KRY Serie 8

(1) Eine Primzahl p heisst eine *sichere Primzahl*, falls sie von der Form $p = 2q + 1$ ist, wobei q selbst auch prim ist.
Erstellen Sie für die Klasse `BigInteger` der Package `mybiginteger` eine `static` Methode

```
public static BigInteger myProbableSafePrime(
                                    int bitLength,
                                    int certainity,
                                    Random rnd),
```

welche eine sichere Primzahl der angegebenen Bitlänge `bitLength` generiert. Der Parameter `certainity` gibt dabei an, wie viele Durchläufe beim verwendeten Primzahltest `BigInteger.isProbablePrime()` ausgeführt werden sollen.

Hinweis: Ähnlich wie man bei den Primzahltests (Fermat- bzw. Miller-Rabin-Test) zuerst bis zu einer definierten Obergrenze Probedivisionen für kleine Primfaktoren durchführt, gibt es auch bei der Suche nach sicheren Primzahlen Möglichkeiten, die Laufzeit zu verbessern:

- Für sichere Primzahlen $p = 2q + 1$ gilt (ohne Herleitung): Die Primzahl q muss zwingend $q = 5(mod\ 6)$ erfüllen.
- Angenommen für eine ungerade Primzahl r gelte $q = \frac{r-1}{2}(mod\ r)$. Dann kann $p = 2q + 1$ unmöglich eine Primzahl sein. Dies bietet also die Möglichkeit, mit kleinen ungeraden Primzahlen r bis zu einer definierten Obergrenze, für potenzielle Kandidaten q zeitlich günstige Vortests durchzuführen. Verwenden Sie für diese Tests die Tabelle `tableOfPrimes` aus Serie 6, ohne Aufruf der Methode `createTableOfPrimes()` (Das erstellen der Tabelle übernimmt die Testumgebung!).

Überprüfen Sie Ihren Algorithmus in der Testumgebung „Prakt. 8.1“ des Programms KryptoTrainer.

(2) Vervollständigen Sie das Source-File `ElGamal.java` des Programms KryptoTrainer. Benutzen Sie dabei für die Schlüsselerzeugung Ihre eigene Methode

```
BigInteger.myProbableSafePrime()
```

aus Aufgabe (1), um eine *sichere Primzahl* der vorgegebenen Bitlänge zu bestimmen. Mit der Verwendung von sicheren Primzahlen p wird die Suche nach erzeugenden Elementen g für die Einheitengruppe $\mathbb{Z}_p^*$ auch bei grossen Bitlängen von p in jedem Fall durchführbar (warum?).
Überprüfen Sie das El Gamal Kryptosystem in der Testumgebung „Prakt. 8.2“ des Programms KryptoTrainer.