

# 11b) Notation von UML-Klassendiagrammen (UML-CD) mit PlantUML

Prof. Dr. Uwe Aßmann  
Institut für Software- und  
Multimediatechnik  
Lehrstuhl Softwaretechnologie  
Fakultät für Informatik  
TU Dresden  
Version 21-0.1, 19.04.21

- ▶ PlantUML <https://plantuml.com/de/> ist ein tolles Werkzeug für das Zeichnen und automatisierte Layout von UML-Diagrammen
  - Klassendiagramme <https://plantuml.com/de/class-diagram>
  - Sequenzdiagramme <https://plantuml.com/de/sequence-diagram>
  - Komponentendiagramme (eingeschränkt)  
<https://plantuml.com/de/component-diagram>
  - Nutzfalldiagramme <https://plantuml.com/de/use-case-diagram>
  - Zustandsdiagramme <https://plantuml.com/de/state-diagram>
  - Aktivitätsdiagramme <https://plantuml.com/de/activity-diagram-beta>
  - Objektdiagramme <https://plantuml.com/de/object-diagram>
- ▶ Man erstellt eine Datei `datei.plantuml` in textueller Syntax mit “Tripeln” (entspricht einfachen Sätzen wie “X ist-ein Y”)
- ▶ `plantuml datei.plantuml --> datei.png`

# Nachinstallation mit einem Paketmanager

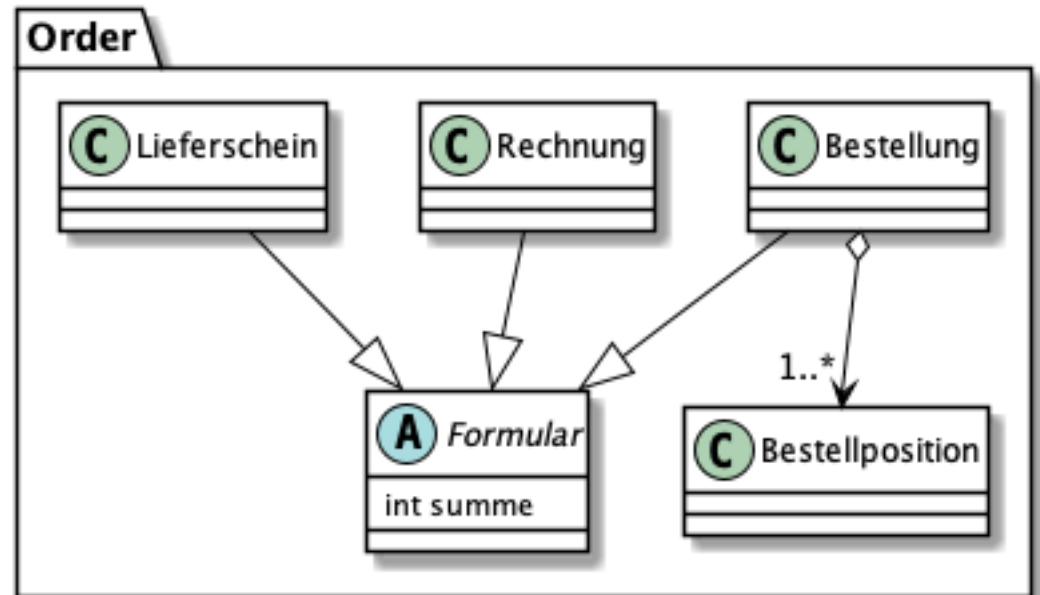
- ▶ Q: Wie bekommt man `plantuml` auf seinen Rechner?
- ▶ A: mit `brew` oder `apt-get`, den Paketmanagern von Linux/Mac:
- ▶ `brew install plantuml`
- ▶ `sudo apt-get install plantuml`
  
- ▶ Lustig:
- ▶ `plantuml -tutxt diagramm.plantuml`
- ▶ `--` erzeugt `utf8text`-Diagramm
- ▶ `plantuml -tpdf diagramm.plantuml`
- ▶ `--` erzeugt `pdf`-Diagramm

# Unsere Formulare

plantuml/simple-form.plantuml

4 Softwaretechnologie (ST)

```
@startuml
' @author Uwe Assmann
' @version 0.1
' @date 2020-05-30
skinparam class {
  BackgroundColor White
  BorderColor Black
  ArrowColor Black
}
namespace Order {
  abstract class Formular
  Formular : int summe
  Bestellung o--> "1..*" Bestellposition
  Bestellung --|> Formular
  Rechnung --|> Formular
  Lieferschein --|> Formular
}
@enduml
```

