Dirk Praetorius
Michele Ruggeri

Sommersemester 2015
02.06.2015

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 10

Aufgabe 10.1. Write a class `Client` that stores a list of deposits. Furthermore, the class should contain an object of the class `Name` from Exercise 9.3. Implement methods for adding and deleting deposits. Moreover, write a function that computes the assets of all deposits. Think of other useful functions. Save your source code as `client.{hpp,cpp}` into the directory `serie10`.

Aufgabe 10.2. Implement the `get` and `set` methods of the class

```
class Fraction {
   long numerator;
   unsigned long denominator;
public:
   Fraction();
   Fraction(long numerator, unsigned long denominator);
   setNumerator(long z);
   setDenominator(unsigned long n);
   double getValue();
};
```

The method `getValue` should return the floating-point value of the fraction. Take care of the fact, that the denominator has to be nonzero. Additionally, implement the constructor `Fraction()` that sets the numerator to 0 and the denominator to 1. Save your source code as `fraction.cpp` into the directory `serie10`.

Aufgabe 10.3. What is the output of the following code? Explain why!

```
#include <iostream>
#include <string>
using namespace std;

class T1 {
   string t1;
public:
   T1(string val) { cout << "I am constructor of " << val << endl; t1=val; }
   T1() { cout << "I am constructor of default" << endl; t1="default"; }
   ~T1() { cout << "I am destructor of " << t1 << endl; }
};

int main() {
   T1 bert("bert");
   T1 bob;
   T1 def("bob");
   return 0;
}
```

Aufgabe 10.4. Erweitern Sie die Klasse Vector aus der Vorlesung mit einer Methode `unique`, die einen Vektor $x \in \mathbb{R}^n$ aufsteigend sortiert und doppelte Einträge streicht. Die Funktion soll also beispielsweise den Vektor $x = (4, 3, 5, 1, 4, 3, 4) \in \mathbb{R}^7$ durch den Vektor $x = (1, 3, 4, 5) \in \mathbb{R}^4$ überschreiben. Speichern Sie den Source-Code unter `unique.cpp` in das Verzeichnis `serie10`.

Aufgabe 10.5. Write a class `Matrix` for the storage of $m \times n$ matrices $A \in \mathbb{R}^{m\times n}$. The entries should be stored columnwise by a `double*`-array of length mn. Write `get/set`-methods for the entries of the matrix and `get`-methods for the dimensions. Moreover, write a constructor with input $m, n \in \mathbb{N}_0$, that allocates memory for a $m\times n$ matrix and initializes all entries with 0. Implement the standard constructor which generates a 0×0-matrix as well as a destructor which frees allocated memory. Save your source code as `matrix.`{`hpp/cpp`} into the directory `serie10`.

Aufgabe 10.6. Extend the class `Matrix` from Exercise 10.5 by the following methods:

- an assignment operator,
- a copy-constructor,
- an operator to be able to perform $-A$,
- a method `scanMatrix(int n,int m)`, which reads a $n \times m$-Matrix from the keyboard,
- a method `printMatrix()` which prints a matrix on screen,
- and a method `transpose`, which overwrites a stored matrix with the transposed matrix. (Hint: The transposed $A^T \in A \in \mathbb{R}^{n\times m}$ of a matrix $A \in \mathbb{R}^{m\times n}$ is defined by the condition $\left(A^T\right)_j k = A_k j$. We then have $\left(A^T\right)^T = A$.)

Moreover, write a main programm to test your implemented methods. Save your source code as `matrix.`{`hpp/cpp`} into the directory `serie10`.

Aufgabe 10.7. What is the output of the following C++ program? Explain why! What are the differences between the different variable types used?

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

const int proc(int & input){input = input*2; return input;}
int proc(const int & input){ int output = input; return output;}

void swap(int& x, int& y){
int tmp;
tmp = x;
x = y;
y = tmp;
}

void swap(const int& x,const int& y){;}

int main() {

int var1 = 1;
int var2 = 2;
int var3 = proc(var1);
int var4 = proc(var2);
const int var5 = proc(var1);
const int var6 = proc(var2);
int var7 = proc(proc(var1));
int var8 = proc(proc(var2));
int& var9 = var1;
int& var10 = var2;
const int& var11 = proc(var1);
const int& var12 = proc(var2);
swap(var3,var4);
swap(var5,var6);
swap(var7,var8);
swap(var9,var10);
swap(var11,var12);

return 0;
}
```

Aufgabe 10.8. Write a `Makefile` for the exercises of this sheet. It should contain:

- The compilation of all solved exercises.
- The generation of a library and an example of its usage.