



3071 Programmieren 3 AI

Klausur

SS 2012

2012-07-05

Name:

Vorname:

Matrikelnr.:

Unterschrift

Erlaubte Hilfsmittel: keine.

Die Lösungen sind auf den Klausurbögen anzufertigen. Verwenden Sie einen dokumentenechten Stift nichtroter Farbe. Lösen Sie die Heftung der Klausur nicht. Notieren Sie die Verwendung von Rückseiten auf den entsprechenden Vorderseiten.

Bearbeitungszeit: 90 Minuten.

Aufgabe	1	2	3	4	5	Gesamt	Note
Punkte	14	20	20	26	20	100	–
erreicht							

Aufgabe 1 (14 Punkte)

Das folgende Java-Programm enthält ein Synchronisationsproblem. Erläutern Sie, wodurch das Problem entsteht und wieso der Quelltext in der angegebenen Form ungeeignet ist. Beziehen Sie sich dabei auch auf die Codezeilen, durch die das Problem zustande kommt, möglichst unter Verwendung entsprechender Fachbegriffe. Beschreiben Sie außerdem eine Möglichkeit, das Problem zu beheben.

```
1 class A implements Runnable {
2     private Object o1;
3     private Object o2;
4
5     public A(Object o1, Object o2) {
6         this.o1 = o1;
7         this.o2 = o2;
8     }
9
10    public void run() {
11        for (int i = 0; i < 100; i++) {
12            synchronized (o1) {
13                synchronized (o2) {
14                    try {
15                        System.out.println(
16                            Thread.currentThread().getName()
17                            + ": " + i);
18                        Thread.sleep(100);
19                    } catch (Exception e) {
20                        e.printStackTrace();
21                    }
22                }
23            }
24        }
25    }
26 }
27 class Main {
28     public static void main(String[] args) {
29         Integer a1 = 17;
30         Integer b1 = 23;
31
32         new Thread(new A(a1, b1)).start();
33         new Thread(new A(b1, a1)).start();
34     }
35 }
```


Aufgabe 2 (20 Punkte)

Jeder der folgenden C++-Quelltexte enthält einen Fehler, der die Übersetzung zu einem ausführbaren Programm verhindert. Erläutern Sie den Fehler und beschreiben Sie, wie er behoben werden kann.

(a)

```
1 #include <string>
2
3 template <typename A, typename B> class P {
4 public:
5     A fst;
6     B snd;
7     P(A x, B y) : fst(x), snd(y) {}
8     void swap() {
9         B x = snd;
10        snd = fst;
11        fst = x;
12    }
13 };
14
15 int main() {
16     P<std::string, std::string>* p1 =
17         new P<std::string, std::string>("hallo", "welt");
18     P<int, std::string>* p2 = new P<int, std::string>(17, "4");
19     p2->swap();
20     p1->swap();
21     return 0;
22 }
```

(b)

```
1 #include <string>
2 #include <iostream>
3 using namespace std;
4
5 class Person {
6 public:
7     string name;
8     string vorname;
9
10    Person(string name, string vorname);
11    string getFullName();
12 };
13
14 Person::Person(string name, string vorname) :
15     name(name), vorname(vorname) {}
16
17 string Person::getFullName() {
18     return vorname+" "+name;
19 }
20
21 int main() {
22     char* n = "Zerlett";
23     string v = "Helmut";
24     Person p = new Person(n, v);
25     cout << p.getFullName() << endl;
26     cout << p.vorname << endl;
27 }
```

(c)

```
1 #include <iostream>
2
3 class A {
4 public:
5     int answer();
6 };
7
8 class B {
9 public:
10     std::string message();
11 };
12
13 class C : public A, B {
14 public:
15     int x;
16     C();
17 };
18
19 int A::answer() {
20     return 42;
21 }
22
23 std::string B::message() {
24     return "we apologise for the inconvenience";
25 }
26
27 C::C() : x(46) {}
28
29 int main() {
30     C c;
31     std::cout << c.answer() << std::endl;
32     std::cout << c.message() << std::endl;
33     std::cout << (c.x / 2) << std::endl;
34     return 0;
35 }
```

(d)

```
1 #include <iostream>
2
3 class Date {
4     int year;
5     int month;
6     int day;
7
8     Date(int year, int month, int day);
9 };
10
11 Date::Date(int year, int month, int day) :
12     year(year), month(month), day(day) {}
13
14 int main() {
15     Date d=Date(2004,1,26);;
16     std::cout << "starting main" << std::endl;
17 }
```

(e)

```
1 #include <iostream>
2
3 int f() {
4     return 42;
5 }
6
7 int main() {
8     int y = f();
9     int& x = f();
10    x=y;
11 }
```

Aufgabe 3 (20 Punkte)

Berechnen Sie die Ausgabe der folgenden C++-Programme von Hand; verzeichnen Sie dabei ggf. auch nicht gefangene Exceptions. Erklären Sie kurz, wie es zu der Ausgabe kommt.

