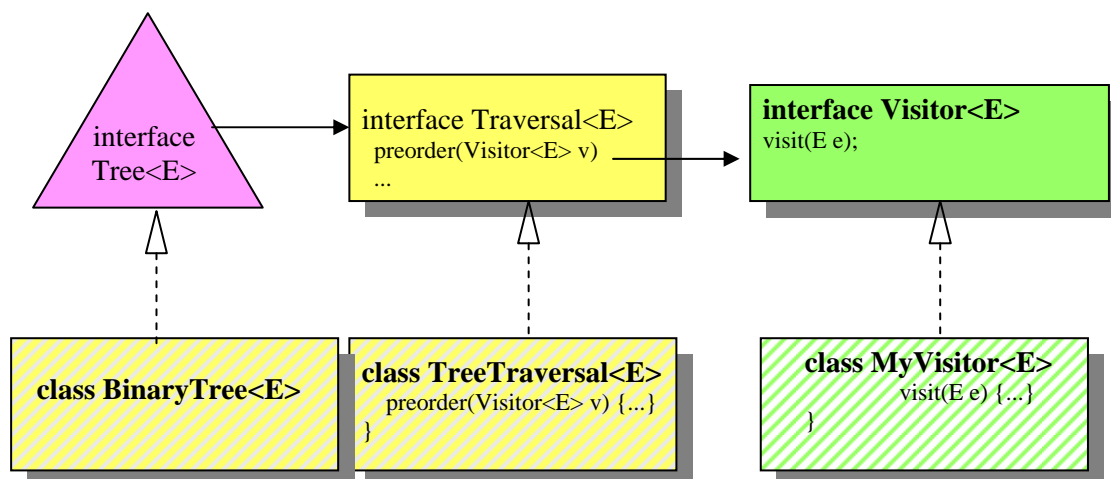


Praktikum 6:  
Rangliste mit Baum

Die Verwaltung von Datensätzen mittels Listen ist nur bis zu einer kleinen Anzahl Elemente effizient, da zum Finden und geordneten Einfügen eines Elements im Schnitt über die Hälfte der Elemente hinweg iteriert werden muss. Aus diesem Grund werden bei grösseren Datenmengen effizientere Datenstrukturen wie zum Beispiel sortierte Binärbäume vorgezogen. Sortierte Binärbäume haben die Eigenschaft, dass Elemente im Mittel in  $\log_2$  Schritten eingefügt und gefunden werden können. Es soll nun die Aufgabe des Praktikums 5 nochmals mittels eines sortierten Binärbaumes implementiert werden. Das heisst, die Teilnehmer des Zürcher Marathons 2003 sollen sortiert nach ihrem Rang ausgegeben werden.

**Aufgabe 1**

Implementieren Sie für diesen Zweck die Interfaces `Tree`, `Traversal` und `Visitor`. Zusätzlich muss ein `RankingTreeServer` erstellt werden, der das `CommandInterpreter` Interface implementiert.

## Hinweise:

### BinaryTree

- Die Klasse `BinaryTree` liegt bereits vor. (Einfügen und Löschen naiv, d.h. der Baum ist nicht balanciert!) Lediglich die Methode `traversal()` muss noch vervollständigt werden. Es muss dabei eine Instanz einer das `Traversal` Interface implementierenden Klasse erzeugt und zurückgegeben werden.

### TreeTraversal

- Überlegen Sie sich, welche Travesierungsart(en) Sie für die Aufgabe benötigen. Für die nicht implementierten Methoden können Sie die `UnsupportedOperationException` werfen.

### MyVisitor

- Überlegen Sie sich, wie Sie etwas in Ihrer Klasse ausgeben können, bzw. wie Sie den String aufbauen können, den Sie als Rückgabewert für die `interpret` Methode des `RankingTreeServer` benötigen.

### RankingTreeServer

- Implementierten Sie eine Klasse `RankingTreeServer`, die den `CommandInterpreter` implementiert – wie gehabt. Sie können das Einlesen der Teilnehmerliste aus dem Praktikum 3 übernehmen.

## Aufgabe 2

Bauen Sie eine `PriorityQueue` auf dem sortierten Binärbaum auf und erstellen Sie einen `RankingQueueServer`.

Hinweis: Priorität kann als Sortierkriterium genommen werden.