Westfälische
Hochschule
Gelsenkirchen Bocholt Recklinghausen

Grundlagen der Informatik und Programmierung 2

Vererbung

Methoden überschreiben

Prof. Dr. Tom Vierjahn

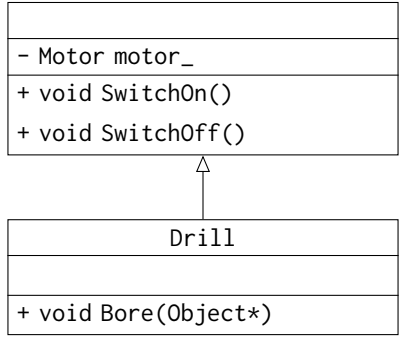
Visual Computing (https://vc.w-hs.de)
Fachbereich Wirtschaft und Informationstechnik
Campus Bocholt

Sommersemester 2020

Neue Methoden in abgeleiteter Klasse

Westfälische Hochschule

- Abgeleitete Klassen können zusätzliche, spezialisierte Methoden hinzufügen.



Neue Methoden in abgeleiteter Klasse

Westfälische Hochschule

- Abgeleitete Klassen können zusätzliche, spezialisierte Methoden hinzufügen.

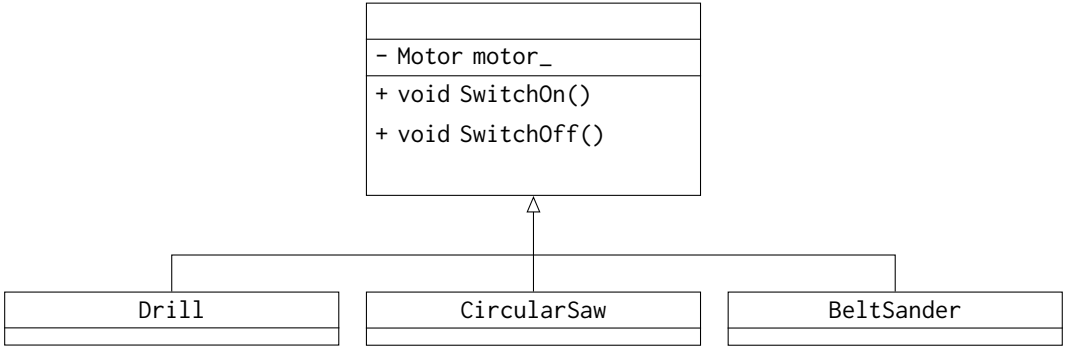
```
class PowerTool {
 public:
  void SwitchOn() { ... }
  void SwitchOff() { ... }

 private:
  Motor motor_;
};

class Drill : public PowerTool {
 public:
  void Bore(Object*) { ... }
};
```

```
Object o;

Drill d;
d.SwitchOn();
d.Bore(&o);
d.SwitchOff();

PowerTool p;
p.SwitchOn();
p.Bore(&o);
p.SwitchOff();
```

Methoden überschreiben

Westfälische Hochschule

- Abgeleite Klassen können Methoden mit spezialisierter Implementierung überschreiben.



Methoden überschreiben

Westfälische Hochschule

- Abgeleite Klassen können Methoden mit spezialisierter Implementierung überschreiben.

Klasse Person:

```
class Person {
 public:
  // Person constructor here

 std::string ToString() const;

protected:
 // now protected -- ex-private -- members here
};
```

Methoden überschreiben

Westfälische Hochschule

- Abgeleite Klassen können Methoden mit spezialisierter Implementierung überschreiben.

Klasse Student:

```
class Student : public Person {
 public:
  // Student constructor here

 std::string ToString() const;

protected:
 // now protected -- ex-private -- members here
};
```

Methoden überschreiben

Westfälische Hochschule

- Abgeleite Klassen können Methoden mit spezialisierter Implementierung überschreiben.

Klasse StudentInProgram:

```
class StudentInProgram : public Student {
 public:
  // StudentInProgram constructor here

 std::string ToString() const;

protected:
 // now protected -- ex-private -- members here
};
```

Methoden überschreiben

Westfälische Hochschule

Implementierung in Person:

```
std::string Person::ToString() const {
  std::stringstream out;
  out << last_name_ << ", " << first_name_;
  return out.str();
}
```

Anwendung:

```
Person john{"John", "Doe"};
std::cout << john.ToString() << std::endl;
```

Ausgabe:

Methoden überschreiben

Westfälische Hochschule

Implementierung in Student:

```
std::string Student::ToString() const {
  std::stringstream out;
  out << last_name_ << ", " << first_name_ << " (" << id_ << ')';
  return out.str();
}
```

Anwendung:

```
Student jrandom{"J. Random", "Student", 199100003};
std::cout << jrandom.ToString() << std::endl;
```