(a)

```
1 #include <iostream>
2
3 int f(int& x) {
4     int r = x;
5     x = (x % 2 == 0) ? (x / 2) : (3 * x + 1);
6     return r;
7 }
8
9 int main() {
10     int i;
11     for (i = 0; i < 4; i++);
12         std::cout << f(i) << std::endl;
13 }
```

(b)

```
1 #include <iostream>
2
3 class C {
4 public:
5     int counter;
6     C();
7     int next();
8 };
9
10 C::C() : counter(42) {}
11
12 int C::next() {
13     counter = counter + 1;
14     if (counter > 43) {
15         counter = 0;
16         throw 17;
17     }
18     return counter;
19 }
20
21 int main() {
22     C c;
23     try {
24         std::cout << c.next() << std::endl;
25         std::cout << c.next() << std::endl;
26         std::cout << c.next() << std::endl;
27         std::cout << c.next() << std::endl;
28     } catch (std::string s) {
29         std::cout << s << std::endl;
30         std::cout << c.next() << std::endl;
31     } catch (int i) {
32         if (i == 42) {
33             std::cout << c.next() << std::endl;
34             std::cout << c.next() << std::endl;
35         } else {
36             std::cout << "unknown error: " << i << std::endl;
37             std::cout << c.next() << std::endl;
38         }
39     }
40 }
```

(c)

```
1 #include <string>
2 #include <iostream>
3 using namespace std;
4
5 class C1 {
6 public:
7     virtual string getDescription();
8 };
9
10 class C2 : public C1 {
11 public:
12     virtual string getDescription();
13 };
14
15 string C1::getDescription(){
16     return "C1";
17 }
18
19 string C2::getDescription(){
20     return "C2";
21 }
22
23 int main() {
24     C1 c1;
25     C2 c2;
26     c1 = c2;
27     C1* pC1 = &c2;
28
29     cout << c1.getDescription() << endl;
30     cout << c2.getDescription() << endl;
31     cout << pC1->getDescription() << endl;
32     pC1 = new C2();
33     cout << pC1->getDescription() << endl;
34 }
```

(d)

```
1 #include <string>
2 #include <iostream>
3 using namespace std;
4
5 class C1 {
6 public:
7     string getDescription();
8 };
9
10 class C2 : public C1 {
11     string getDescription();
12 };
13
14 string C1::getDescription() {
15     return "C1";
16 }
17
18 string C2::getDescription() {
19     return "C2";
20 }
21
22 int main() {
23     C1* c1 = new C1();
24     C1* c2 = new C2();
25
26     cout << c1->getDescription() << endl;
27     cout << c2->getDescription() << endl;
28 }
```

Aufgabe 4 (24 Punkte)

(a) Schreiben Sie folgende Funktion im Stil der Algorithmen der Standardbibliothek:

```
1 template <typename Iterator, typename ElemType>
2 ElemType groesstes(Iterator begin, Iterator end,
3     bool(*groesser)(ElemType, ElemType));
```

`groesstes` soll das der übergebenen Vergleichsfunktion nach größte Element im Iteratorbereich zurückgeben. Falls der Iteratorbereich kein Element enthält, soll die Funktion eine Ausnahme werfen. Schreiben Sie außerdem eine Beispielanwendung Ihrer Funktion, die für eine Sammlung von Strings den längsten dieser Strings ermittelt.

Hinweis: Eine Vergleichsfunktion für Ganzzahlen könnte so aussehen:

```
bool intgroesser(int a, int b) {
    return (a > b);
}
```

(b) Gegeben sei folgende Template-Klasse, die einen binären Suchbaum realisiert:

```
1 template <typename T> class TreeNode {
2 private:
3     T wert;
4     TreeNode<T>* leftChild;
5     TreeNode<T>* rightChild;
6 public:
7     TreeNode(T w) : wert(w), leftChild(0), rightChild(0) {}
8 };
```

Jeder Knoten ist Eigentümer seiner Kindknoten, sofern diese existieren. Ergänzen Sie die Klasse um

1. die Methode `bool contains(T t) const`; die zurückgibt ob der Baum den Wert `t` enthält. Nutzen Sie dabei die Sortierung des Baums, um unnötigen Suchaufwand zu vermeiden.
2. die Methode `void postorder(std::vector<T>& result) const`; die die im Baum gespeicherten Werte in Postorder-Reihenfolge in `result` einfügt.
3. einen geeigneten Destruktor.

Aufgabe 5 (20 Punkte)

(a) Wenn Sie in C++ eine Methode nicht als `virtual` markieren, findet für diese kein Late Binding statt. Worin kann dabei der Vorteil liegen?

(b) Wie werden in C++ Template-Klassen vom Compiler übersetzt? Wie unterscheidet sich dies von den Generics in Java?

(c) Was versteht man in C++ unter einem Default-Konstruktor?

(d) Wie kann das Schlüsselwort `explicit` in C++ verwendet werden, und was bewirkt es? Wann ist sein Einsatz sinnvoll?

(e) Das Sternsymbol `*` hat in C++ mehrere Bedeutungen. Welche?

(f) Welche Arten der Instanziierung von Templates gibt es in C++? Wie unterscheiden sie sich in der Organisation der Quelltexte, und welche Vor- und Nachteile resultieren daraus?

(g) Warum sollte in C++ der Datentyp einer `catch`-Klausel immer ein Referenztyp sein?

(h) Aus welchen Schritten besteht das Übersetzen eines C++-Quelltexts zu einem ausführbaren Programm? Was passiert bei den einzelnen Schritten?

(i) Das Exception-Handling von C++ kennt (im Gegensatz zu Java) keine `finally`-Klausel. Warum wird sie nicht benötigt?

(j) Welcher Zusammenhang besteht in C++ zwischen Destruktor, Kopierkonstruktor und Zuweisungsoperator?