

Name

Matr.-Nr.

Klausur Programmieren III – TI3

WS 13

Prof. Dr. K. Baer

Hilfsmittel: 1 Blatt DIN A4, beidseitig beschr.

Bearbeitungszeit: 90 Min.

1. Bitte tragen Sie zuerst Name und Matrikelnummer ein!
2. Kontrollieren Sie die Vollständigkeit der Aufgabenblätter.
3. Die Klausur besteht aus 6 Aufgaben. Verschaffen sie sich einen kurzen Überblick über die Aufgaben und beginnen Sie am besten mit der Aufgabe, die Ihnen am ehesten ein Erfolgserlebnis bringt.
4. Lesen Sie die Aufgabenstellung aufmerksam durch, bevor Sie eine Aufgabe lösen!
5. Verwenden Sie zur Beantwortung der Fragen den vorgesehenen Platz auf den Aufgabenblättern.
6. Schreiben Sie leserlich. Nicht lesbare Teile werden mit 0 Punkten bewertet!

Viel Erfolg!

Aufgabe	1	2	3	4	5	6	Summe
Punkte	20	16	18	14	10	12	90
erreicht							

Aufgabe 1 (20 Punkte = 16+2+2)

Folgende Klassendefinition sei gegeben:

```
class Top {
public:
    Top()          { cout << " cTop"; }
    Top(Top& a)    { cout << " copyTop"; }
    ~Top()         { cout << " ~Top"; }
    void doSomething(){ cout << " doTop"; }
};

class A : public Top {
public:
    A()          { cout << " cA"; }
    A(A& a)      { cout << " copyA"; }
    ~A()         { cout << " ~A"; }
    void doSomething(){ cout << " doA"; }
};

class B : public Top {
private:
    Top* ptrTop;
    A objA;
public:
    B()          { cout << " cB"; }
    B(B& a)      { cout << " copyB"; }
    ~B()         { cout << " ~B"; }

    Top* getptrTop() {return ptrTop; }
    void setptrTop(Top* t){ ptrTop = t; }

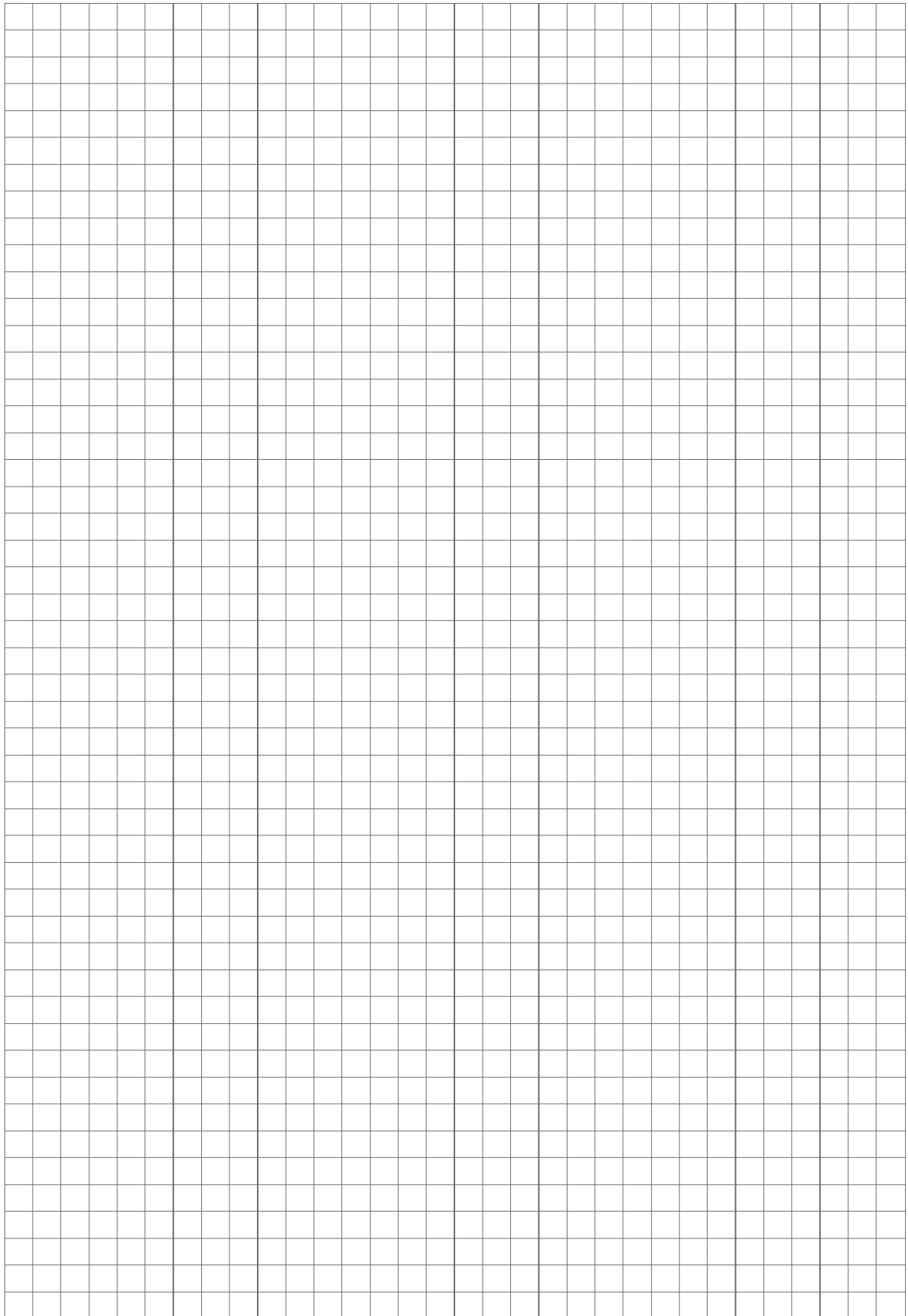
    A getobjA() {return objA; }
    void setA(A a){objA = a; }
};
```

