

Anmerkungen

- Sichern Sie ihre Berechnungen, so weit wie möglich gegen fehlerhafte Eingaben ab.
- Verwenden Sie sprechende Variablennamen und formatieren Sie Ihren Programmtext ordentlich.
- Erstellen Sie Inputprompts und Ausgaben, sodass das Programm für BenutzerInnen angenehm zu bedienen ist.
- Schreiben Sie Ihre Programm so, dass mehrere Berechnungen ausgeführt werden können, ohne das Programm immer wieder neu starten zu müssen.
- Testen Sie Ihre Programme und unterziehen Sie die Ergebnisse einer Prüfung (Schätzung des erwarteten Ergebnisses, bzw. im Notfall mit dem Taschenrechner prüfen).
- Mit * markierte Beispiele sind etwas anspruchsvoller
- Mit + markierte Beispiele sollen Aufgabenstellungen repräsentieren, wie sie in der Praxis an einen herangetragen werden. Die schwammige Definition der gewünschten Funktionalität ist Absicht. Treffen Sie geeignete, sinnvolle Annahmen, oder fragen Sie gegebenenfalls im Forum nach.

Aufgabe 1

Implementieren Sie die Klasse Vector mit der gesamten bisher in den Übungen vorgegebenen Funktionalität.

Aufgabe 2

Die Klasse Geheim dient zum Speichern von wichtigen Informationen, die mit Passwörtern geschützt werden können. Jedes Objekt enthält außer der zu speichernden Zeichenkette noch eine zweite Zeichenkette, das Passwort. Ist das Passwort leer, so ist das Objekt offen, ansonst ist es gesperrt. Implementieren und testen Sie zumindest die folgenden Methoden:

Geheim::Geheim(char inhalt[]) // offenes Objekt mit inhalt als Information

Geheim::Geheim(char inhalt[], char passwort[]) // mit passwort geschütztes Objekt

bool Geheim::lock(char passwort[]) // sperrt ein Objekt mit passwort. Das darf nur funktionieren, wenn das Objekt vorher offen war. Sonst muß diese Funktion false liefern und das Objekt unverändert lassen.

bool Geheim::unlock(char passwort[]) // entsperrt ein Objekt, falls das Passwort korrekt ist. Ansonst wird false zurückgeliefert.

Geheim Geheim::operator+(Geheim) //funktioniert nur, wenn die beiden zu verknüpfenden Objekte offen sind. Dann werden die beiden Zeichenketten im Ergebnisobjekt einfach aneinandergehängt. Andernfalls wird ein leeres Objekt retourniert.

int Geheim::len() //Länge der Zeichenkette. Liefert 0 für gesperrte Objekte

void Geheim::print() //Gibt die Information aus. Für gesperrte Objekte wird nichts ausgegeben.

Aufgabe 3

Schreiben Sie eine Klasse Complex zur Repräsentation von komplexen Zahlen. Implementieren Sie den nötigen Programmcode, sodass das folgende Programm das erwartete Ergebnis bringt.

```
#include <iostream.h>
#include "Complex.h"
int main() {
    Complex c1{2},c2{3,-4};
    double d1=2, d2{3};
    c2.print(); cout << " ist " << (c2==d2 ? "" : "nicht ") << "gleich ";
    cout << d2 << endl;
    c1.print(); cout << " ist " << (c1!=d1 ? "nicht " : "") << "gleich ";
    cout << d1 << endl;
    cout << d1 << " ist " << (d1==c1 ? "" : "nicht ") << "gleich ";
    c1.print(); cout << endl;
    cout << d2 << " ist " << (d2!=c2 ? "nicht " : "") << "gleich ";
    c2.print(); cout << endl;
    cout << "Die konjugiert komplexe Zahl zu "; c2.conj().print();
    cout << " ist "; c2.print(); cout << endl;
    return 0;
}
```

Aufgabe 4

Die Ackermann-Funktion $A(n,m)$ ist für natürliche Zahlen n und m wie folgt definiert:

$$A(0,m)=m+1$$

$$A(n,0)=A(n-1,1) \text{ falls } n>0$$

$$A(n,m)=A(n-1,A(n,m-1)) \text{ falls } n>0 \text{ und } m>0$$

- Schreiben Sie eine rekursive Funktion, die die Ackermannfunktion für eingelesene Werte von n und m berechnet und ausgibt.
- Implementieren Sie eine Klasse Stack. Verwenden Sie diese Klasse, um die Ackermann Funktion iterativ (also nur mit Verwendung von Schleifen und ohne Rekursion) zu lösen.

