



10. Übung Softwaretechnik

Generative Programmierung

Abgabe: 20.01.2006, 12 Uhr

1 STL Templates (13 Punkte)

Zur Lösung dieser Aufgabe müssen Sie ein Programm schreiben. Geben Sie dazu bitte eine ausgedruckte Version Ihres Programms ab, sowie eine Fassung per Email mit dem Betreff *Exercise 10.1* an `se0506@st.cs.uni-sb.de`. Übersetzen Sie Ihre Programme mit g++.

Sie wollen in einem `std::set<int>` alle Elemente um Eins erhöhen. Hier ein Implementierungsvorschlag:

```
#include <iostream>
#include <set>

static void increment_set(std::set<int>&s) {
    for (std::set<int>::iterator i = s.begin(); i != s.end(); i++)
        (*i)++;
}

int main() {
    std::set<int> s;
    s.insert(2); s.insert(3); s.insert(5);
    s.insert(7); s.insert(11);

    increment_set(s);

    for (std::set<int>::const_iterator i = s.begin(); i != s.end(); i++)
        std::cout << *i << std::endl;

    return 0;
}
```

Leider funktioniert das nicht, da der Übersetzer eine Fehlermeldung auswirft.

- Ermitteln Sie die Fehlermeldung, die Ihr Compiler auswirft. Wirft Ihr Compiler keine Fehlermeldung aus, geben Sie Name und Versionsnummer des Compilers an. (Übermitteln Sie dem Hersteller ausserdem einen Bugreport; der Compiler ist dann nicht standardkonform.) (2 Punkte)
- Welchen Grund hat das Verhalten des Compilers? Geben Sie eine Fehlermeldung an, die es einem Programmierer eher ermöglicht, den Fehler zu finden und zu beheben. (3 Punkte)
- Ist es möglich, eine funktionierende Version von `increment_set` zu implementieren, ohne dessen Signatur zu verändern? Begründen Sie Ihre Antwort. (3 Punkte)
- Implementieren Sie eine funktionierende Version von `increment_set`. Wenn Sie im vorangehenden Aufgabenteil „nein“ geantwortet haben, dürfen Sie die Signatur verändern, sonst nicht. Sie dürfen aber keinen anderen Container als `std::set<int>` verwenden. (5 Punkte)

2 Spezialisierung von Templates in C++ (17 Punkte)

Zur Lösung dieser Aufgabe müssen Sie ein Programm schreiben. Geben Sie dazu bitte eine ausgedruckte Version Ihres Programms ab, sowie eine Fassung per Email mit dem Betreff *Exercise 10.2* an `se0506@st.cs.uni-sb.de`. Übersetzen Sie Ihre Programme mit `g++`.

Implementieren Sie eine Template-Klasse `Buffer<typename T, unsigned int size>` mit folgenden Eigenschaften:

1. Die Standard-Implementierung enthält einen Zeiger auf `T`.
2. Bei der Konstruktion wird dieser Zeiger mit einem Array von `T` der Länge `size` initialisiert.
3. Dieses Array soll bei der Destruktion wieder freigegeben werden.
4. Es gibt einen Indizierungs-Operator mit der Signatur

```
const T operator[] (unsigned int n) throw(std::out_of_range)
```

Hinweis: `std::out_of_range` wird im Header `<stdexcept>` definiert.

5. Dieser Indizierungs-Operator gibt das `n`-te Element des Puffers zurück, falls `n < size` und wirft eine `std::out_of_range`-Exception sonst.
6. Für den Fall, dass `T = bool` und `size = 16` ist, soll `Buffer` spezialisiert werden, so dass statt eines `T*` ein `unsigned short` verwendet wird. (Verifizieren Sie vorher, dass in Ihrer Implementierung `sizeof(short) ≥ 2` ist.) Das k -te Element des Puffers ist dann repräsentiert durch das k -te Bit der Ganzzahl.
7. Erweitern Sie Ihre Implementierung um folgendes kurze Programm:

```
int main() {
    Buffer<bool, 15> b1;
    Buffer<bool, 16> b2;

    std::cout << "Size 1 = " << sizeof(b1) << std::endl;
    std::cout << "Size 2 = " << sizeof(b2) << std::endl;

    return 0;
}
```

Sie sollten in der ersten Zeile mindestens 4 und in der zweiten Zeile mindestens 2 erhalten; beide Zahlen sollten verschieden sein.

Abgabe

Bilden Sie Teams aus je zwei Studenten, erarbeiten Sie die Lösung *gemeinsam* und reichen Sie *eine Lösung pro Team* ein. Drucken Sie ihre Lösungen aus, klammern Sie sie zusammen und werfen Sie sie in die mit "Softwaretechnik" beschrifteten Übungskästen vor Hörsaal 1 in Gebäude E1 1 (45). Abgaben per E-mail werden nicht akzeptiert. Einzelabgaben können nur in begründeten Ausnahmen akzeptiert werden. Dazu wenden Sie sich bitte *vor* der Abgabe per email an Valentin Dallmeier <dallmeier@st.cs.uni-sb.de>. Nicht genehmigte Einzelabgaben werden mit 0 Punkten bewertet.

Fragen?

Fragen zu diesem Übungsblatt können sie in Ihrer Übungsgruppe stellen.