- a) Geben Sie für untenstehende Funktion `test()` hinter jeder Zeile an, welche Ausgaben bei der Auswertung auf der Konsole erscheinen.

Nr	Code	Ausgabe
1	void test(){	
2	Top t[3];	
3	A a1;	
4	B* b1 = new B();	
5	t[1] = a1;	
6	t[2] = *b1;	
7	A a2 = a1;	
8	b1->setptrTop(&a2);	
9	t[1].doSomething();	
10	t[2].doSomething();	
11	a2.doSomething();	
12	b1->doSomething();	
13	b1->getptrTop()->doSomething();	
14	b1->getobjA().doSomething();	
15	delete b1;	
16	}	

A full-page sheet of white graph paper with a light gray grid. The grid consists of small squares, approximately 10 units wide by 10 units high. There are no margins or additional markings on the page.

A full page of blank graph paper with a light gray background and a uniform grid of thin gray lines. The grid consists of small squares covering the entire area. There are no margins, text, or other markings on the page.



Aufgabe 2 (16 Punkte=4+4+6+2)

- a) Was versteht man unter Elementinitialisierer-Ausdrücken und wozu dienen sie? Geben Sie ein Beispiel.

- b) Welche Probleme können bei expliziten Typumwandlungen (Casts) im Zusammenhang mit abgeleiteten Klassen auftreten?

Aufgabe 3 (18 Punkte=5+3+10)

Aus zwei einfach verketteten Listen A und B soll eine neue Liste konstruiert werden, wobei abwechselnd jeweils aus A und dann aus B ein Element an die neue Liste angehängt wird, bis alle Elemente aus A und B in der resultierenden Liste enthalten sind. Falls das Ende einer Liste erreicht wird, werden die restlichen Elemente der anderen Liste angehängt.

