



Informatik I – Eprog HS12

Übung 10

1 Aufgabe: OOP

1.1 Lernziele

1. Vertiefung der OO Abbildung einer Problemstellung.
2. Vererbungsstrukturen erkennen.

1.2 Ausgangslage

- Paul ist ein 13-jähriger Junge. Natürlich kann er Lesen und Schreiben. Am liebsten isst er Pizza. Wie alle Menschen hat auch er eine Oma. Sein Hobby ist Fussball spielen.
- Miezi ist ein junges Kätzchen. Sie gehört Oma Hugentobler und frisst am liebsten Katzi-Katzenfutter. Ihr Fell ist getigert. Wenn sie einen Laut von sich gibt, dann miaut sie.
- Lupo ist bereits 4 Jahre alt. Lupo, der Hund, gehört zu Paul. Wurst frisst er für sein Leben gern. Gibt er einen Laut von sich, dann bellt er. Sein Fell ist braun.
- Oma Hugentobler ist 81 Jahre alt. Paul ist ihr Enkel. Lesen und Schreiben kann sie schon seit langem! Ein gutes Stück Apfelkuchen kann sie nie ablehnen. Lesen ist ihr grösstes Hobby.
- Herr Schulze ist 62 Jahre alt. Er ist kein Opa, denn Kinder oder Enkel hat er nicht. Auf Arbeit muss er viel lesen und schreiben, daher verbringt er seine Freizeit lieber mit Schwimmen. Er isst gern Kuchen.
- Wenn Menschen Geräusche von sich geben, sind das meistens Worte.

1.3 Aufgabenstellung

Identifizieren Sie basierend auf der Ausgangslage (abstrakte) Klassen, Attribute und Methoden. Bilden Sie aus den zu den Objekten gehörenden Klassen eine Vererbungshierarchie, indem Sie Gemeinsames und Verwandtes in eine gemeinsame «Elternklasse» verschieben. Nutzen Sie keine Interfaces für diese Übung. Dies ist eine Modellierungsübung. Sie brauchen daher keine Getter/Setter/Konstruktoren implementieren! Es reichen die Klassengerüste mit nötigen Attributen und sonstigen Methoden.

2 Aufgabe: Nachrichtensystem

2.1 Lernziele

1. Anwendung von Interfaces üben.

a) Aufgabenbeschreibung

Sie sollen eine Art zentrale Mailingliste für ein Unternehmen entwickeln. An diese Liste können alle Mitarbeiter Emails senden. Emails besitzen einen Betreff, einen Absender und sowie einen Inhalt. Emails werden anhand ihres Betreffs Kategorisiert. Eine Konvention schreibt vor, dass jeder Betreff mit einem Schlagwort in eckigen Klammern zu beginnen hat. Mögliche Schlagworte sind: Business, IT, News und Sonstiges. Mails ohne Schlagwort bekommen die Kategorie Sonstiges.

Ein Beispiel Betreff sieht wie folgt aus: [IT] Tipps zum Umgang mit Spam Mail

Postfächer können sich für Emails einer oder mehrerer Kategorie registrieren. Sie bekommen dann alle Mails der entsprechenden Kategorie weitergeleitet. Das gilt auch für Nachrichten die schon im Pool sind. Später sollen Nachrichten auch an andere Empfänger als Postfächer weitergeleitet werden.

Ein zentrales Mailsystem empfängt die Emails. Um Spam zu minimieren wird jede Mail zuerst durch eine Filterkette geleitet. Jeder Filter analysiert die Emails anhand unterschiedlicher Kriterien. Ein Filter kann eine Mail ablehnen. In diesem Fall wird sie nicht weiter verarbeitet. Die Anwendung soll einen Stopwort Filter bekommen. Dieser enthält eine Liste von kritische Begriffen welche entsprechend der Unternehmenskultur nicht akzeptiert werden. Ein Absenderfilter sorgt dafür, dass nur Emails aus der Domain des Unternehmens - someCompany.com - akzeptiert werden.

Sobald eine Mail alle Filter überstanden hat, folgt eine Verarbeitungskette. Hier wird die Mail verarbeitet und aufbereitet. Die Mailingliste soll nur Plaintext Emails speichern. Entsprechend werden in einem ersten Verarbeitungsschritt alle - falls vorhanden - HTML Tags aus dem Mailinhalt entfernt. In einem weiteren Verarbeitungsschritt wird die Kategorie aus dem Betreff der Mail extrahiert.

Die Mail wird nun in einem Nachrichtenpool gespeichert. Der Pool überprüft die Kategorie der Nachricht und schick die Mail an alle Postfächer, welche sich für die entsprechende Kategorie registriert haben. Dabei soll jedes Postfach aber nur eine Kopie der Nachricht erhalten.

Eine Skizze des Ablaufes finden Sie unten.

Implementierungshinweise:

1. Nutzen Sie Interfaces. Zum Beispiel sollten Sie Interfaces für die Elemente der Verarbeitungskette sowie die Filter nutzen. Auch sollte die Interaktion zwischen Postfächern und dem Nachrichtenpool über Interfaces ablaufen.
2. Eine mögliche Lösungen könnte auf Interfaces mit folgenden Bezeichnungen beruhen: IFilter, IProcessor, IMessageSource, IMessageReceiver
3. Sie brauchen keine tiefen HTML Kenntnisse. Entfernen Sie einfach alle Ausdrücke der Art "< .. >". Daher zum Beispiel "
"

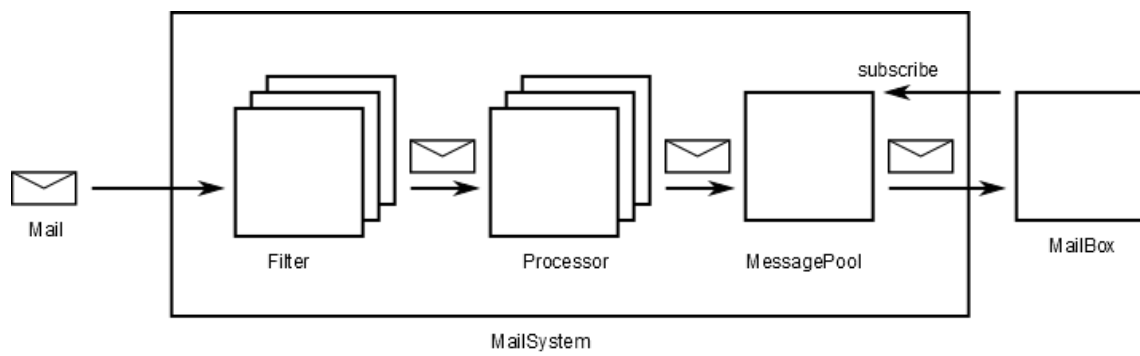


Figure 1: Schematischer Ablauf

3 Aufgabe: Rekursion

3.1 Lernziele

1. Rekursive Programmierung üben

3.2 Aufgabenstellung

Schreiben Sie je eine rekursive Funktion zur Lösung der folgenden Aufgaben.

- Invertieren eines Strings. Beispiel: `invert("Test") -> tseT`
- Berechnung der Fakultät. Beispiel: `fac(4) -> 1 x 2 x 3 x 4 -> 24`
- Grösster gemeinsamer Teiler. Beispiel: `gcd(4,8) -> 4`