



Seminarblatt 2

1. Programmparameter

Beim Start eines Programms können Argumente gegeben werden, z.B. `myProg --import datenbank`. In C++ kann die `main`-Funktion diese (optional) als Parameter übernehmen. Es reicht die Signatur folgenderweise anzupassen:

```
int main(int argc, char *argv[])
```

`argc` ist die Anzahl der Argumente, und `argv` ist ein Pointer auf char Arrays (C-Strings).

Beispiel:

```
myProg --import datenbank
```

```
argc <- 3, argv[0] <- "myProg", argv[1] <- "--import", argv[2] <- "datenbank"
```

2. Dateien

Für die Dateioperationen werden Objekte der Klasse `fstream` verwendet. Wird in die Datei nur geschrieben, kann stattdessen die Klasse `ofstream` verwendet werden. Für reine Eingabedateien bietet sich die Klasse `ifstream` an. Auf die Objekte dieser Klassen können Ein- und Ausgabeoperatoren (`>>` und `<<`) angewandt werden. Vor einem Zugriff muss die Datei mit der Methode `open()` geöffnet werden, und nach dem Lesen oder Schreiben muss die Datei mit `close()` geschlossen werden.

`printTextToFile.cpp`

```
#include <fstream>
using namespace std;

int main()
{
    fstream f;
    f.open("test.dat", ios::out);
    f << "Dieser Text geht in die Datei"
    << endl;
    f.close();
}
```

`printFileToStandardOutput.cpp`

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

int main(int argc, char *argv[])
{
    fstream f;
    char cstring[256];
    f.open(argv[1], ios::in);
    while (!f.eof())
    {
        f.getline(cstring, sizeof(cstring));
        cout << cstring << endl;
    }
    f.close();
}
```

3. Random

`int rand()` = gibt eine zufällige Ganzzahl zwischen 0 und `RAND_MAX` zurück

`RAND_MAX` = eine Konstante definiert in `<cstdlib>`

```
int v1 = rand() % 100;           // v1 in the range 0 to 99
int v3 = rand() % 30 + 1985;     // v3 in the range 1985-2014
double v4 = static_cast<double>(rand());
void srand (unsigned int seed);  // stellt den Zustand (seed) des Pseudozufallszahlengenerators
```

4. Casting

In C++ gibt es für die explizite Typwandlung eine Funktion `static_cast`, dem der Namen eines Datentyps in spitzen Klammern und dann in runden Klammern der Ausdruck folgt, dessen Wert in den angegebenen Datentyp zu wandeln ist. So ist der Ausdruck `static_cast<double>(123)` ein Ausdruck vom Datentyp `double`, dessen Wert sich aus der Wandlung des Wertes des Numerales 123 (vom Datentyp `int`) in den Datentyp `double` ergibt, das sollte also der Wert 123.0 sein.

```
#include <iostream>
int main() {
    std::cout << static_cast<int>(2.7) << std::endl;
    std::cout << static_cast<int>('A');          // ASCII CODE
}
```

5. Klassen

Objektorientierte Programmierung erfordert eine neue Denkweise:

1. Welche Objekte/Klassen sind schon im Problemraum vorhanden?
2. Welche Dienste muss jede Klasse erbringen?

Vergleichen Sie dies mit der prozeduralen Programmierung: anstatt das Problem in Daten + Algorithmen zu zerlegen, zerlegen wir es zuerst in Objekte (Daten + Verhalten).

Objekte liefern Abstraktionen. Ein Objekt kann ohne Wissen über die konkrete Implementierung verwendet werden. Dadurch kann Software aus Komponenten erstellt werden - genau wie andere Engineering-Systemen.

Date.h (Headerdatei)

```
class Date {
public: // access functions
    int get_day();
    int get_month();
    int get_year();

    // function to set the date
    void set_date(int d, int m, int y);

private:
    int day;
    int month;
    int year;
};
```

Date.cpp (Implementierung)

```
int Date::get_day(){ return day; }
int Date::get_month(){ return month; }
void Date::set_date(int d, int m, int y){
    // check month is 1..12
    if(m < 1 || m > 12) raise_error();

    // check day is 1..31
    if(d < 1 || d > 31) raise_error();

