Klassenextension

Dieser Artikel erfüllt die GlossarWiki-Qualitätsanforderungen **nur teilweise**:

Korrektheit: 3 (zu größeren Teilen überprüft)	**Umfang**: 4 (unwichtige Fakten fehlen)	**Quellenangaben**: 1 (fehlen großteils)	**Quellenarten**: 4 (sehr gut)	**Konformität**: 5 (ausgezeichnet)

Inhaltsverzeichnis

1 Definition (W. Kowarschick[1])

Die zu einer Klasse gehörende Klassenextension ist eine (i. Allg. endliche) Menge, die zu jedem Zeitpunkt genau diejenigen Objekte, die der Klasse zugeordnet sind, enthält.

2 Bemerkungen

Der Begriff der Klassenextension ist zunächst einmal für die formales Definition des Begriffes Klasse von großem Vorteil. Aber auch für die Implementierung von Programmsystemen und Anwendungen ist dieser Begriff wichtiger, als man zunächst meinen könnte, da die meisten objektorientierte Sprachen keinen direkten Zugriff auf die Klassenextension realisieren. Viele objektorientierte Programmiersprachen verwalten die Klassenextension dennoch explizit (ohne dem Programmierer einen direkten Zugang zu gewähren) — zum Beispiel, damit ein Garbage Collector korrekt arbeiten kann.

Objektorientierte Datenbanksysteme machen die Klassenextension dagegen immer allen denjenigen Benutzern zugänglich, die die entsprechenden Rechte haben. In derartigen Systemen wird an Stelle des Begriffs Klassenextension üblicherweise der Begriff Tabelle verwendet. Die dauerhafte Speicherung von Daten in Tabellen ist ja gerade die wesentliche Aufgabe eines jeden Datenbanksystems.

Aber auch in Anwendungen, wie z.B. bei der Programmierung von Computerspielen, ist es häufig wichtig, dass der Programmierer einen direkten Zugang zur Extension einer Klasse hat. Wenn der Programmierer beispielsweise direkt auf die Extension der Klasse `Combatant` der Spielgegner zugreifen kann, kann er diesen bei Spielende die Nachricht zukommen lassen, dass sie sich alle auflösen oder sonstwie die Bühne verlassen.

Der direkte Zugriff auf die Extension einer Klasse s kann auf mehrere Arten relativ einfach realisiert

werden:

Mittels Klassenmethoden (siehe nachfolgendes Beispiel).
Mittels einer Singleton-Klasse (z.B. `Extension`), die den Zugriff auf Klassenextensionen über eine Methode `extension(p_class: Class): Dictionary` ermöglicht.
Mittels einer eigenen Singleton-Klasse für jede Klasse, auf deren Extension man zugreifen möchte (z.B. `CombatantExtension` oder auch `Extension<Combatant>`).

2.1 Beispiel

Im folgenden Beispiel wird die Extension der Klasse `Combatant` in einem statischen Attribut `extension` (der Art `Dictionary`) zur Verfügung gestellt. Jedes Mal, wenn ein neues Objekt erzeugt wird, fügt der Konstruktor dieses neue Objekt automatisch in die Extension ein.

```
public class Combatant
{
  private static var sv_extension: Dictionary = new Dictionary();
  //besser wäre: Dictionary<Combatant>; dies wird aber von Actionscript 3
nicht unterstützt

  public static function get extension(): Dictionary
  {
    return sv_extension;
  }

  public function Combatant(...)
  {
    sv_extension[this] = this;
    ...
  }
}
```

Man beachte, dass `extension` nicht als Array, sondern als Dictionary realisert wurde, damit das Löschen von Elementen aus der Extension besonders effizient implementiert werden kann. Diese Löschen kann und sollte in Sprachen wie C++ von Destruktor übernommen werden. In Sprachen mit Garbage Collector (wie ActionScript oder Java) müssen dagegen eigene Delete-Operationen implementiert werden.

Vorzugsweise sollte eine Objektmethode `delete` definiert werden, die jeweils auf das zu löschende Objekt angewendet wird:

```
  public function delete(): void
  {
    delete sv_extension[this];
  }
```

Aber auch der Einsatz einer Klassenmethode (statische Methode) ist möglich:

```
public static function delete(p_combatant: Combatant): void
{
  delete sv_extension[combatant];
}
```

In beiden Fällen muss der Programmierer jedoch zusätzlich dafür Sorge tragen, dass auch alle übrigen Verweise auf das zu löschende Objekt entfernt werden, damit der Garbage Collector das Objekt auch wirklich löschen kann.

2.2 Subklassen

Laut Definition ist die Extension einer Subklasse B Teilmenge der Extension der zugehörigen Superklasse A. Insbesondere besteht die Extension einer Abstrakten Klasse oder eines Interfaces aus der Vereinigung der Extensionen aller nicht-abstrakten Subklassen, d.h. aller Subklassen, die eine eigenene Extension haben.

Die Implementierung der Extensionsverwaltung einer Klassenhierarchie ist nicht wesentlich aufwendiger, als die Implementierung der Extensionsverwaltung einer einzelnen Klasse. Falls man die Extension eines Interfaces benötigt, kann man die oben vorgestellte Extensionverwaltung auf Basis von Klassenmethoden nicht verwenden, da Interfaces in der Regel keine statischen Methoden unterstützen. Hier muss man eine der anderen zuvor genannten Alternativen wählen.

3 Quellen

Kowarschick (MMProg): Wolfgang Kowarschick; Vorlesung „Multimedia-Programmierung“; Hochschule: Hochschule Augsburg; Adresse: Augsburg; Web-Link; 2018; Quellengüte: 3 (Vorlesung)
Kowarschick (2002a): Wolfgang Kowarschick; Multimedia-Programmierung – Objektorientierte Grundlagen; Hrsg.: Michael Lutz und Christian Märtin; Reihe: Informatik interaktiv; Verlag: Fachbuchverlag Leipzig im Carl Hanser Verlag; ISBN: 3446217002; 2002; Quellengüte: 5 (Buch)

Kategorien:
Objektorientierte Programmierung
Glossar
Kapitel:Multimedia-Programmierung