**Familienname:**

**Vorname:**

**Matrikelnummer:**

Aufgabe 1 (3 Punkte):
Aufgabe 2 (4 Punkte):
Aufgabe 3 (2 Punkte):
Aufgabe 4 (2 Punkte):
Aufgabe 5 (2 Punkte):
Aufgabe 6 (1 Punkte):
Aufgabe 7 (3 Punkte):
Aufgabe 8 (1 Punkte):
Aufgabe 9 (1 Punkte):
Aufgabe 10 (2 Punkte):
Aufgabe 11 (3 Punkte):
Aufgabe 12 (2 Punkte):
Aufgabe 13 (5 Punkte):
Aufgabe 14 (3 Punkte):
Aufgabe 15 (4 Punkte):
Aufgabe 16 (2 Punkte):

Gesamtpunktzahl:

# Schriftlicher Test (120 Minuten)
# VU Einführung ins Programmieren für TM

## 25. Jänner 2016

**Aufgabe 1 (3 Punkte).** Was sind die Bestandteile $M$, $e_{\min}$, $e_{\max}$ des Gleitkommazahlsystems $\mathbb{F}(2, M, e_{\min}, e_{\max})$? Wie lässt sich jede Gleitkommazahl $x \in \mathbb{F}(2, M, e_{\min}, e_{\max})$ darstellen? Welchen Wert haben die größte und die kleinste positive normalisierte Gleitkommazahl im `double`-Gleitkommazahlsystem $\mathbb{F}(2, 53, -1021, 1024)$?

**Lösung zu Aufgabe 1.**

**Aufgabe 2 (4 Punkte).** Was ist der Shell-Output des folgenden Programms?

```
#include <iostream>
using std::cout;

class dp {
private:
  int x;
  int y;
public:
  dp(int x, int y) {
    this->x = x;
    this->y = y;
    cout << "new: x=" << x << ", y=" << y <<"\n";
  }
  ~dp() {
    cout << "old: x=" << x << ", y=" << y <<"\n";
  }
  dp(const dp& input) {
    int a = input.x;
    int b = input.y;
    int c = 0;
    while ( a != b ) {
      if ( a < b ) {
        c = a;
        a = b;
        b = c;
      }
      a = a - b;
      cout << "a=" << a << ", b=" << b << ", c=" << c << "\n";
    }
    x = input.x/a;
    y = input.y/b;
  }
};


int main() {
  dp q(20,45);
  dp r = q;
  return 0;
}
```

**Lösung zu Aufgabe 2.**

**Aufgabe 3 (2 Punkte).** Der folgende C++ Code hat 4 verschiedene Syntax-Fehler. Markieren Sie diese und erläutern Sie, was warum falsch ist!

```cpp
#include<iostream>
#include<string>

using std::cout;
using std::string;

class Base {

private:
  int a;
public:
  Base(int init) : a(init) {};
  void printinfo() {
    cout << "Base, a=" << a << endl;
  };
  void printinfo(string input) {
    cout << "Base, a=" << a << ", input= " << input << endl;
  };
};

class Derived::public Base {

private:
  int b;
public:
  Derived(int init1, int init2):Base(init1) {
    b = init2;
  };
  void printinfo() {
    cout << "Derived, a=" << a << ", b = " << b << endl;
  };
};

int main() {
  string ah = "Stefan und Michele!";
  Base bs(5);
  Derived gg(5,10);
  Derived dp(12,15);
  bs.printinfo(ah);
  gg.printinfo();
  dp.Derived::printinfo();
  gg.printinfo(ah);
  return 0;
}
```

**Lösung zu Aufgabe 3.**

**Aufgabe 4 (2 Punkte).** Was ist eine rekursive Funktion und was darf dabei nicht fehlen? Erläutern Sie das Konzept anhand eines selbstgewählten Beispiels und geben Sie einen entsprechenden C/C++ Code an!