Beispiel:

Liste A:

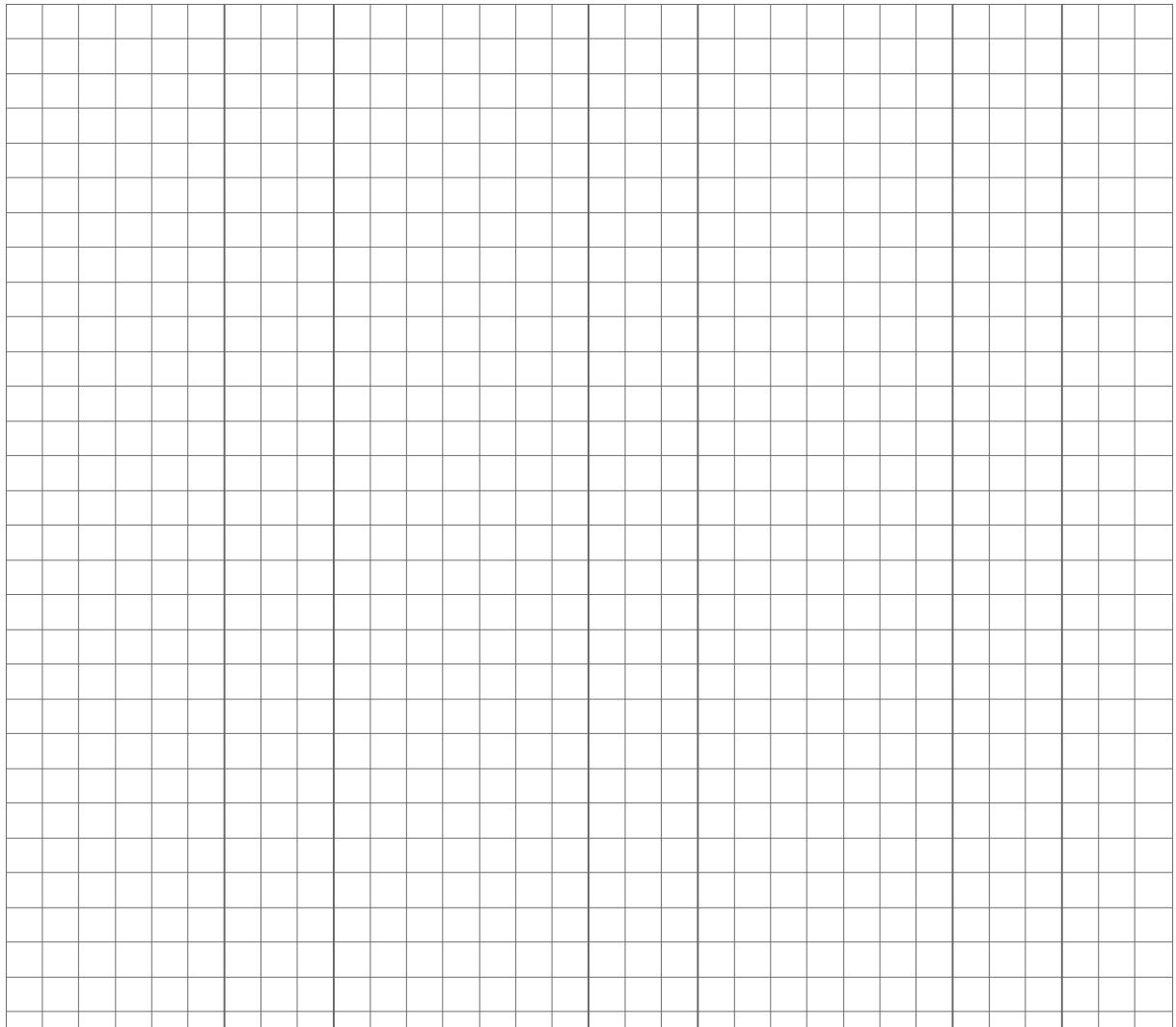
10	11	12	13	14	15	16	17
----	----	----	----	----	----	----	----

Liste B	20	21	22	23	24	25	26	27	28	29
---------	----	----	----	----	----	----	----	----	----	----

