Name, Matrikelnummer:

technische universität dortmund

fakultät für informatik

Andre Droschinsky
Denis Kurz
Roman Kalkreuth

Dortmund, den 01. November 2018

Praktikum zur Vorlesung Einführung in die Programmierung WS 18/19

Blatt 2

Es können 16 Punkte erreicht werden.

Allgemeine Hinweise

1. Bitte lesen Sie vor der Bearbeitung **alle** Aufgaben sorgfältig durch! Dies erspart Ihnen unnötige Arbeit und somit auch Zeit!
2. Die einzigen Header, die Sie zur Bearbeitung der Aufgaben verwenden dürfen, sind `iostream` und solche, die laut Aufgabenstellung explizit erlaubt werden.
3. Lassen Sie sich fertiggestellte Aufgaben bitte möglichst **frühzeitig** testieren. In der letzten halben Stunde vor Schluss wird nur noch **eine** Aufgabe testiert!
4. Wir akzeptieren ein Testat nur, wenn die Lösung eigenständig auf Anhieb erklärt werden kann. Andernfalls müssen wir die entsprechende Teilaufgabe mit 0 Punkten bewerten.

Grundlage: Ein- und Ausgabe auf der Konsole

Auf dem letzten Praktikumsblatt wurden bereits Zeichenketten und der Inhalt von Variablen auf der Konsole ausgegeben. Dies geschieht über den Ausgabestrom `std::cout`. Dabei können auch mehrere Zeichenketten und Variablen in einer Zeile ausgegeben werden. Ein Zeilenumbruch kann durch `std::endl` eingefügt werden. Beispiel:

```
1  int var1 = 1;
2  int var2 = 2;
3  cout << "Die Werte sind " << var1 << " und " << var2 << endl;
4  cout << "Dieser Text erscheint in der nächsten Zeile ";
5  cout << "und dieser Text erscheint in der gleichen Zeile" << endl;
```

Hinweis: Über die Direktive `using namespace std` wird der Namensbereich `std` verwendet. Da dieser auch den Ausgabestream `cout` beinhaltet muss nach der `using`-Direktive die Kennzeichnung `std::` nicht mehr vor den Befehl gestellt werden. Das Gleiche gilt für den Zeilenumbruch `endl` oder den Eingabestrom `cin`.

Tastatureingaben können durch den Eingabestrom `std::cin` erfolgen. Im Gegensatz zu dem Ausgabeoperator `<<` muss der Eingabeoperator `>>` verwendet werden. Die Eingabe wird immer zeilenweise durchgeführt; d.h. sie wird mit der Taste `Enter` beendet. Beispiel:

```
1  int x;
2  cin >> x; // liest Zeichen ein bis Enter gedrückt wird und weist diese x zu
3  cout << "Die Zahl " << x << " wurde eingegeben.";
```

Hinweis 2: Achten Sie darauf, dass Sie nur zulässige Eingaben machen. Wenn die einzulesende Variable z.B. vom Datentyp `int` ist, dann dürfen nur Ziffern eingegeben werden. Bei falschen Eingaben wird eine 0 ausgegeben und der Eingabestrom ist für alle folgenden Eingaben (zur Laufzeit) ungültig. Wie mit falschen Eingaben umgegangen werden kann, wird im späteren Verlauf der Vorlesung behandelt.

Hinweis 3: Achten Sie darauf, dass in Eclipse die *Console-View* aktiv ist, während Sie die Eingabe machen.

Aufgabe 1: Fehlersuche (3 Punkte)

Ein Computer-Virus hat Ihren Rechner befallen und die Inhalte einiger Dateien verändert. Zufällig wurden Zeilen vertauscht und einzelne Zeichen gelöscht & verändert. Hier sehen Sie einen Ausdruck der Datei `widerstand.cpp`, die durch den Virus in Mitleidenschaft gezogen wurde.

```
using namespace std;
int main(){
\\ Dieses Programm berechnet den elektrischen Widerstand
double strom = 5.0
cout << "Berechneter Widerstand: ";
double widerstand = spannung / strom;
cout << widerstand >> endl;
double spannung = 230,0;
return 0; // fertig
}
#include <iostream>
```

Ihre Aufgabe besteht darin, die durch den Virus hervorgerufenen Änderungen rückgängig zu machen, so dass das Programm seinen ursprünglichen Zweck erfüllt. Legen Sie dazu ein neues Projekt mit dem Namen `Aufgabe_2_1` an. Fügen Sie die ihnen zur Verfügung gestellte vom Virus degenerierte Datei[1] `widerstand.cpp` hinzu. Stellen Sie anschließend dessen Originalzustand wieder her, um ein lauffähiges Programm zu erhalten.