**Lösung zu Aufgabe 4.**

**Aufgabe 5 (2 Punkte).** Eine obere Dreiecksmatrix $A \in \mathbb{R}^{n\times n}$ ist eine Matrix mit der Eigenschaft $A_{jk} = 0$ für $j > k$, d.h.

$$A = \begin{pmatrix} A_{00} & A_{01} & A_{02} & \dots & A_{0,n-1} \\ 0 & A_{11} & A_{12} & \dots & A_{1,n-1} \\ 0 & 0 & A_{22} & \dots & A_{2,n-1} \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & A_{n-1,n-1} \end{pmatrix}$$

Zur effizienten Speicherung wird $A \in \mathbb{R}^{n\times n}$ in Form eines Vektors $a \in \mathbb{R}^N$ mit $N = \sum_{j=1}^n j = \frac{n(n+1)}{2}$ abgelegt, d.h. $A_{jk} = a_\ell$ für einen geeigneten Index $\ell$, der eindeutig von $j$ und $k$ abhängen muss. Leiten Sie eine Formel für $\ell$ her (in Abhängigkeit von $j$ und $k$). Begründen Sie Ihre Formel.

**Lösung zu Aufgabe 5.**

**Hinweis.** In den folgenden Aufgaben seien die oberen Dreiecksmatrizen $A \in \mathbb{R}^{n \times n}$ in Objekten der C++ Klasse `TriMatrix` gespeichert, die unten definiert ist. Neben Konstruktor (mit optionalem Initialisierungswert), Kopierkonstruktor, Destruktor und Zuweisungsoperator gibt es eine Methode, um die Dimension $n$ auszulesen (`size`). Der Koeffizientenvektor `coeff` speichert nur die $\frac{n(n+1)}{2}$ nicht-trivialen Einträge $A_{jk}$ mit $j \le k$, auf die mittels `A(j,k)` lesend und schreibend zugegriffen wird:

```
1  class TriMatrix {
2  private:
3    int n;
4    double* coeff;
5  public:
6    TriMatrix(int n=0, double init=0);
7    TriMatrix(const TriMatrix&);
8    ~TriMatrix();
9    TriMatrix& operator=(const TriMatrix&);
10   int size() const;
11   const double& operator()(int j, int k) const;
12   double& operator()(int j, int k);
13 };
```

**Aufgabe 6 (1 Punkt).** Erläutern Sie die Bedeutung der beiden `const` in Zeile 11 der Klassendefinition.

**Lösung zu Aufgabe 6.**

**Aufgabe 7 (3 Punkte).** Schreiben Sie den Konstruktor der Klasse `TriMatrix`. Stellen Sie mittels `assert` sicher, dass $n \geq 0$ ist, wobei für $n = 0$ eine leere Matrix angelegt werde.
**Hinweis.** Beachten Sie, dass `coeff` ein Vektor der Länge $\frac{n(n+1)}{2}$ ist.

**Lösung zu Aufgabe 7.**

**Aufgabe 8 (1 Punkt).** Schreiben Sie den Destruktor der Klasse `TriMatrix`.

**Lösung zu Aufgabe 8.**

**Aufgabe 9 (1 Punkt).** Schreiben Sie die Methode `size` der Klasse `TriMatrix`.

**Lösung zu Aufgabe 9.**

**Aufgabe 10 (2 Punkte).** Schreiben Sie den Koeffizientenzugriff der Klasse `TriMatrix` für `const`-Objekte. Stellen Sie mittels `assert` sicher, dass für $A \in \mathbb{R}^{n\times n}$ die Indizes $0 \leq j \leq k \leq n-1$ erfüllen.
**Hinweis.** Verwenden Sie Ihre Formel aus Aufgabe 5.

**Lösung zu Aufgabe 10.**

**Aufgabe 11 (3 Punkte).** Schreiben Sie den Zuweisungsoperator der Klasse `TriMatrix`.

**Lösung zu Aufgabe 11.**

**Aufgabe 12 (2 Punkte).** Beweisen Sie mathematisch, dass das Produkt $C = AB \in \mathbb{R}^{n\times n}$ zweier oberer Dreiecksmatrizen $A, B \in \mathbb{R}^{n\times n}$ wieder eine obere Dreiecksmatrix ist, indem Sie die Laufindizes der Summe des allgemeinen Matrizenprodukts

$$C_{j\ell} = \sum_{k=0}^{n-1} A_{jk} B_{k\ell} \quad \text{für } j, \ell = 0, \ldots, n-1$$

mithilfe der Dreiecksstruktur von $A$ und $B$ vereinfachen.
**Hinweis:** Eine obere Dreiecksmatrix $A \in \mathbb{R}^{n\times n}$ ist durch $A_{jk} = 0$ für $j > k$ charakterisiert.