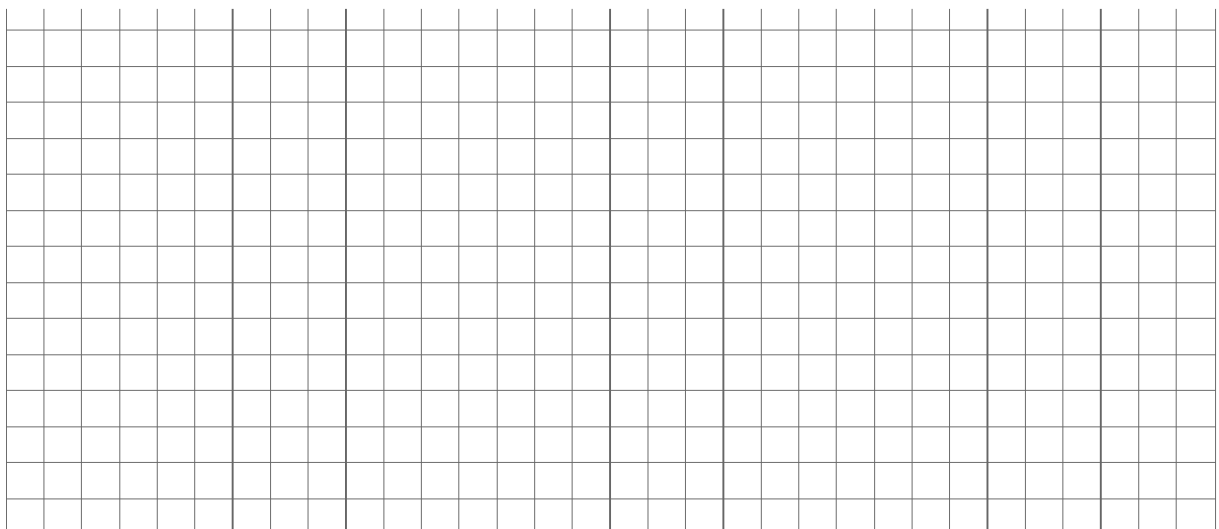
Resultat	10	20	11	21	12	22	13	23	14	24	15	25	16	26	17	27	28	29
----------	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

- a) Entwickeln Sie eine geeignetes Klassen-Templete für diesen Listentyp. Implementieren Sie lediglich die für die beschriebene merge-Operation und den Auf-/Abbau der Liste benötigten Methoden (Konstruktor, Destruktor, Get-/Set-Methoden) sowie ihre Attribute.

A full-page sheet of white graph paper with a light gray grid. The grid consists of small squares, approximately 10 units wide by 10 units high. There are no margins or additional markings on the page.



- b) Überladen Sie den << Operator für Ihre Listen-Klasse. Die Ausgabe soll die Werte der Liste getrennt durch Minuszeichen ausgeben, z.B:
10 – 43 – 21 – 23 – 54