Hinweis 4: Orientieren Sie sich dabei am Programmtext von Aufgabe 3 vom letzten Mal.

_____ (3)

Aufgabe 2: Geradengleichung (7 Punkte)

Aus der elementaren Algebra sind Geradengleichungen mit dem Muster $f(x) = mx + b$ bekannt. Es braucht lediglich zwei Punkte, um eine Gerade eindeutig zu beschreiben.

a) Legen Sie ein neues Projekt mit dem Namen `Aufgabe_2_2a` an und fügen Sie das Codegerüst aus der zur Verfügung gestellten Datei namens `gerade.cpp` hinzu.
Ergänzen Sie den Code, indem Sie die Berechnung der Geradenparameter m, b unter Zuhilfenahme der Eingabevariablen `x1, y1, x2, y2`. Benutzen Sie dabei den Operator zur Division `/`. Die Werte für die Eingabevariablen `x1, y1, x2, y2` sollen über die Tastatur eingelesen und danach entsprechend zugewiesen werden.

_____ (3)

b) Tragen Sie die Ergebnisse Ihrer Berechnung in folgender Tabelle ein:

Punkte (x, y)	m	b
$(7, 8), (-3, -2)$		
$(4, 7), (1, -2)$		
$(2, 5), (-5, 5)$		
$(-3, 9), (6, -9)$		

_____ (1)

c) Erweitern Sie den Code so, dass zusätzlich überprüft werden soll, ob ein dritter Punkt auf der bereits berechneten Gerade liegt. Überprüfen Sie den Code, indem Sie zu allen berechneten Geraden aus `Aufgabe_2_2b` bestimmen, ob der Punkt $(3, 4)$ auf der Gerade liegt. Geben Sie dazu die Meldung „Der Punkt (3, 4) liegt [nicht] auf der Geraden“ aus, je nachdem ob der Punkt auf der Geraden liegt oder nicht.

_____ (3)

[1] https://ls11-www.cs.tu-dortmund.de/teaching/ep1819sopra

Aufgabe 3: Ausgabe von Pokèmon (6 Punkte)

Die Art und Namen von sechs Pokèmon sind folgendermaßen definiert:

```
Elektro = {Pikachu,Magneton,Voltobal}
Feuer = {Glumanda,Vulpix,Arkani}
```

Legen Sie ein neues Projekt mit dem Namen `Aufgabe_3` an und fügen sie die bereitgestellte Datei `pokemon.cpp` hinzu.

a) Ergänzen Sie den Code so, dass alle möglichen *Namen* der Pokèmon in der unten stehenden Reihenfolge ausgegeben werden. Verwenden Sie dazu eine **for**-Schleife mit einer Laufvariablen vom Typ `int`. Die Elektropokèmon haben den Indexbereich 0 bis einschließlich 2, Feuerpokèmon 3 bis einschließlich 5. Im Schleifenrumpf wird eine **switch-case** Anweisung benötigt, welche den zum `int`-Wert passenden Namen des Pokèmon ausgibt.

_____ (2)

```
Pikachu
Magneton
Voltobal
Glumanda
Vulpix
Arkani
```

b) Ergänzen Sie den Code aus der vorherigen Teilaufgabe so, dass alle Arten der Pokèmons mit dem zugehörigen Pokèmon ausgegeben werden. Die Ausgabe hat dann folgende Gestalt:

```
Art Elektropokemon:
Pikachu
Magneton
Voltobal

Art Feuerpokemon:
Glumanda
Vulpix
Arkani
```

Sie können dazu **switch-case** sowie **if** Anweisungen und **for** Schleifen benutzen.

_____ (2)

c) Legen sie eine neue Datei `pokemon_enum.cpp` an. Fügen Sie zu Beginn der `main`-Methode die Zeilen

```
enum feuer {Pikachu,Magneton,Voltobal};
enum elektro {Glumanda,Vulpix,Arkani};
```

ein und erzeugen Sie die Ausgabe so wie in Teilaufgabe b unter Verwendung von Aufzählungstypen. Sie können dazu **switch-case** sowie **if** Anweisungen und **for**-Schleifen benutzen.

_____ (2)