

Hausaufgabe 3: Übersicht

- Erweitert das Composite Pattern für Files vom letzten Mal auf ein Visitor Pattern.
- Zählt Dateien mit Hilfe eines *CountFilesVisitor*, der sich die Gesamtanzahl merkt (im Junit-Test prüfen)!
- Baut einen weiteren Visitor zur Summierung der Dateigrößen aller Dateien auf (*SummarizeFileSizesVisitor*)
- Baut einen Visitor der alle Dateien zählt, auf deren Namen ein mitgelieferter regulärer Ausdruck matcht (z.B. alle class-Dateien in lowercase-Notation)

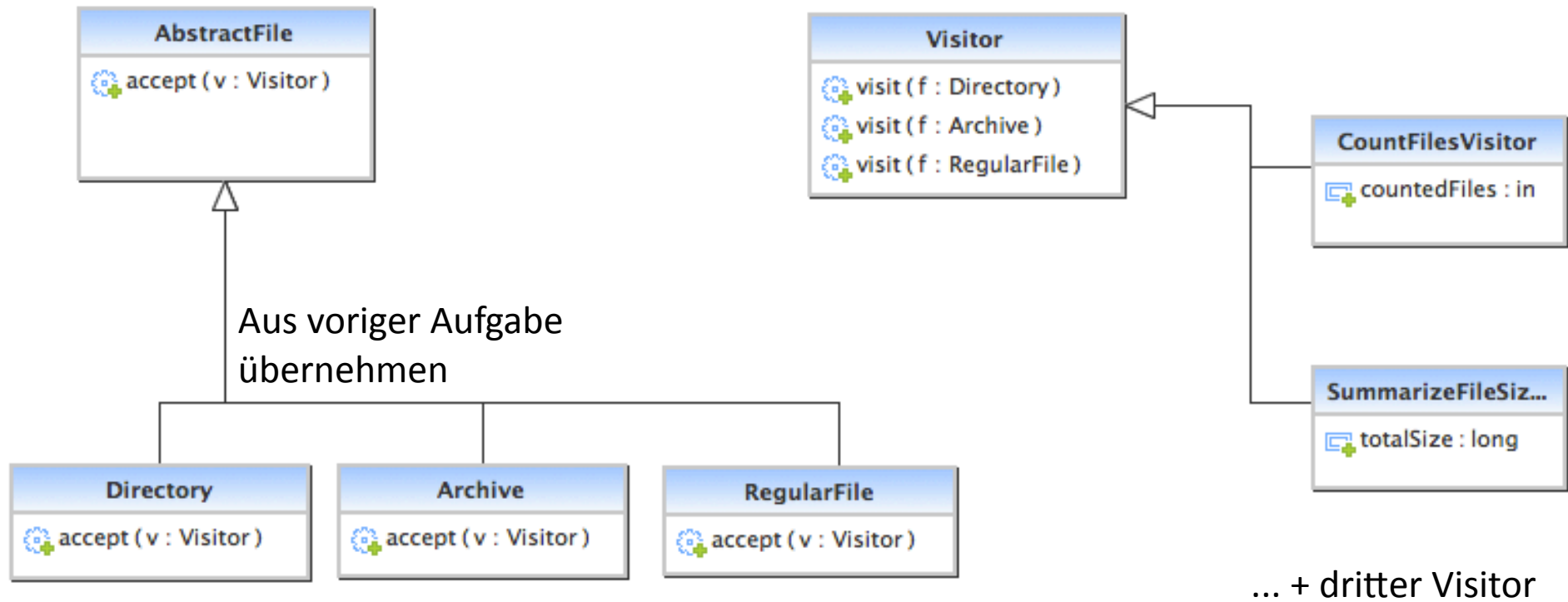
Hinweise:

- Verwendet das Projekt der letzten Hausaufgabe (Dateisystem mit Composite-Pattern) weiter:
 - Die Unterklassen von (Abstract)File müssen also um passende accept()-Methoden ergänzt werden.
 - Die traversierende Methode muss einen Visitor als Parameter bekommen!
- Es sollen sich drei Visitor-Unterklassen im Projekt befinden.

Bonusteil:

- Implementiere ein generisches accept() mit Reflection, wie in der Vorlesung gezeigt

Klassendiagramm



Abgabe:

- Bis 10.5.2011, 15:00 Uhr
- Abgabe per Mail an ruben.jubeh@uni-kassel.de
- Betreff: DesignPatterns_HA1_<Matrikelnr>
- .zip im Anhang (Siehe nächste Folie)
- Das Projekt soll einen(!) JUnit Test enthalten (Start mit einem Click!)

.zip Format

- Gezippter Workspace (!) mit nur einem(!) Projekt drin und natürlich in Quellcode (Es empfiehlt sich, mit einem leeren Workspace anzufangen)

Hausaufgabenbewertung

- 2 Hausaufgaben können ausgelassen werden (n-2 zählen)
- 0-10 Punkte je HA, 50% reicht zum Bestehen
- Einsicht während der Übungszeit (Di 17-19 Uhr), ab nächster Woche.