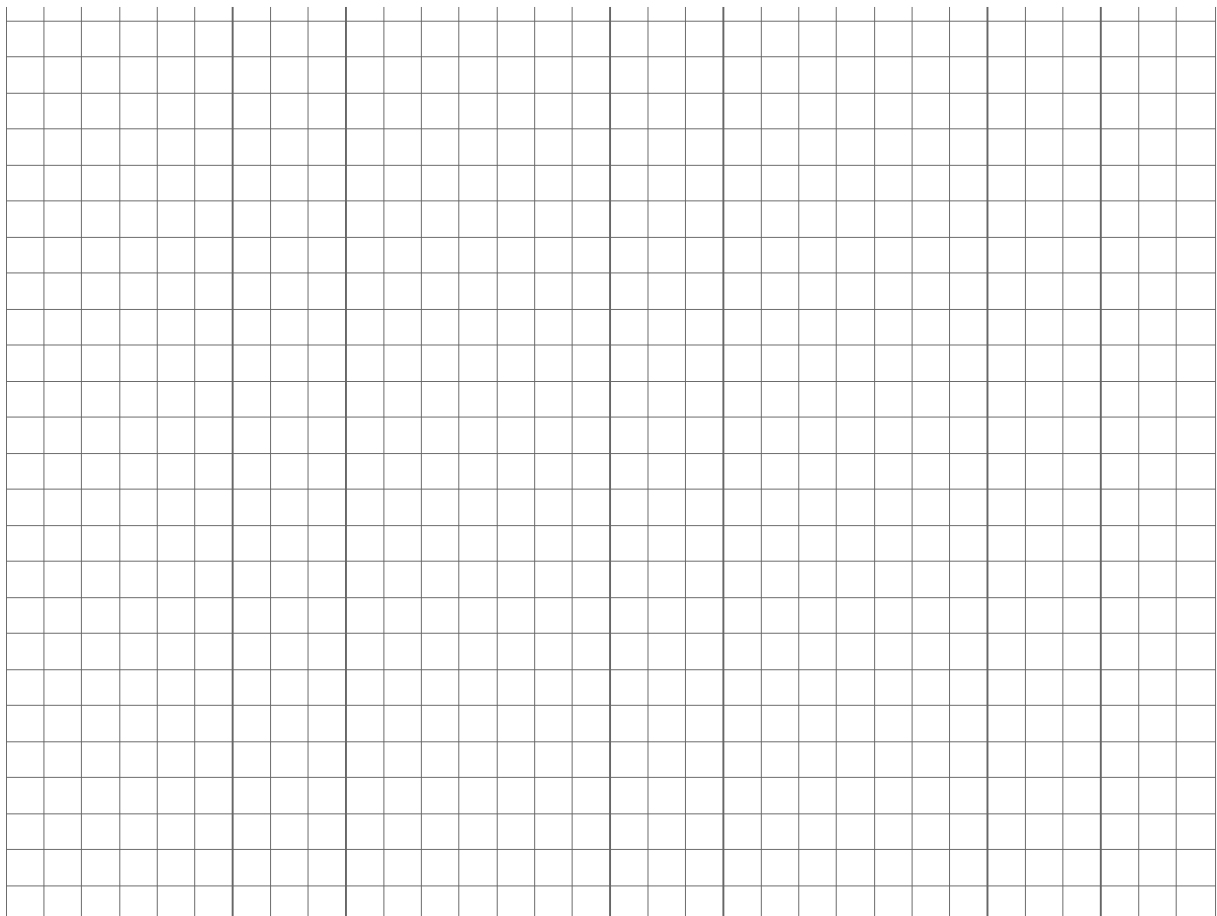
- c) Schreiben Sie eine globale Template-Methode `merge`, die zwei einfach verkettete Listen `listA` und `listB` Ihrer Klasse zu einer neuen Liste `result` entsprechend obiger Vorgabe zusammenfügt. Die Elemente von `listA` und `listB` sollen nicht kopiert, sondern nach `result` verschoben werden. Die übergebenen Zeiger auf die Listen `listA`, `listB` sollen danach auf `NULL` zeigen. Ihre Methode `merge` könnte z.B. wie folgt in einer `main()` aufgerufen werden:

