

# C++: Klassen

---

Carsten Gips (HSBI)

Unless otherwise noted, this work is licensed under CC BY-SA 4.0.

# OOP in C++

```
public abstract class Dummy {  
    public Dummy(int v) { value = v; }  
    public abstract int myMethod();  
  
    private int value;  
}
```

```
class Dummy {  
public:  
    Dummy(int v = 0);  
    int myMethod();  
    virtual ~Dummy();  
private:  
    int value;  
};
```

# Objekte erzeugen mit Konstruktoren

```
class Dummy {  
public:  
    Dummy(int c=0) { credits = c; }  
private:  
    int credits;  
};
```

```
Dummy a;  
Dummy b(37);  
Dummy c=99;
```

=> Kein Aufruf von `new`!

## C++: Trennung .h und .cpp

```
// .h  
class Dummy {  
public:  
    Dummy(int c=0);  
private:  
    int credits;  
};
```

```
// .cpp  
Dummy::Dummy(int c) {  
    credits = c;  
}
```

# Konstrukturen: Normale (Java-like) Initialisierung

```
class Student {  
public:  
    Student(const string &n, const Date &d, double c) {  
        name = n;  
        birthday = d;  
        credits = c;  
    }  
private:  
    string name;  
    Date birthday;  
    double credits;  
};
```

# Konstrukturen: Initialisierungslisten

```
class Student {  
public:  
    Student(const string &n, const Date &d, double c)  
        : name(n), birthday(d), credits(c)  
    {}  
private:  
    string name;  
    Date birthday;  
    double credits;  
};
```

# C++ und explizite Konstruktoren

- Implizite Konvertierung mit einelementigen Konstruktoren:

```
class Dummy {  
public:  
    Dummy(int c=0);  
};  
  
Dummy a;  
a = 37;      // Zuweisung(!)
```

- Falls unerwünscht: Schlüsselwort `explicit` nutzen

```
explicit Dummy(int c=0);
```

- Klassendefinition mit Semikolon abschließen (!)
- Sichtbarkeiten blockweise, keine für Klasse
- Daten liegen direkt im Objekt (anderenfalls Pointer nutzen)
- Attribute sind echte Objekte: Initialisieren mit `NULL` nicht möglich
- Konstruktoren: Kein `new` nötig (würde Objekt auf Heap anlegen und Pointer liefern)





Unless otherwise noted, this work is licensed under CC BY-SA 4.0.