Inheritance

Manuel Ranzmeir

Allgemein

- Klassen können folgende Attribute vererben:
  - Methoden
  - Properties
  - Eigenschaften von anderen Klassen
- Die erbenden Klassen können auf Attribute von den Vererbenden Zugreifen (Methoden aufrufen/überschreiben)

Basisklasse

- Eine Klasse welche von keiner anderen Klasse erbt

```
class Vehicle {
    var currentSpeed = 0.0
    var description: String {
        return "traveling at \(currentSpeed) miles per hour"
    }
    func makeNoise() {
        // do nothing - an arbitrary vehicle doesn't necessarily make a noise
    }
}
```

```
let someVehicle = Vehicle()
```

```
print("Vehicle: \(someVehicle.description)")
// Vehicle: traveling at 0.0 miles per hour
```

Vererben

- Diese Klasse erbt alle Charakteristiken der Basisklasse bzw. überschreibt diese.

```
class SomeSubclass: SomeSuperclass {
    // subclass definition goes here
}
```

```
class Bicycle: Vehicle {
    var hasBasket = false
}
```

```
let bicycle = Bicycle()
bicycle.hasBasket = true
```

```
bicycle.currentSpeed = 15.0
print("Bicycle: \(bicycle.description)")
// Bicycle: traveling at 15.0 miles per hour
```

```
class Tandem: Bicycle {
    var currentNumberOfPassengers = 0
}
```

```
let tandem = Tandem()
tandem.hasBasket = true
tandem.currentNumberOfPassengers = 2
tandem.currentSpeed = 22.0
print("Tandem: \(tandem.description)")
// Tandem: traveling at 22.0 miles per hour
```

Auf Attribute der Basisklasse zugreifen

- Methode: super.someMethod()
- Property: super.someProperty
- Subskript: super[someIndex]

Überschreiben von Methoden

- Präfix „override“ verwenden

```
class Train: Vehicle {
    override func makeNoise() {
        print("Choo Choo")
    }
}
```

```
let train = Train()
train.makeNoise()
// Prints "Choo Choo"
```

Überschreiben von Properties

- Die Getter bzw. Setter der Basisklasse werden überschrieben.

```
class Car: Vehicle {
    var gear = 1
    override var description: String {
        return super.description + " in gear \(gear)"
    }
}
```

```
let car = Car()
car.currentSpeed = 25.0
car.gear = 3
print("Car: \(car.description)")
// Car: traveling at 25.0 miles per hour in gear 3
```

Überschreiben von Property-Observner

- Der Gang des Autos wird automatisch gewählt.

```
class AutomaticCar: Car {
    override var currentSpeed: Double {
        didSet {
            gear = Int(currentSpeed / 10.0) + 1
        }
    }
}
```

```
let automatic = AutomaticCar()
automatic.currentSpeed = 35.0
print("AutomaticCar: \(automatic.description)")
// AutomaticCar: traveling at 35.0 miles per hour in gear 4
```

Überschreiben/Vererbung verhindern

- Präfix „final“ verwenden:
  - final var
  - final func
  - final class func
  - final subscript
- Klasse kann nicht mehr vererben: final class

Beispiel

- Erstelle die Klasse „Adresse“ und speichere darin: PLZ, Straße, Ort und Hausnummer Diese Klasse darf nicht vererben und muss eine passende init Methode haben.

- Erstelle die Klasse „Person“, welche einen Namen, ein Geburtsdatum und eine Adresse enthält! Diese soll eine init Methode und eine, welche alle Eigenschaften in Form eines Strings zurückliefert, besitzen .

- Erstelle die Klasse „Schüler“, welche von „Person“ erbt, und gib die Attribute aus!

- Erstelle die Klasse „Lehrer“, welche von „Person“ erbt! Diese soll zusätzlich ein Kürzel (z.B. „HELF“ oder „HELM“) beinhalten. Überschreibe die Methoden der Basisklasse und gib die Attribute aus!