Klausur
Programmieren 2
für ElektrotechnikerInnen
SS 2001

Prüfer: Dr. H. Joseph

Name: .....

Vorname: .....

Matr.-Nr.: .....

Aufgabe	Punkte	Erreichte Punkte
1	15	
2	25	
3	24	
4	12	
Summe		
Note		

Lesen Sie die Aufgabenstellung sorgfältig durch und bearbeiten Sie die Aufgaben gewissenhaft, eindeutig und vollständig.

Hilfsmittel: Ein Buch Ihrer Wahl und Vorlesungsunterlagen (Mitschrift, Skript und Folien).

Es sind KEINE elektronischen Hilfsmittel wie z.B. Taschenrechner, Laptop, PDA oder ähnliches erlaubt!!!

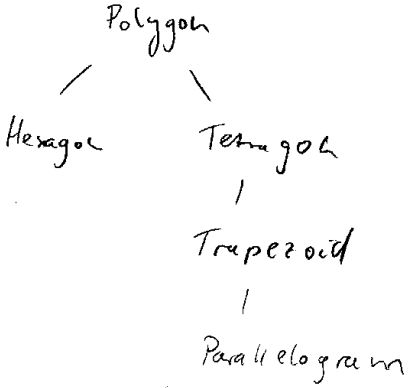
Aufgabe 1 *(Vererbung, virtuelle Funktionen)*

Gegeben sei folgendes Programm:

```
class Polygon
  {
     protected:
        int edges;

     public:
       Polygon (){};
       Polygon (int i) {edges =i;}
       virtual void print()= 0;
  };

class Tetragon : public Polygon
  {
    public:
            Tetragon (){edges = 4;};
            void print() {cout<<"Polygon mit "<<edges<<" Kanten: allg.Viereck\n";};
  };

class Hexagon : public Polygon
  {
    public:
      Hexagon() {edges = 6;}
      void print() {cout<<"Polygon mit "<<edges<<" Kanten: Sechseck\n";}
  };

class Trapezoid : public Tetragon
  {
    public:
      Trapezoid() : Tetragon(){};
      void print() {cout<<"Polygon mit "<<edges<<" Kanten: Trapez\n";};
  };

class Parallelogram : public Trapezoid
  {
    public:
      Parallelogram() : Trapezoid(){};
      void print() {cout<<"Polygon mit "<<edges<<" Kanten: Parallelogramm\n";};
  };
```

```
main ()
 {
   Polygon *pp;
   Hexagon hn;
   Tetragon tn;
   Parallelogram pg;

   hn.print();

   tn = pg;
   tn.print();

   pp = &pg;
   pp->print();

   Trapezoid * tdp = new Trapezoid;
   tdp->print();

   tn  = *tdp;
   tn.print();
}
```

1. Geben Sie die Klassenhierarchie an!
2. Wie lauten die Ausgaben des Hauptprogramms?



5

? Polygon mit 6Kanten: Sechseck
? " " 4 " : allg. Viereck
? " " " " : Parallelogramm
? " " " " : Trapez
? " " " " : allg. Viereck

10

Aufgabe 2 *(Klassenkonzept)*

Entwerfen sie eine generische Klasse (Template) *Queue*!

Eine Queue wird auch als Warteschlange oder FIFO-Speicher bezeichnet. Elementarer Bestandteil sind die Methode *put,* die ein Objekt an die Warteschlange anhängt und die Methode *get,* die das erste Element aus der Queue entnimmt.

Die Länge der Warteschlange soll beschränkt, aber parametrisierbar sein. Bei Erreichen der Maximallänge wird jedes weitere Anhängen freundlich abgelehnt, bis wieder freie Plätze zur Verfügung stehen.

Überlegen Sie, welche Methoden notwendig und/oder sinnvoll sind!

Schreiben Sie ein (kurzes) Testprogramm, mit dem die (generische) Klasse getestet werden kann!

Aufgabe 3 ***(Komplexität der Matrixmultiplikation)***

1. Entwerfen Sie in Form eines Struktogramms ein Unterprogramm, das (quadratische, nxn-) Matrizen multipliziert (Eingabe 2 Matrizen, Ausgabe: das Produkt).

2. Bestimmen Sie für dieses Unterprogramm die Komplexität, gemessen an der Zahl der auszuführenden Additionen, wobei eine Multiplikation im Aufwand 3 Additionen entsprechen soll.

3. Zu welcher Komplexitätsklasse gehört dieses Programm?

Hinweis: Gehen Sie davon aus, dass die Matrizen als zweidimensionale Felder vorliegen.

Aufgabe 3.)

1.1

Eingabe matrizen: E1[n,n] E2[n,
Ausgabe matrix: A[n,n]

für alle Zeilenindices $1 \leq i \leq n$
  für alle Spaltenelemente einer Zeile $1 \leq j \leq n$
    A[i,j] mit 0 initialisieren
    für alle Werte $1 \leq k \leq n$
      A[i,j] = A[i,j] + E1[i,k] + E2[k,j]

12

2.)

$$T_{n\,mult} = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} (c_{add} + c_{mult})$$

$$= 4\, c_{add} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} 1$$

$$= 4\, c_{add} * n^3$$

$$= 4n^3\, c_{add}$$

8

3.) Die Matrix-Multiplikation hat eine polynomielle Komplexität von n^3, wobei n die Anzahl der Zeilen/Spalten ist.

4

Aufgabe 4 (Pointer, Referenzen)

Wie lauten die Ausgaben des folgenden Programm:

```
int subp(int, int);
int subp(int*, int*);
int &sub (int&, int&);

main ()
 {
   int a, b, c;
   int &d = b;

   a=3; b=4; c=5; d=6;

   d=subp(a,b);
   printf ("a: %d, b: %d, c: %d, d: %d\n", a,b,c,d);

   d=subp(&a, &c);
   printf ("a: %d, b: %d, c: %d, d: %d\n", a,b,c,d);

   sub (a,c) = 7;
   printf ("a: %d, b: %d, c: %d, d: %d\n", a,b,c,d);
}

int subp (int x, int y)
 {return x+y;}

int subp(int *x, int *y)
  {return (*x = *x+*y);}

int & sub(int &x, int &y)
  {
    x=4; y=3;
    return (x<y) ? y : x;
  }
```

Ausgaben:

1. Ausgabe: a= 3 b= 9 c= 5 d= 9
2. Ausgabe: a= 8 b= 8 c= 5 d= 8
3. Ausgabe: a= 7 b= 8 c= 3 d= 8

12