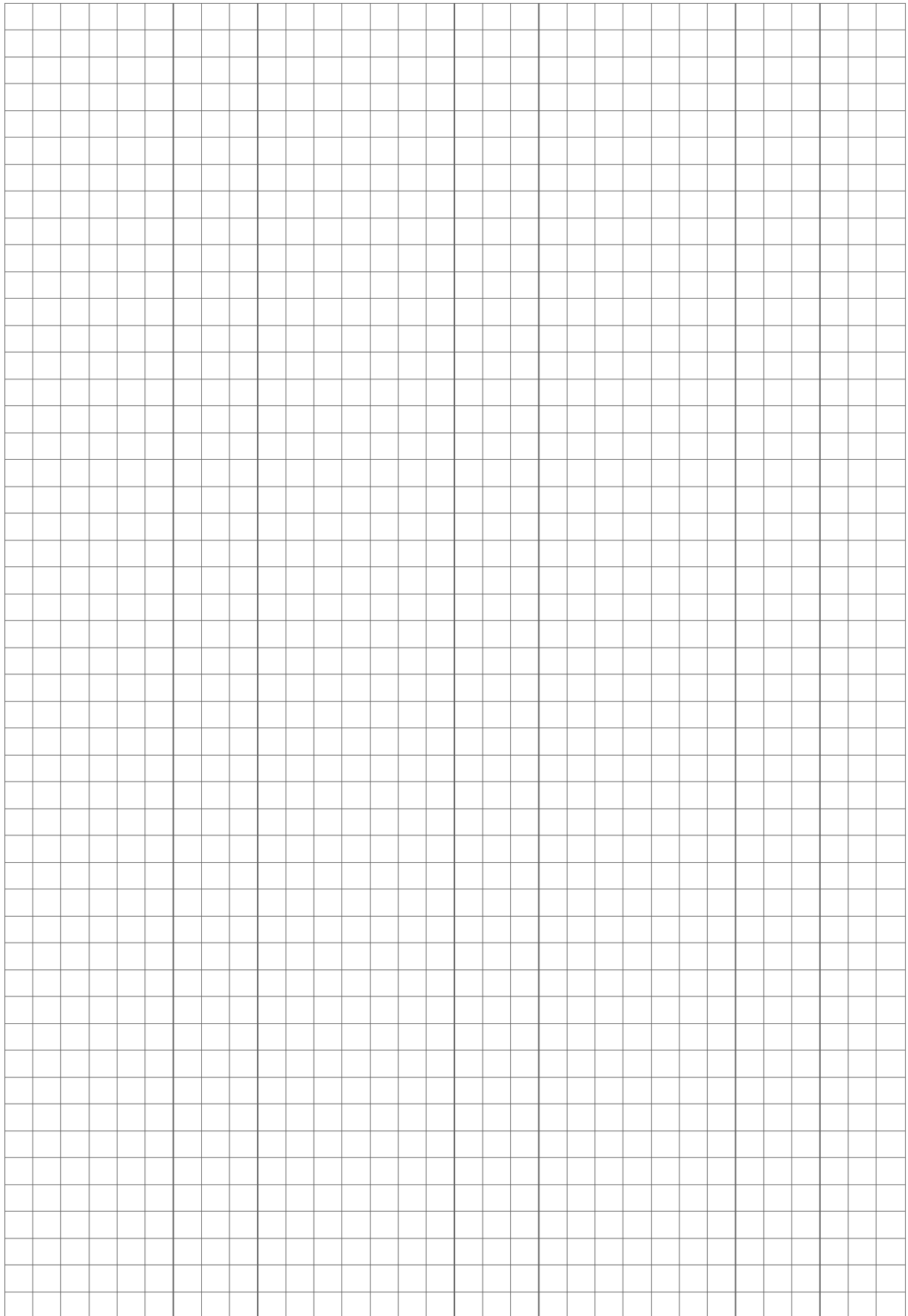
```
int main() {
    SimpleList<int> * listA = new SimpleList<int>(10);
    // füllen der Liste

    SimpleList<int> * listB = new SimpleList<int>(20);
    // füllen der Liste

    SimpleList<int> * mergedList = merge(&listA, &listB);
    cout << "mergedList nach Merge: " << (*mergedList);

    if (!listA)
        cout << "\n" << "listA ist leer" << endl;
    if (!listB)
        cout << "\n" << "listB ist leer" << endl;
    //...
    delete mergedList;
} // main( )
```





Aufgabe 4 (14 Punkte = 8+3+3)

In einem Programm zur Artikelverwaltung müssen die gespeicherten Artikel häufig nach ganz unterschiedlichen Kriterien sortiert werden. Bspw. müssen Artikel

- nach ihrer Bezeichnung alphabetisch aufsteigend oder absteigend
- nach ihrem Preis aufsteigend oder absteigend.

sortiert werden können.

Die STL bietet Ihnen hierzu einiges an Unterstützung. Sie können die Liste der Artikel in einem Container verwalten und mittels des `sort()`-Algorithmus den Container nach den gewünschten Kriterien sortieren.

