Technische Universität Berlin **WS 2012/2013**
Fakultät II, Institut für Mathematik
Sekretariat MA 5–2, Frau Kiefer

Dr. Britta Peis
Martin Groß
Dr. Max Klimm
Madeleine Theile

Aylin Acikel, Katharina Bütow, Christian Döblin, Alexander Hopp, Daniel Kuske, Olivia Röhrig, Robert Rudow, Daniel Schmand, Hendrik Schrezenmaier, Judith Simon, Sebastian Spies, Steffen Suerbier, Fabian Wegscheider, Jan Zur

Computerorientierte Mathematik I

Große natürliche Zahlen

6. Programmieraufgabe

Abnahme: spätestens am 6./7.12.2012 (je nach Gruppennummer).

Gerade Gruppennummern geben spätestens am Donnerstag, ungerade Gruppennummern geben spätestens am Freitag ab.

Schreibt eine Klasse, die beliebige große natürliche Zahlen verwalten kann. Diese Funktionalität wird auch von der Java eigenen Klasse `java.math.BigInteger` bereit gestellt, die ihr jedoch nicht benutzen sollt. Die Zahl soll ziffernweise in einem Array von Bytes `digits` abgelegt werden, das heißt jeder Eintrag des Arrays soll für eine Ziffer $0, 1, \ldots, 8, 9$ der Zahl im Zehnersystem stehen. Dabei soll die letzte Ziffer der Zahl in `digits[0]`, die vorletzte in `digits[1]`, usw. abgelegt werden.

(a) Legt eine Klasse `BigInt` mit einem privaten Datenfeld `byte[] digits` an und implementiert die get- und set-Methoden. Implementiert die Methode `toString()`, die eine String-Representation zurück gibt und die Methode `equals(BigInt other)`, die zwei `BigInt`s miteinander vergleicht.

(b) Implementiert drei Konstruktoren: `BigInt()` erstellt ein `BigInt`, das die Zahl 0 repräsentiert, `BigInt(long input)` erhält ein `long` als Input und erstellt ein `BigInt`, das die übergebene Zahl repräsentiert. `BigInt(byte[] newDigits)` erhält ein Array von Bytes das in die interne Datenstruktur kopiert werden soll. Dabei sollen führende Nullen ignoriert werden.

(c) Implementiert die Methode `public BigInt add(BigInt addend)`, die einen `BigInt addend` übergeben bekommt, diesen zum aktuellen `BigInt` addiert und das Ergebnis zurück gibt.

(d) Implementiert die Methode `public BigInt multiply(BigInt factor)`, die einen `BigInt factor` übergeben bekommt, diesen mit dem aktuellen `BigInt` multipliziert und das Ergebnis zurück gibt. **Hinweis:** Überlegt euch zunächst wie die Multiplikation großer Zahlen funktioniert, bevor ihr die Methode implementiert. Nutzt die Methode aus (c), wenn nötig.

(e) Testet eure Implementation in der `main`-Methode, indem ihr für $k = 1, \ldots, 100$ die Summe $\sum_{i=0}^{k} i$ und die Fakultät $k!$ ausgebt.

Dokumentiert die Klasse, alle Methoden und Datenfelder nach dem Javadoc-Standard. Erstellt vor der Abgabe eures Programms eine Javadoc-Dokumentation, indem ihr im Verzeichnis mit eurem Programm den Befehl `javadoc BigInt.java` aufruft.

Viel Spaß und Erfolg!