



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

In folgenden Fällen wird der Test als durchgefallen bewertet:

Wer nicht in der Lage ist, die Klasse in einem Editor bzw. einer Entwicklungsumgebung zu schreiben und zusammen mit dem vorgegebenem Rahmenprogramm zu übersetzen und auszuführen, gilt als durchgefallen.

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.

Fehlerhafte Programme mit fehlenden bzw. nicht aussagekräftigen Kommentaren oder linksbündig geschriebenen Quellcode werden ebenfalls nicht bewertet.

Definieren Sie eine Klasse `dupTest` zur Erkennung von Duplikaten von ganzen Zahlen (`int`-Werten). Die Klasse soll einen Speicher für eine Menge von `int`-Werten enthalten (maximal 100). Sie können den Speicher z.B. als Array implementieren.

Wenn ein Objekt der Klasse angelegt wird, soll der Speicher leer sein (initialisieren Sie das Objekt passend im Konstruktor).

Anschließend kann der Benutzer die Methode `store` aufrufen, z.B.

```
dupTest o;  
int status = o.store(10);
```

Diese Methode soll eine Zahl (hier 10) in den Speicher des Objektes einfügen. Die Methode soll folgende Rückgabewerte (`status`) haben:

- 1, wenn die Zahl vorher noch nicht in der zu verwaltenden Menge enthalten war (sie also wirklich neu eingefügt wurde),
- 0, wenn die Zahl bereits vorher eingefügt wurde (die Menge somit gar nicht geändert werden muß),
- -1, wenn die Zahl neu ist, aber nicht eingefügt werden kann, weil das zur Implementierung verwendete Array voll ist (also bereits 100 verschiedene Zahlen enthalten sind).

Im obigen Beispiel muß `status` also 1 sein, weil die Zahl 10 direkt nach dem Anlegen des Objektes sicher neu ist.

Beachten Sie bitte, daß die zu speichernden Zahlen beliebige `int`-Werte sein können, also nicht auf den Bereich 1–100 eingeschränkt sind. Nur die Anzahl der verschiedenen Werte ist maximal 100.

Als Implementierung würde sich daher anbieten, neben dem Array noch eine Variable für Anzahl der Werte im Array zu verwenden. Dies könnte immer den nächsten freien Platz im Array angeben.

Neben dem Konstruktor und der Methode `store` müssen Sie als Drittes noch eine Methode `output` implementieren, die die aktuelle Menge von Zahlen ausgibt, die in dem Objekt gespeichert sind. Diese Methode hat keinen Parameter und keinen Rückgabewert. Sie soll die vorher mit `store` gespeicherten Zahlen ohne Duplikate ausgeben (auf der Standard-Ausgabe, ein Wert pro Zeile). Die Reihenfolge ist dabei egal. Zum Beispiel würde das Programmstück

```
dupTest o;  
int status;  
status = o.store(10); // status ist 1  
status = o.store(20); // status ist 1  
status = o.store(10); // status ist 0  
status = o.store(30); // status ist 1  
o.output();
```

die Ausgabe

```
10  
20  
30
```

erzeugen (in beliebiger Permutation). Wenn die Menge von Zahlen im Objekt leer ist, soll nichts ausgegeben werden.

Das Hauptprogramm, mit welchem die Klasse zur Bewertung getestet wird, befindet sich im Ordner `oop08_test` in ihrem Homeverzeichnis unter dem Namen `dup.cpp`. Falls Sie die Datei nicht vorfinden oder Sie diese versehentlich zerstört haben, können Sie sie auch vom Betreuer bekommen.