**Lösung zu Aufgabe 12.**

**Aufgabe 13 (5 Punkte).** Überladen Sie den `*` Operator so, dass er das Produkt $C = AB \in \mathbb{R}^{n\times n}$ zweier oberer Dreiecksmatrizen $A, B \in \mathbb{R}^{n\times n}$ berechnet. Stellen Sie mittels `assert` sicher, dass $A$ und $B$ dieselbe Dimension haben.
**Hinweis.** Beachten Sie, dass die Funktion nur Koeffizienten $C_{jk}$ für $0 \le j \le k \le n-1$ berechnen soll und auch nur auf entsprechende Koeffizienten von $A$ und $B$ zugreifen darf. Verwenden Sie dazu Ihre Erkenntnisse aus Aufgabe 12.

**Lösung zu Aufgabe 13.**

**Hinweis.** In den folgenden Aufgaben seien Vektoren $x \in \mathbb{R}^n$ in Objekten der C++ Klasse `Vector` gespeichert, die unten definiert ist. Neben Konstruktor, Kopierkonstruktor, Destruktor und Zuweisungsoperator gibt es eine Methode, um die Dimension $n$ auszulesen (`size`). Auf die Koeffizienten $x_j$ des Vektors kann mittels `x(j)` für $0 \leq j \leq n-1$ zugegriffen werden. Sie müssen keine der genannten Methoden implementieren!

```
class Vector {
private:
  int n;
  double* coeff;
public:
  Vector(int n=0, double init=0);
  Vector(const Vector&);
  ~Vector();
  Vector& operator=(const Vector&);
  int size() const;
  const double& operator()(int j) const;
  double& operator()(int j);
};
```

**Aufgabe 14 (3 Punkte).** Leiten Sie für gegebenes $b \in \mathbb{R}^n$ eine Formel her, um für eine obere Dreiecksmatrix $A \in \mathbb{R}^{n \times n}$ mit $A_{jj} \neq 0$ für alle $j = 0, \ldots, n-1$ die Lösung $x \in \mathbb{R}^n$ von $Ax = b$ zu berechnen, indem Sie die Formel des Matrix-Vektor-Produkts

$$b_j = (Ax)_j = \sum_{k=0}^{n-1} A_{jk} x_k$$

mithilfe der Dreiecksstruktur von $A$ vereinfachen.
**Hinweis:** Eine obere Dreiecksmatrix $A \in \mathbb{R}^{n \times n}$ ist durch $A_{jk} = 0$ für $j > k$ charakterisiert.

**Lösung zu Aufgabe 14.**

**Aufgabe 15 (4 Punkte).** Überladen Sie den `|` Operator so, dass `x = A|b` für eine obere Dreiecksmatrix $A \in \mathbb{R}^{n\times n}$ (vom Typ `TriMatrix`) und einen Vektor $b \in \mathbb{R}^n$ (vom Typ `Vector`) die Lösung $x \in \mathbb{R}^n$ von $Ax = b$ (als Objekt vom Typ `Vector`) berechnet. Stellen Sie mittels `assert` sicher, dass $A$ und $b$ passende Dimension haben.

**Lösung zu Aufgabe 15.**

**Aufgabe 16 (2 Punkte).** Bestimmen Sie den Aufwand Ihrer Funktion aus Aufgabe 15. Falls die Funktion für $n = 10^3$ eine Laufzeit von 3 Sekunden hat, welche Laufzeit erwarten Sie aufgrund des Aufwands für $n = 10^4$? Begründen Sie Ihre Antwort!

**Lösung zu Aufgabe 16.**