



Name: _____

Abiturprüfung 2009

Informatik, Leistungskurs

Aufgabenstellung:

Justus betreibt Ahnenforschung. Zur Verwaltung der Daten seiner Vorfahren stellt er die Personen generationsweise in einem Binärbaum dar, wobei die Mütter immer links und die Väter immer rechts stehen.

Folgende Ergebnisse der Ahnenforschung liegen für Justus bereits vor:

Vater von Justus:	Henry
Mutter von Justus:	Margit
Vater von Henry:	Heinz
Mutter von Henry:	Lotte
Vater von Margit:	Werner
Mutter von Margit:	Antonie
Mutter von Werner:	Elisabeth

- a) Stellen Sie die Baumstruktur zum obigen Beispiel in geeigneter Form graphisch dar.

Durch weitere Recherche konnte festgestellt werden, dass die beiden Väter von Werner und Antonie Robert und Bernhard hießen. Bernhard war der Ehemann von Elisabeth. Erweitern Sie den Baum entsprechend.

Die Mutter von Heinz, also Justus' Urgroßmutter väterlicherseits, hieß ebenfalls Lotte.

Erweitern Sie den Baum entsprechend. (6 Punkte)

- b) Man möchte natürlich nicht nur die Vornamen der Vorfahren speichern, sondern auch weitere Informationen wie deren Nachname, Geschlecht und deren Geburtsdatum.

Entwerfen Sie eine Klasse Ahne.

Die gesamte Ahnenverwaltung soll in der Klasse AhnenVerwaltung realisiert werden.

Entwerfen Sie ein Klassendiagramm mit den Klassen AhnenVerwaltung, Ahne und BinTree. Die Methoden von AhnenVerwaltung und BinTree brauchen nicht aufgeführt zu werden. (7 Punkte)



Name: _____

- c) Mit einer Zeichenkette kann man die Vorfahrenlinie, also den Weg durch den Baum, charakterisieren (z. B. Robert bezüglich Justus: Mutter, Mutter, Vater, abgekürzt – ,mmv’).

(1) Geben Sie die Zeichenkette für Elisabeth bezüglich Justus an.

(2) Entwerfen Sie eine Strategie zur Implementierung einer Methode

```
public void fuegeHinzu(Ahne pAhne, String pLinie),
```

die eine neue Person in den Baum einfügt. Dabei soll pLinie die Vorfahrenlinie enthalten. Der Baum soll in einem Attribut der Ahnenverwaltung mit dem Namen hatAhnenbaum vorliegen.

Sie können davon ausgehen, dass der Baum mindestens die Wurzel „Justus“ enthält und pLinie zum Baum passt, d. h., eine Fehlerbehandlung ist nicht erforderlich.

(3) Implementieren Sie die Methode und eventuelle Hilfsmethoden vollständig.

(15 Punkte)

- d) Sucht man nun in einem großen Stammbaum eine Person, die eindeutig durch die Kombination aus ihrem Vornamen, ihrem Nachnamen und ihrem Geburtsdatum unter allen Personen identifizierbar ist, so möchte man unter anderem wissen, in welcher Generation die gesuchte Person gelebt hat (für diese Aufgabe gilt: beim Eingangsbeispiel lebt Justus in der ersten Generation, leben Henry und Margit in der zweiten Generation ...). Die Klasse Ahnenverwaltung wird dazu um eine Anfrage ergänzt, die folgendermaßen beginnen soll:

```
public int ermittleGeneration(String pName,  
                             String pVorname, String pGebDatum)
```

Erläutern Sie eine Strategie zur Bestimmung der Generation. Sie können weitere Hilfsmethoden definieren. Der Baum soll in einem Attribut der Ahnenverwaltung mit dem Namen hatAhnenbaum vorliegen. Sie können voraussetzen, dass der Baum mindestens „Justus“ enthält. Ist ein Name nicht vorhanden, so soll 0 zurückgegeben werden.

Implementieren Sie die Methode ermittleGeneration(..) und eventuelle Hilfsmethoden vollständig.

(12 Punkte)

- e) Gegeben ist die Methode wasTueIch(..) (siehe unten).

Dokumentieren Sie den Ablauf der Abarbeitung der Methode unter Verwendung des Beispiels aus der Aufgabenstellung a).

Dabei genügt es, in einer Aufzählung die Namen als Vertreter der Bäume zu notieren.

Benutzen Sie das folgende Dokumentationsschema.



Name: _____

lQueue

Justus

Margit — Henry

lString

Justus

Der Baum soll in einem Attribut der Ahnenverwaltung mit dem Namen `hatAhnenbaum` vorliegen.

Analysieren Sie die folgende Methode `wasTueIch()` und beschreiben Sie ihre Funktionsweise.

```
1  public String wasTueIch() {
2      String lString = "";
3      Queue lQueue = new Queue();
4      lQueue.enqueue(hatAhnenbaum);
5      while (!lQueue.isEmpty()) {
6          BinTree lBaum = (BinTree)lQueue.front();
7          lQueue.dequeue();
8          Ahne lAhne = (Ahne)lBaum.getRootItem();
9          lString = lString + " " + lAhne.gibVorname();
10         if (!lBaum.getLeftTree().isEmpty()) {
11             lQueue.enqueue(lBaum.getLeftTree());
12         }
13         if (!lBaum.getRightTree().isEmpty()) {
14             lQueue.enqueue(lBaum.getRightTree());
15         }
16     }
17     return lString;
18 }
```

(10 Punkte)

Zugelassene Hilfsmittel:

- Wörterbuch zur deutschen Rechtschreibung
- Taschenrechner