Den Algorithmus `sort()` gibt es in 2 Varianten:

```
template<class RA_Iter>
void sort( RA_Iter first,  RA_Iter last);

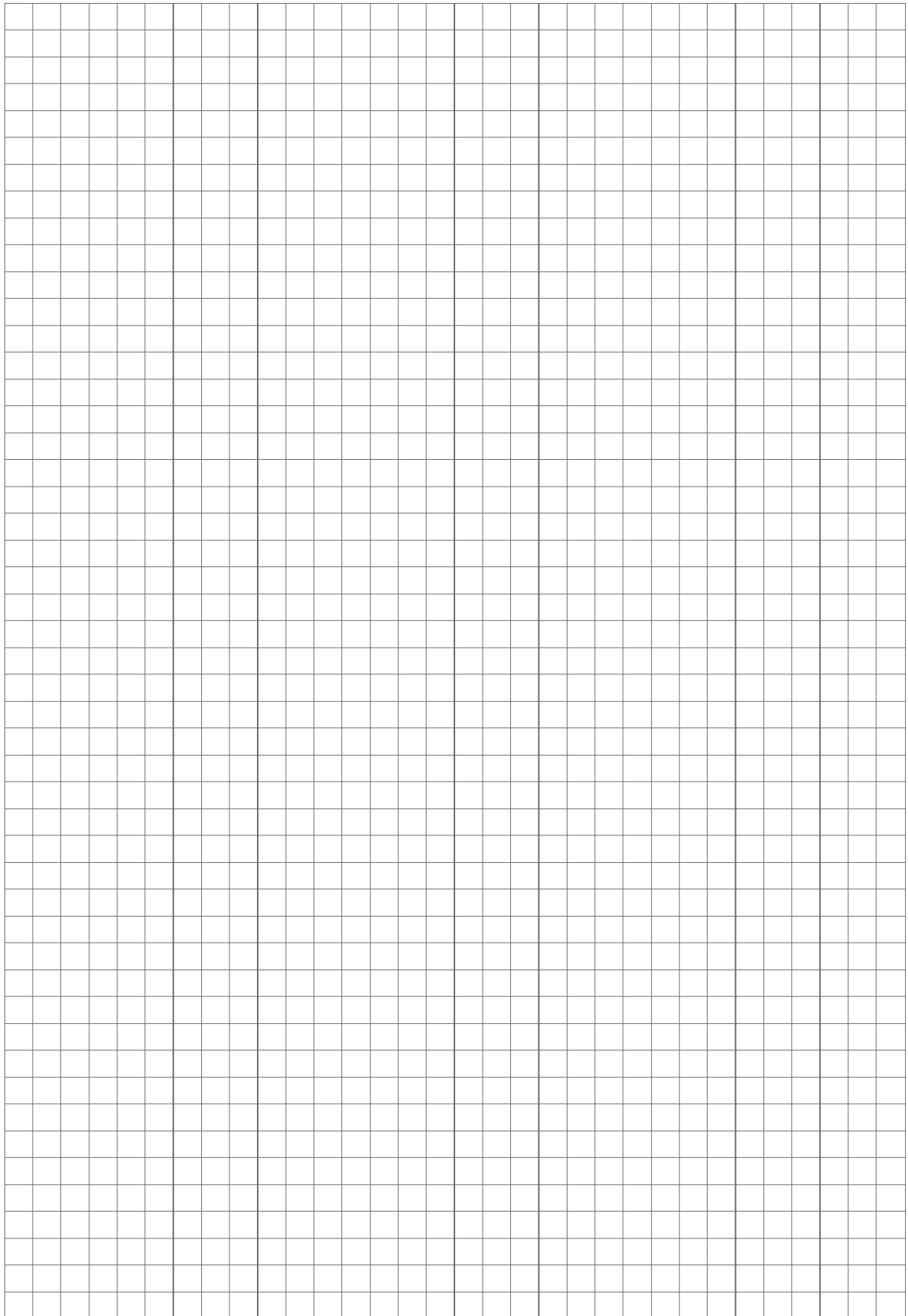
template<class RA_Iter, class Compare>
void sort( RA_Iter first,  RA_Iter last, Compare comp);
```

`sort()` sortiert die Elemente des Bereichs `[first, last)` in Bezug auf `operator<`, wenn kein Vergleichskriterium angegeben ist, und ansonsten in Bezug auf `comp`.

- a) Implementieren Sie eine Klasse `Artikel` mit den **privaten** Attributen
- ```
string bezeichnung
double preis
```
- (andere Attribute sollen vernachlässigt werden), sowie allen benötigten Methoden, Operatoren, Funktoren, um die oben genannten Sortierungen mit Hilfe des `sort()`-Algorithmus produzieren zu können.

Die Standardsortierung – also die 1. Variante ohne zusätzliches Funktionsionobjekt – soll eine alphabetisch aufsteigende Sortierung nach Bezeichnung sein.