Aufgabe 5

Ein Tischler hat ein Lager mit Brettern in drei Holzsorten (Buche, Eiche u. Mahagoni). Alle Bretter haben den selben Querschnitt, sind aber unterschiedlich lang. Wird ein Brett einer bestimmten Länge verkauft, so wird, falls die entsprechende Länge nicht vorrätig ist, einfach das passende Stück von einem der vorhandenen Bretter abgeschnitten. Um den Verschnitt nicht allzu groß werden zu lassen, soll dabei immer das kürzest mögliche Brett zersägt werden. Schreiben Sie ein Programm, das dem Tischler hilft, sein Lager zu verwalten. Folgende Funktionen sind unabdingbar: Erfassung des aktuellen Bestandes,

Einkauf von Brettern beliebiger Länge, Verkauf von Brettern beliebiger Länge (das Programm soll das kürzest mögliche Brett automatisch vorschlagen), Aufräumen des Lagers (dabei werden alle Bretter, die eine vom Benutzer festzulegende Länge unterschreiten, aus dem Lager entfernt).

Mögliche Klassenkandidaten: Brett, Lager.

Aufgabe 6 +

Ein Autoverleih verleiht Autos in den Klassen Economy, Normal und Luxus. Die Verwaltung des Fuhrparks soll durch ein Programm erleichtert werden. Folgende Funktionen sind unabdingbar: Verleih (Der Kunde spezifiziert die gewünschte Klasse und bekommt den durch das Programm ermittelten Stellplatz mitgeteilt), Rückgabe (Der Rechner schlägt einen freien Stellplatz vor und ermittelt die gefahrenen Kilometer), Ausmustern (Die Stellplätze aller Wagen, die eine vom Benutzer vorzugebende Kilometerleistung überschritten haben, sollen vom Rechner ermittelt und ausgegeben werden. Da diese Wagen woanders veräußert werden, sind die entsprechenden Stellplätze ab diesem Zeitpunkt auch wieder als frei zu betrachten).

Klassenkandidaten: Fahrzeug, Garage.

Aufgabe 7 +

Ein befreundeter, fanatischer CD-Sammler hat Sie gebeten, ihm ein einfaches Programm zur Verwaltung seiner Sammlung zu schreiben. Er wünscht sich folgende Funktionen: Für jedes Album sollen Interpret, Name und Stilrichtung speicherbar sein. Eingabe, Ändern und Löschen von Eintragungen soll möglich sein. Eine Suche soll entweder nach Interpret, nach Albumname oder Stilrichtung vorgenommen werden können und eine Liste aller jeweils passenden CDs liefern.

Klassenkandidaten: Album, Sammlung

Aufgabe 8 +

Für eine Bar soll ein Programm zur Unterstützung des Einkaufs Verwendung finden. Es wird eine kleine Anzahl von verschiedenen Spirituosen verwendet, um eine ebenfalls kleine Zahl von festgelegten Drinks herzustellen. Die Rezepte für die Drinks sind bekannt und Änderungen im Angebot sind keine zu erwarten. Das Programm soll folgende Funktionen zur Verfügung stellen: Einkauf (Einkauf von Spirituosen), Verkauf (Verkauf von Drinks), Bestandstest (Ausgabe aller Spirituosen, deren Bestand unter einer vom Benutzer zu definierenden Menge liegt) und Spezialverkauf. Beim Spezialverkauf wird für den Kunden nach Wunsch ein Drink gemixt, der nicht auf der Karte aufgeführt ist.

Klassenkandidaten: Drink, Rezept, Bar.

Aufgabe 9 +

Ein Kneipenbesitzer hat Sie um ein Programm zur Unterstützung seiner regelmäßigen Turnierveranstaltungen gebeten. Nach Eingabe der angemeldeten Teilnehmer und der Anzahl der verfügbaren Geräte (=Anzahl der Spiele, die gleichzeitig durchgeführt werden können) soll der Computer

einen Spielplan für ein simples KO-System entwerfen und angeben, welche Paarungen auf welchem Gerät spielen müssen. Nach Eingabe eines Ergebnisses für ein Gerät, soll die nächste Paarung (für die beide Spieler verfügbar sind) ermittelt, und das Gerät für diese reserviert werden. Nach Beendigung des Turnieres sollen die zwei besten TeilnehmerInnen ausgegeben werden, um die Preisverleihung vorzubereiten.

Klassenkandidaten: Turnier, Teilnehmer, Gerät