Dirk Praetorius,
Michele Ruggeri

Wintersemester 2017/18
11./12.01.2018

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 10

Aufgabe 10.1. 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 10.2. Extend the class `Fraction` from slide 230 and Exercise 9.1 by the

- the standard constructor (without parameters), which sets $p = 0$ and $q = 1$,
- a constructor, which takes $p, q \in \mathbb{Z}$ and $q \neq 0$ as input, and stores the corresponding `Fraction`,

- the copy constructor,
- the assignment operator, and
- the destructor.

Use `assert` to ensure a valid input, i.e, $q \neq 0$. Note that, for the case $q < 0$, you have to store the fraction $(-p)/|q|$. Test your implementations appropriately! Save your source code as `Fraction.{hpp/cpp}` into the directory `serie10`.

Aufgabe 10.3. Overload the sign operator $-$, which, given $x \in \mathbb{Q}$ stored as `Fraction` from Exercise 10.2, returns the fraction $-x$, as well as the `<<`-operator, in order to be able to print to the screen a fraction $x := p/q$ in the form `p/q` (see slide 290 for an example with class `Complex` from the lecture). Test your implementation accurately!

Aufgabe 10.4. Overload the $+$, $-$, $*$ and $/$ operator in order to be able to calculate the sum, the difference, the product and the quotient of two fractions stored in the format `Fraction` from Exercise 10.2. For $/$, use `assert` to avoid dividing by 0. In all cases, the result is returned in reduced form. Test your implementations appropriately!

Aufgabe 10.5. Write a class `Polynomial` to save polynomials of degree $n \in \mathbb{N}$, which are represented with respect to the monomial basis, i.e.,

$$p(x) = \sum_{j=0}^{n} a_j x^j.$$

The class contains the dynamical vector $(a_0, \ldots, a_n) \in \mathbb{R}^{n+1}$ of the coefficients `(double*)` as well as the degree $n \in \mathbb{N}$. Implement the following features:

- destructor, constructor to allocate the zero-polynomial of degree n, copy-constructor,
- assignment operator,
- access to the coefficients of the polynomials via `[ ]`, i.e., for $0 \leq j \leq n$ `p[j]` returns the value a_j and
- the possibility to print a polynomial `p` on screen via the syntax `cout << p`.

Keep in mind that, in the copy-constructor, you have to allocate new dynamic memory for the coefficient vector of the output (Deep Copy). Explain why! Moreover, write a main program to test your implementation.

Aufgabe 10.6. The sum of two polynomials is still a polynomial. Implement for the class `Polynomial` from Exercise 10.5 the feature of adding two polynomials p and q via `r=p+q`. Moreover, implement the opportunity to add a number $a \in \mathbb{R}$ stored as `double` or `int` to a polynomial p in an appropriate way via `r=a+p` and `r=p+a`. Write a main program to test your implementations.

Aufgabe 10.7. The product of two polynomials is still a polynomial. Implement for the class `Polynomial` from Exercise 10.5 the feature of multiplying two polynomials p and q via `r=p*q`. Moreover, implement the opportunity to multiply a number $a \in \mathbb{R}$ stored as `double` or `int` with a polynomial p in an appropriate way via `r=a*p` and `r=p*a`. Write a main program to test your implementations.

Aufgabe 10.8. The Taylor polynomial of degree n of a function $f \in C^n(\mathbb{R})$ at $x_0 \in \mathbb{R}$ is given by

$$T^{(n)} f(x) = \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k.$$

Write a constructor for the class `Polynomial` from Exercise 10.5 which creates for $n \geq 0$ the Taylor polynomial of degree n of the functions `sin`, `cos` oder `exp` (use a string input to choose between them) at $x_0 = 0$. Test your implementation appropriately!