

## Vorlesung „Objektorientierte Programmierung“ — 2. Programmierertest (Aufgabe D) —

### Hinweise/Regeln:

- Bearbeitungsdauer: 75 Minuten.
- Am Ende gibt es nur „bestanden“ und „nicht bestanden“, keine Punkte für partiell korrekte Lösungen.
- Sie dürfen bis zu 3 Blätter „Spickzettel“/„Quick Reference“ verwenden, sowie ein Buch/Hefter (nicht zu groß, es muß noch auf den Tisch passen ohne zu stören).
- Eigene Notebooks, PDAs, etc. dürfen nicht verwendet werden. Mobiltelefone bitte ausschalten (oder mit der Aufsicht besprechen).
- Legen Sie bitte ein neues Verzeichnis `oop10/praxistest2` in Ihrem Homeverzeichnis an. Sie dürfen ein eventuell existierendes `Makefile` hineinkopieren oder Shellskripte zur Programmentwicklung. Ansonsten dürfen Sie keine weiteren Dateien kopieren oder öffnen, sondern nur den Quellcode des zu entwickelnden Programms.
- Kopieren Sie sich dann bitte die Datei `~brass/oop10/test2/p2d.cpp` in dieses Verzeichnis. Diese Datei enthält ein Hauptprogramm, das als Test dient. Ihre Aufgabe ist, an der markierten Stelle die in der Aufgabenstellung beschriebene Klasse einzufügen. Am Ende muss das Hauptprogramm genau der vorgegebenen Version entsprechen, zwischenzeitlich können Sie natürlich eigene Testaufrufe einbauen.
- Sie dürfen selbstverständlich die zur Programmentwicklung üblichen Programme verwenden (auch `man`), aber nicht auf das Internet zugreifen (z.B. kein Web-Browser, kein EMail-Programm).
- Die Homeverzeichnisse werden für Zugriffe von außen gesperrt. Falls Sie spezielle Zugriffsrechte gesetzt hatten, müssen Sie diese nach dem Test selbst wieder herstellen.
- Selbstverständlich dürfen Sie auch Microsoft Visual Studio benutzen. Das abgegebene Programm muss aber unter Linux/g++ laufen.
- Tauschen Sie keinesfalls irgendwelche Dinge mit den Nachbarn aus. Notfalls rufen Sie eine Aufsichtsperson zur Kontrolle.
- Sie müssen Mindestanforderungen an den Programmierstil erfüllen, z.B. entsprechend der Programmstruktur einrücken, sinnvolle Variablennamen wählen und zum Verständnis notwendige Kommentare schreiben.
- Fragen Sie, wenn Ihnen die Aufgabe nicht klar ist! Melden Sie sich bitte auch, wenn es technische Schwierigkeiten mit Ihrem Rechner gibt.
- Wenn Sie an einer unverständlichen Fehlermeldung länger festhängen, können Sie probieren, zu fragen. Wir wollen aber nicht zu viele Tipps geben.

## Aufgabe (Variante D)

Zur Überwachung der Streusalzvorräte protokolliert der Winterdienst in Halle im Dezember den täglichen Verbrauch des Streusalzes. Schreiben Sie hierfür eine Klasse mit dem Namen `statistik`. In dieser Klasse soll in einem Array vom Typ `double` mit dem Namen `streusalz` für jeden Tag des Monats Dezember die benötigte Streusalzmenge in `t` gespeichert werden.

Falls nötig, so kann die Klasse um weitere Attribute ergänzt werden.

Die Klasse `statistik` implementiert die folgende Methoden:

- Konstruktor, welcher das Array initialisiert
- `set_stsmenge`, welche zwei Parameter übergebenen bekommt und einen Rückgabewert vom Typ `bool` hat.

Im ersten Parameter steht der Tag und im zweiten die zugehörige verbrauchte Streusalzmenge. Die Methode prüft, ob der Tag gültig ist (Wert zwischen 1 und 31) und ob die übergebene Streusalzmenge  $\geq 0$  ist. Falls dies der Fall ist, so wird die übergebene Streusalzmenge im Array an der richtigen Position gespeichert und die Methode gibt `true` zurück. Falls einer der Parameter nicht gültig ist, so gibt die Methode `false` zurück. Die Methode nimmt in diesem Fall keine Veränderungen am Array vor.

- `get_max_tag`, liefert im Returnwert den Tag mit dem höchsten Streusalzverbrauch zurück.
- `get_summe`, liefert den Gesamtverbrauch an Streusalz im Monat Dezember per Returnwert zurück.

### Beispiel: Das Programmstück

```
int main ()
{ statistik dezember;
  int tag=1;
  float menge= 1.0;
  float streusalz[31] = { 0.4, 1.5, 3.2, 9.1, 0.0, 0.0, 1.34, 6., 0.32,
                        0.01, 0.5, 0., 2.4, 5.9, 7.2, 4.0, 0.0, 0.0, 0.04, 0.6,
                        6.2, 3.98, 1.23, 0.0, 1.2, 1.5, 3.8, 5.4, 0.0, 0., 3.6 };

  for (int i = 0; i < 31; i++)
    if (! dezember.set_stsmenge(i+1, streusalz[i]))
      cout << "Ungueltige Eingabe!" << endl;

  cout << "Streusalzstatistik Dezember: " << endl << endl;
  cout << "Tag mit dem groessten Streusalzverbrauch : ";
  dezember.get_max_tag () << ". Dezember" << endl;

  cout << "Insgesamt wurden " << dezember.get_summe()
        << " t Streusalz verbraucht." << endl;
  return 0;
}
```

erzeugt die Ausgabe:

```
Streusalzstatistik Dezember:
```

```
Tag mit dem groessten Streusalzverbrauch : 4. Dezember
Insgesamt wurden 18.8 t Streusalz bentigt.
```