Ausgabe:

Methoden überschreiben

Westfälische Hochschule

Implementierung in StudentInProgram:

```
std::string StudentInProgram::ToString() const {
  std::stringstream out;
  out << last_name_ << ", " << first_name_ << " (" << id_ << ')';
  out << ": " << program_ << " (" << regulations_ << ')';
  return out.str();
}
```

Anwendung:

```
StudentInProgram jane{"Jane", "Appleseed", 199100001, "I.S", "2018"};
std::cout << jane.ToString() << std::endl;
```

Ausgabe:

Basisklassen-Methode aufrufen

Westfälische Hochschule

Implementierung in Person:

```
std::string Person::ToString() const {
  std::stringstream out;
  out << last_name_ << ", " << first_name_;
  return out.str();
}
```

Anwendung:

```
Person john{"John", "Doe"};
std::cout << john.ToString() << std::endl;
```

Ausgabe:

Basisklassen-Methode aufrufen

Westfälische Hochschule

Implementierung in Student:

```
std::string Student::ToString() const {
  std::stringstream out;
  out << Person::ToString() << " (" << id_ << ')';
  return out.str();
}
```

Anwendung:

```
Student jrandom{"J. Random", "Student", 199100003};
std::cout << jrandom.ToString() << std::endl;
```

Ausgabe:

Basisklassen-Methode aufrufen

Westfälische Hochschule

Implementierung in StudentInProgram:

```
std::string StudentInProgram::ToString() const {
  std::stringstream out;
  out << Student::ToString() << ": " << program_ << " (" << regulations_
      << ')';
  return out.str();
}
```

Anwendung:

```
StudentInProgram jane{"Jane", "Appleseed", 199100001, "I.S", "2018"};
std::cout << jane.ToString() << std::endl;
```

Ausgabe:

Überschreiben verdeckt Namen

Westfälische Hochschule

Klassen:

```
class A {
 public:
  int Sum(int a, int b) { return a + b; }
  int Sum(int a, int b, int c) { return a + b + c; }
};

class B : public A {
 public:
  float Sum(float a, float b) { return a + b; }
};
```

Anwendung:

```
A a;
std::cout << a.Sum(1, 2) << std::endl;
std::cout << a.Sum(1, 2, 3) << std::endl;
```

Überschreiben verdeckt Namen

Westfälische Hochschule

Klassen:

```
class A {
 public:
  int Sum(int a, int b) { return a + b; }
  int Sum(int a, int b, int c) { return a + b + c; }
};

class B : public A {
 public:
  float Sum(float a, float b) { return a + b; }
};
```

Anwendung:

```
B b;
std::cout << b.Sum(1.0, 2.0) << std::endl;
std::cout << b.Sum(1, 2, 3) << std::endl;
```

Überschreiben verdeckt Namen

Westfälische Hochschule

Klassen:

```
class A {
 public:
  int Sum(int a, int b) { return a + b; }
  int Sum(int a, int b, int c) { return a + b + c; }
};

class B : public A {
 public:
  float Sum(float a, float b) { return a + b; }
};
```

Abhilfe:

```
B b;
std::cout << b.Sum(1.0, 2.0) << std::endl;
std::cout << b.A::Sum(1, 2, 3) << std::endl;
```

using-Schlüsselwort

Westfälische Hochschule

Klassen:

```
class A {
 public:
  int Sum(int a, int b) { return a + b; }
  int Sum(int a, int b, int c) { return a + b + c; }
};

class B : public A {
 public:
  float Sum(float a, float b) { return a + b; }
  using A::Sum;
};
```

Anwendung:

```
B b;
std::cout << b.Sum(1.0, 2.0) << std::endl;
```

using-Schlüsselwort

Westfälische Hochschule

Klassen:

```
class A {
 public:
  int Sum(int a, int b) { return a + b; }
  int Sum(int a, int b, int c) { return a + b + c; }
};

class B : public A {
 public:
  float Sum(float a, float b) { return a + b; }
  using A::Sum;
};
```

Abhilfe:

```
B b;
std::cout << b.Sum(1.0f, 2.0f) << std::endl;
```

Zusammenfassung

Westfälische Hochschule

- neue Methoden in abgeleiteter Klasse
- Methoden überschreiben
- Basisklassen-Methoden aufrufen
- Verdecken von Namen

Prof. Dr. Tom Vierjahn

- E-Mail: tom.vierjahn@w-hs.de

Visual Computing

- Web: `https://vc.w-hs.de`
- YouTube: Visual Computing WH
- Twitter: @VisComputingWH

Westfälische Hochschule
Fachbereich Wirtschaft und Informationstechnik
Campus Bocholt