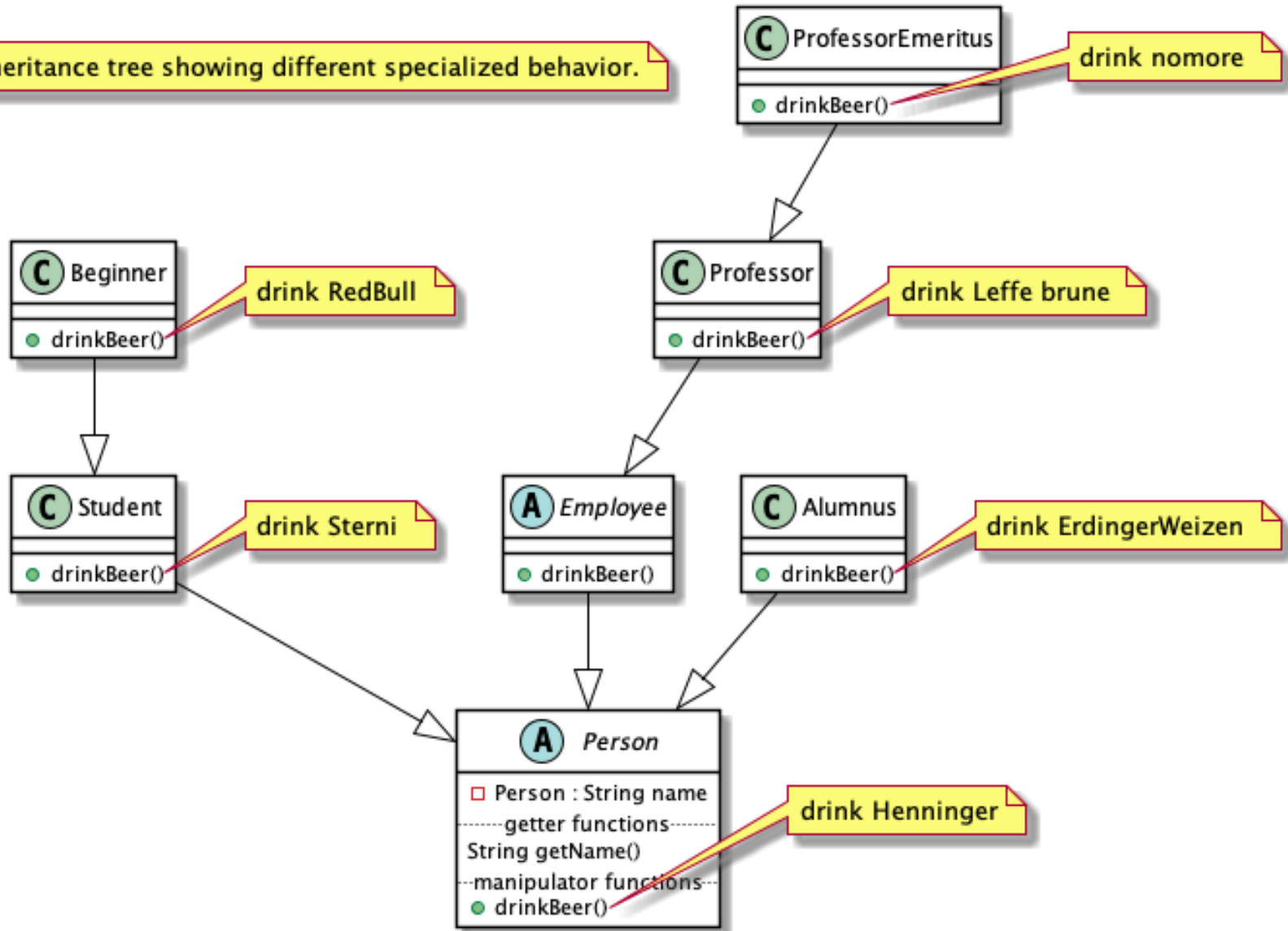


# Personen und Vererbung

plantuml/persons.plantuml

Order

This is a simple class inheritance tree showing different specialized behavior.



```

@startuml
  @author Uwe Assmann
  @version 0.1
  @date 2021-04-12

  skinparam class {
    BackgroundColor White
    BorderColor Black
    ArrowColor Black
  }

  namespace Order {
    note "This is a simple class inheritance
  tree showing different specialized behavior." as N1
    abstract class Person {
      -Person : String name
      .. getter functions ..
      String getName()
      .. manipulator functions ..
      +drinkBeer()
    }
    note right of Person::drinkBeer()
      drink Henninger
    end note
    Alumnus --|> Person
    class Alumnus {
      +drinkBeer()
    }
    note right of Alumnus::drinkBeer()
      drink ErdingerWeizen
    end note
  }

```

```

Employee --|> Person
abstract class Employee {
  +drinkBeer()
}

Student --|> Person
class Student {
  +drinkBeer()
}
note right of Student::drinkBeer()
  drink Sterni
end note

Professor --|> Employee
class Professor {
  +drinkBeer()
}
note right of Professor::drinkBeer()
  drink Leffe brune
end note

ProfessorEmeritus --|> Professor
class ProfessorEmeritus {
  +drinkBeer()
}