    // if April, May, September, November
    if(m==4 || m==6 || m==9 || m==11)
        if(d > 30)
            Raise_Error();

    // if February use leap year rules
    if(m==2){
        if((y%4==0)^(y%100==0)^(y%400==0)){
            if(d > 29) raise_error();
        }
        else if(d > 28) raise_error();
    }

    // if we got here d,m,y are OK
    day = d; month = m; year = y;
```



}

Übungen

1. Schreibe ein Programm das zwei Parameter von der Kommandozeile übernimmt. Diese Parameter sind der Name einer Eingabedatei, und einer Ausgabedatei.

Aus der Eingabedatei sollten die ersten 5 Zeilen auf der Konsole ausgegeben werden und in die Ausgabedatei geschrieben werden.

Achtung, auf die Überprüfung von nicht-existierender Eingabedatei, und inkorrektem Ausgabedateinamen.

2. Implementieren Sie eine Klasse für Bankkonten. Die Klasse soll den aktuellen Saldo als Fließkommazahl beinhalten. Weiters soll sie Methoden für folgende bereitstellen: Ein- und Auszahlungen vorzunehmen.

auszahlen(betrag)

einzahlen(betrag)

kontostand()

Bedenke wie die Signaturen der Methoden zu deklarieren sind.

Stellen Sie sicher, dass der Konstruktor den Saldo mit Null initialisiert.

3. Datei schreiben. Erweitern Sie die Account-Klasse um eine Methode, die den Typ und Betrag jeder Transaktion in eine Datei speichert. Format: IN/OUT Betrag Saldo

Beispiel:

IN 110 110

OUT 5 105

IN 1 106

4. Datei lesen. Erweitern Sie die Account-Klasse um eine Methode, die den Typ und Betrag jeder Transaktion aus einer Datei einliest. Format: IN/OUT Betrag

Zusatz

5. Man kann die Kreiszahl Pi näherungsweise bestimmen:

(a) Stellen Sie sich einen Kreis vor, dessen Fläche ist: $\text{Fläche} = \pi * R^2$

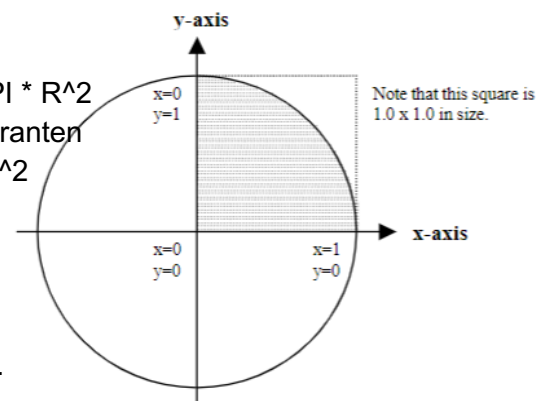
(b) Stellen Sie sich nun vor, Sie teilen den Kreis genau in 4 Quadranten

(c) Die Fläche eines Quadranten ist daher: $\text{Fläche} = 0,25 * \pi * R^2$

(d) Setzen wir nun den Radius auf $R = 1$;

(e) Die Gleichung lautet daher: $\text{Fläche} = 0,25 * \pi$

(f) Wir können also einfach sagen: $\pi = \text{Fläche} * 4$



Die Frage ist nun, wie man die grau schattierte Fläche berechnet.

Idee (stochastisch)

Man stelle sich einen Kreis mit Mittelpunkt (0,0) und Radius 1 vor. Es werden zufällig Punkte erzeugt, bei denen sowohl x als auch y im Intervall [0,1] liegen. Dann wird die Entfernung dieser Punkte zum Ursprung ermittelt. Ist diese Entfernung kleiner als 1, so liegt der Punkt innerhalb des Kreises. Setzt man bei einer



ausreichenden Zahl von Zufallspunkten die Zahl der Treffer in das richtigen Verhältnis zur Gesamtzahl der Punkte, so erhält man einen Näherungswert für π ($\pi = 4 * \text{AnzahlTreffer} / \text{AnzahlGesamt}$).

Schreiben Sie eine Funktion, das auf oben beschriebene Weise π berechnet.