Aufgabe 1 Programmiersprachen

Philipp Tornow
Matrikelnummer: 118332
Uebungsgruppe: 1

Abgabe 23.04.18

Aufgabe 1.3

siehe: https://github.com/PhilippTeepunkt/programmiersprache-aufgabe-1.git

Aufgabe 1.4

Typen, Variablen, Werte

Typen

Im Beispiel: int, bool, char → ganzzahlige bzw. integerale Typen double → Gleitkomma Typ

Variablen

Im Beispiel: a,b,c,d

Werte

Im Beispiel: 9, false, 'a', 1.3

const - declaration

Mit dem const Schlüsselwort, wird dem Compiler mitgeteilt, eine Änderung des Wertes oder Pointers einer Variable zu verhindern. Variablen, die mit dem const-Schlüsselwort definiert werden, sind also konstant.
Konstante Variablen und Pointer können auch als Parameter in Methoden verwendet werden, um Änderungen durch die Methode zu verhindern.
Im Beispiel: int **const** two =2;
→ der Wert 2 kann über die Variable two nicht verändert werden.

Typumwandlung/Konvertierung

In C++ gibt es die automatische Typumwandlung und die Typkonvertierung über Casting. Allgemein, versteht man unter einer Typkonvertierung eine Änderung des Typs einer Variable. Zu Problemen kann es kommen, wenn bei der Konvertierung Informationen verloren gehen. Grundsätzlich muss der kleinere Datentyp in einen größeren konvertiert werden. In C++ nimmt der Compiler automatische Anpassungen vor. So kann zum Beispiel vom Typ char(sign./unsign.) und short (sign./unsign.) in den Typen int konvertiert werden oder von int in bool,
wobei 1 = true bzw. 0 = false.

Aufgabe 1.5

Initialisierung

Eine Initialisierung beinhaltet eine Definition, welche wiederum eine Deklaration impliziert. Eine Initialisierung entspricht also einer Definition mit Anfangswertzuweisung.
Beispiel: int x = 1; oder int x{1};

Zuweisung

Bei einer Zuweisung wird einer bereits definierten Variable ein neuer Wert zugewiesen. Wurde die Variable bereits initialisiert, wird der Wert überschrieben. Im Gegensatz zur Initialisierung hat die Zuweisung nur eine Form: b = h;
Am besten wird einem der Unterschied bewusst, wenn man den folgenden Ausschnitt betrachtet:

```
class MeineKlasse {
        private:
                int const _a;
                int const _b;
        public :
                MeineKlasse{int a, int b}:_a{a} { _b = b; }
};
```

Während _a den Wert zugewiesen bekommt, durch eine Initialisierung in der Initialisierungsliste des Konstruktors, gäbe es eine Fehlermeldung bei _b. Im Gegensatz zu _a wird im Körper des Konstruktors versucht _b einen Wert mithilfe des Zuweisungsoperators zu zuweisen. Da vor dem Konstruktor _a und _b bereits initialisiert werden müssen und beide konstant sind, gibt es einen Fehler.

Aufgabe 1.6

Deklaration

Deklarationen benennen Variablen, Namespaces, Funktionen und Klassen, die im Code verwendet werden sollen. Außerdem wird bei einer Deklaration der Typ der Komponente angegeben. Details werden hierbei nicht genannt. Außer für Funktionen, Objekte und Variablen mit ëxtern"Bezeichner oder vom statischen Typ sind Deklarationen auch Definitionen. Objekte werden hierbei über einen Default-Konstruktor konstruiert und Variablen erhalten einen Zufallswert. Bei einer Deklaration einer Funktion, wird nur ihre Existenz bekannt gegeben und ihre Signatur.
Beispiel:

Klasse: class X;
Konstruktor: public: X(X const&);
Methode: float create() const;
Variable: int a_;

Definition

Definitionen legen den Aufbau einer Klasse/Datentyps fest, gibt die Implementierung einer Funktion oder den Speicherortu und Wert einer Variable an. Beispiel:

Klasse: class x { ... }
Konstruktor: public x(...)...
Methode: float create(){ return 2.1;}
Variable: int x{5};

Unterschied: Bei einer Deklaration wird dem Compiler mitgeteilt, dass es ein Objekt, eine Funktion, eine Variable ... gibt, aber nicht wie die Komponente aussieht bzw. was Sie tut. Eine Definition nennt diese Details.

Aufgabe 1.7

Die Signatur einer Funktion ist durch die Eingangsparameter der Funktion gegeben. Eine Funktion kann verschieden implementiert werden, dabei aber jeweils den gleichen Namen tragen (Overloading). Damit der Compiler die Funktionen jedoch unterscheiden kann, muss die Methode mit verschiedenen Signaturen angegeben werden.
Beispiel: void create();
void create(int const& zahl);

Gültigkeitsbereich: int var = 3; -> class scope
(int var) -> method scope

Aufgabe 1.17