine Methode riechen():int

- Blume: Methode riechen (NUR in blume; alle Unterklassen erben das). Rückgabewert int nicht vergessen

Alle Rosen haben eine Methode stechen(person:Person):void

- Rose: Methode stechen mit entsprechendem Parameter einfügen

4 Klassendiagramm

Erstellen Sie ein Klassendiagramm zum Szenario (Violet UML)