



## Übungsblatt 8: Klassen und Objekte

Ausgabe: 4.12.17

Abgabe: 11.12.17 12:00 Uhr elektronisch mittels Git

### Aufgabe 1

Erstellen Sie eine Klasse `BigInt`, welche das Rechnen mit sehr großen natürlichen Zahlen (inklusive 0) ermöglicht.

Der Konstruktor der Klasse soll eine Zahl als `String` entgegennehmen und dann ziffernweise in einem Feld vom Typ `int[]` speichern. Besteht der Parameter lediglich aus Ziffern, dann wird das Array mit der nötigen Länge erzeugt (es soll lediglich so groß wie unbedingt notwendig sein) und die Zahl wird ziffernweise darin gespeichert. Bei ungültigem Parameter (keine natürliche Zahl im `String`) soll der Konstruktor eine `PRException` mit geeigneter Fehlermeldung auslösen. Beispiel:

```
BigInt zahl1 = new BigInt("23");
BigInt zahl2 = new BigInt("45");
```

Die Klasse soll über folgende Methoden verfügen:

- `void add( BigInt number )`

Addiert die als Parameter übergebene Zahl vom Typ `BigInt` zu der im Objekt vorliegenden Zahl hinzu. Das Ergebnis der Addition wird also anstelle der zuvor vorliegenden Zahl im Objekt gespeichert (so dass man damit weiterarbeiten kann). Das als Parameter übergebene Objekt bleibt unverändert. Beispiel:

```
zahl1.add(zahl2);
```

Tipp: Vergegenwärtigen Sie sich bei der Implementierung die schriftliche Addition.

- `String toString()`

Liefert die vorliegende Zahl als `String`. Beispiel:

```
println(zahl1.toString()); // gibt "68" auf der Konsole aus
println(zahl2.toString()); // gibt "45" auf der Konsole aus
```

- `int[] getDigits()`

Liefert das Feld in dem die Ziffern der vorliegenden Zahl gespeichert wurden. Beispiel:

```
println(zahl1.getDigits()[0]); // gibt "6" auf der Konsole aus
println(zahl1.getDigits()[1]); // gibt "8" auf der Konsole aus
```

- `int length()`

Liefert die Anzahl der Ziffern der vorliegenden Zahl. Beispiel:

```
println(zahl1.length()); // gibt "2" auf der Konsole aus
```

- `boolean equals(BigInt number)`

Liefert genau dann `true`, wenn beide Objekte die gleiche Zahl repräsentieren, sonst `false`. Beispiel:

```
println(zahl1.equals(zahl2)); // gibt "false" auf der Konsole aus
println(zahl1.equals(new BigInt("68"))); // gibt "true" auf der Konsole aus
```

- a) Erstellen Sie die Klasse `BigInt` wie oben beschrieben.
- b) Erstellen Sie in `BigInt` eine `main`-Methode, um Ihre Methoden zu testen.
- c) Angenommen Sie wissen, dass keine `BigInts` mit mehr als 50 Stellen benötigt werden. Überlegen Sie: Was müssten Sie an Ihrem Programm ändern, um immer mit Arrays der Länge 50 zu arbeiten?

## Allgemeines

- Die Aufgaben sind in Eclipse zu bearbeiten und beim Testat vorzuführen.
- Legen Sie für die Bearbeitung dieses Übungsblattes ein Paket namens `pr1.uebung08` an, in dem Sie Ihre Klassen anlegen.
- Erlaubt sind `MakeItSimple`-Funktionen (keine nicht besprochene Funktionalität aus der Java-Standard-Bibliothek) und das bisher erworbene Wissen aus den PR1-Vorlesungen. Sie müssen alle(!) von Ihnen verwendeten Konstrukte der Sprache sowie alle verwendeten Methoden, die nicht aus der Hilfsbibliothek `MakeItSimple` stammen, gut erklären und nötigenfalls im Testat selbst programmieren können!
- Sie geben ab, indem Sie vor Ende der Abgabefrist Ihr Projekt mit den lauffähigen Programmen des Übungsblattes in das Repository auf *ovid* pushen. Achten Sie darauf, ob Sie die Kontroll-E-Mail bekommen! Die letzte hochgeladene Version Ihres Projekts wird gewertet. Andere Abgaben, ob elektronisch oder auf Papier, zählen als **nicht abgegeben!**