[illegible]



- b) Ergänzen sie folgendes kleines Testprogramm um Ihre Funktionsaufrufe gemäß den Angaben in den Kommentaren. Um Schreibarbeit zu sparen, geben Sie nur die Veränderungen zur vorherigen Angabe an.

```
void test(){
 vector<Artikel> b; // Liste von Artikeln

 // Einfuegen von Buechern
 b.push_back(Artikel("C-Artikel", 40));
 b.push_back(Artikel("A-Artikel", 50));
 b.push_back(Artikel("B-Artikel", 30));

 // Ausgeben der unsortierten Liste

 // alphabetisch aufsteigend sortieren & ausgeben:

 // alphabetisch absteigend sortieren & ausgeben:

 // nach Preis aufsteigend sortieren und ausgeben:

 // nach Preis absteigend sortieren und ausgeben:

}
```

- c) Was muss ergänzt werden, um die Ausgabe mittels `copy` und eines Stream-Iterators machen zu können? Ergänzen Sie Ihre Implementierung um die notwendigen Dinge, falls nicht ohnehin schon geschehen.

Die Ausgabe soll in folgender Form erfolgen:

Bezeichnung: "C-Artikel" Preis: 40

Bezeichnung: "A-Artikel" Preis: 50

Bezeichnung: "B-Artikel" Preis: 30

A full page of blank graph paper. The grid consists of small squares formed by thin gray lines. There are 20 columns and 20 rows of squares, creating a total area of 400 small squares. The grid covers the entire page except for a narrow margin at the top.

### Aufgabe 5 (10 Punkte = 3 + 1 + 6)

Folgendes Programm sei gegeben:

```
int main() {
 typedef multimap<string, string> VokMulMap;
 typedef VokMulMap::value_type V;
 const V f[] = { V("two", "Zwei"), V("three", "Drei"),
 V("four", "Vier"), V("four", "Doppelvierer")};

 VokMulMap vokabeln(f, f + 4);

 const VokMulMap::iterator ins = vokabeln.insert(V("one",
 "Eins"));
 vokabeln.insert(ins, V("one", "man"));

 VokMulMap::const_iterator iter = vokabeln.begin();
 for (; iter != vokabeln.end(); ++iter)
 cout << "(" << iter->first << ", " << iter->second << ")\n";

 const VokMulMap::const_iterator
 von = vokabeln.lower_bound("one"),
 bis = vokabeln.upper_bound("one");
 for (iter = von; iter != bis; ++iter)
 cout << iter->second << ' ';
 cout << endl;
 const pair<VokMulMap::iterator, VokMulMap::iterator>
 p = vokabeln.equal_range("four");
 for (iter = p.first; iter != p.second; ++iter)
 cout << iter->second << ' ';

 VokMulMap::size_type n = vokabeln.count("one");
 cout << '\n' << n << endl;
 n = vokabeln.erase("one");
 cout << n << endl;
}
```

a) Was gibt das Programm aus?

[illegible]



b) Welchen Datentyp nutzt eine Map/MultiMap?

[illegible]

c) Wie kann man z.B. nach dem Wert „vier“ suchen, wenn man den zugehörigen Schlüssel nicht kennt?

Denken Sie an den `find_if`-Algorithmus, der folgendermaßen deklariert ist:

```
template<class T, class Predicate>
InputIterator find_if(InputIterator first,
 InputIterator last,
 Predicate pred);
```

Was muss man tun, um mit Hilfe dieses Algorithmus nach einem bestimmten Wert zu suchen?

Hinweis:

Die Template-Klasse `pair` stellt Typdefinitionen bereit, mit denen der Typ des ersten und zweiten Elements ermittelt werden kann:

```
typedef T1 first_type;
typedef T2 second_type;
```

[illegible]

[ 6 P ]

[illegible]

## Aufgabe 6 (12 Punkte)

Entwickeln Sie ein Programm, das die Wörter in einem übergebenen Text zählt und ausgibt.


Hierfür sind Stream-Iteratoren sehr nützlich. Der `istream_iterator` hat verschiedene Konstruktoren:

```
istream_iterator();
```

Der Default-Konstruktor erzeugt einen end-of-stream Iterator.

```
istream_iterator (istream_type& s);
```

erzeugt einen Iterator, der mit einem Eingabe-Strom `s` verknüpft ist, von dem Sie den Text lesen können.

A large grid of graph paper, consisting of 20 columns and 30 rows of small squares, intended for the student to write their code solution.