



Objektorientierte Programmierung - Praxistest

Hinweise

- Bearbeitungsdauer: 60 Minuten
- Der Name Ihrer Datei, welche die Klasse enthält, muss der Konvention `Matrikelnummer_Login.cpp` entsprechen.
- Skript, Bücher, Notizen sind erlaubt. Notebooks, PDAs, etc. dürfen nicht verwendet werden. Mobiltelefone müssen ausgeschaltet werden.
- Zugriffe auf das Internet während des Tests sind nicht zulässig.
- Zugriffe auf Verzeichnisse anderer Benutzer sind während des Tests nicht zulässig. Ihr Homeverzeichnis wurde automatisch für andere Benutzer gesperrt.
- Achtung! Zur Kontrolle werden Daten über Ihre Aktivitäten aufgezeichnet.
- Tauschen Sie keinesfalls irgendwelche Dinge mit den Nachbarn aus. Notfalls rufen Sie eine Aufsichtsperson zur Kontrolle.
- Fragen Sie, wenn Ihnen eine Aufgabe nicht klar ist!
- Achten Sie auf aussagekräftige Kommentare und Ihren Programmierstil.

Bewertung

- Der Test wird als durchgefallen gewertet, wenn Sie nicht in der Lage sind, die Klasse in einem Editor bzw. einer Entwicklungsumgebung zu schreiben und zusammen mit dem vorgegebenem Rahmenprogramm zu übersetzen und auszuführen.
- Bei unverständlichen Fehlermeldungen des Compilers können Sie fragen. Wir geben allerdings nur begrenzt häufig Tipps, und nur, wenn es wirklich eine ungewöhnlichere Fehlermeldung ist.
- Programme, die durch den Compiler laufen, aber noch nicht die richtige Ausgabe liefern, werden von den Betreuern nach strengen Maßstäben gewertet: Nur Programme, die wirklich ganz dicht an einer korrekten Lösung sind, werden als ausreichend akzeptiert.
- Auch wenn ein Programm für den Test die richtige Ausgabe erzeugt, kann es abgelehnt werden, wenn es Fehler enthält oder allgemein anerkannte Regeln guten Programmierstils verletzt (z.B. keine Kommentare, keine Einrückung).
- Wird beim späteren Vergleich erkannt, dass Ihre Abgabe ein Plagiat von einem anderem Studenten ist, so werden beide Arbeiten als ungültig gewertet und Sie sind durchgefallen.

Definieren Sie eine Klasse `binaerzahl` für 64 Bit Binärzahlen. Sie können die Zahl intern z.B. durch ein `int`-Array der Länge 64 darstellen, aber Sie sind in der Implementierung frei.

Wenn ein Objekt der Klasse angelegt wird, soll die Binärzahl mit dem Wert 0 initialisiert werden.

Die Klasse soll folgende Methoden besitzen:

- a) Konstruktor, welcher alle Bits der Zahl auf 0 initialisiert.
- b) Methode `set_bit`, mit einem Parameter, einer Bitposition (vom Typ `int`). Das Bit an dieser Position soll auf 1 gesetzt werden. Bitpositionen werden dabei von 0 an gezählt, gültige Bitpositionen sind also 0 bis 63. Das Bit an Position 0 ist das niederwertigste Bit, das an Position 63 das höchstwertige Bit.

Falls keine gültige Bitposition übergeben wurde, soll die Methode `false` zurückliefern, und die Binärzahl unverändert lassen.

Wurde eine gültige Bitposition übergeben, so wird diese auf 1 gesetzt und die Methode liefert `true` als Return-Wert zurück.

- c) Methode `increment`, welche die Binärzahl um 1 erhöht: Dazu werden alle Bits, beginnend vom niederwertigsten Bit, durchgegangen. Ist das Bit 0, so wird es auf 1 gesetzt, und man ist fertig. Ist das Bit 1, so wird es auf 0 gesetzt, und man muß mit dem nächsten Bit weitermachen (man hat dann ja einen Überlauf in die nächste Binärstelle). Falls vorher alle Bits der Zahl 1 waren, werden so alle alle Bits auf 0 gesetzt. Das ist so beabsichtigt, es soll kein Fehler gemeldet werden.

Die Methode `increment` hat also keinen Parameter und liefert auch keinen Wert zurück.

- d) Methode `output`, welche den aktuellen Wert der Binärzahl auf der Standardausgabe ausgibt. Dabei sollen alle Bits in einer Zeile hintereinander ausgegeben werden, beginnend natürlich mit dem höchstwertigen Bit (das Bit an Position 0 muß ganz am Ende ausgegeben werden). Die Funktion soll weder einen Wert zurückliefern noch wird ein Parameter benötigt.

Zum Beispiel würde das Programmstück

```
binaerzahl bzahl;
if (bzahl.set_bit(0) && bzahl.set_bit(1))
{
    bzahl.increment();
    bzahl.increment();
    cout << "Dezimalzahl: ";
    bzahl.dezimal();
    cout << endl;
}
else
    cout << "Es wurden ungueltige Bitpositionen angegeben!\n";
```

die Ausgabe

```
Binaerzahl: 0...0000000000000101
```

erzeugen.

Das Hauptprogramm, mit welchem die Klasse zur Bewertung getestet wird, befindet sich im Ordner `oop08_test2` in ihrem Homeverzeichnis unter dem Namen `binaer.cpp`. Falls Sie die Datei nicht vorfinden oder Sie diese versehentlich zerstört haben, können Sie sie auch vom Betreuer bekommen.