note right of ProfessorEmeritus::drinkBeer()
  drink nomore
end note

Beginner --|> Student
class Beginner {
  +drinkBeer()
}
note right of Beginner::drinkBeer()
  drink RedBull
end note
@enduml

```

# Projektpläne im GANTT Planungsformat (Balkendiagramm)

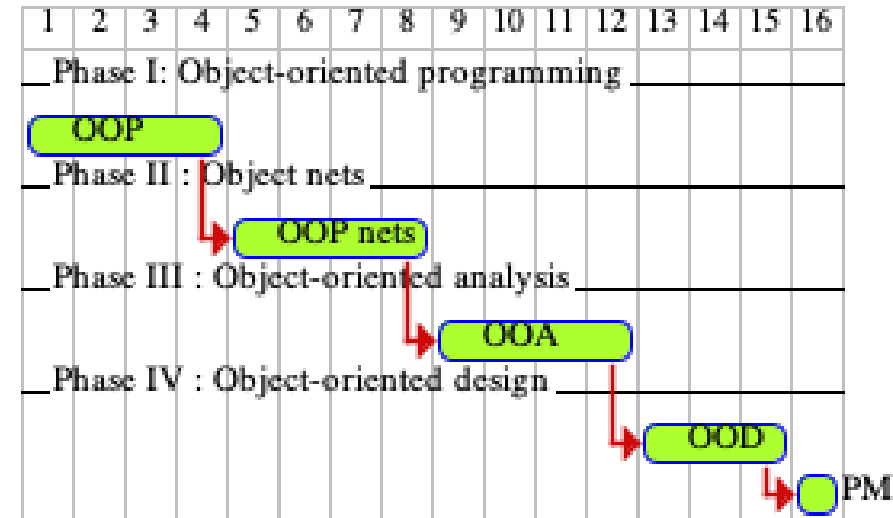
7

Softwaretechnologie (ST)

plantuml/course-outline.plantuml

- ▶ plantuml kann nicht nur UML, sondern auch andere Diagramme
- ▶ GANTT ist wichtig für Projektzeitplanung
- ▶ Achtung: "days" sind hier einfach Zeiteinheiten
- ▶ (Probieren Sie das mal mit "weeks" aus)

```
@startgantt
-- Phase I: Object-oriented programming --
[OOP] lasts 4 days
-- Phase II : Object nets --
[OOP nets] lasts 4 days
[OOP nets] starts at [OOP]'s end
-- Phase III : Object-oriented analysis--
[OOA] lasts 4 days
[OOA] starts at [OOP nets]'s end
-- Phase IV : Object-oriented design--
[OOD] lasts 3 days
[OOD] starts at [OOA]'s end
[PM] lasts 1 day
[PM] starts at [OOD]'s end
@endgantt
```



# Ende

- ▶ Erklären Sie den Begriff “Domänen-spezifische Sprache”. Denken Sie daran, was plantuml von Java unterscheidet.
- ▶ Erklären Sie, ob und warum Sie lieber mit textuellem plantuml oder mit diagrammatischem UML arbeiten.



## 11b) Notation von UML-Klassendiagrammen (UML-CD) mit PlantUML

Prof. Dr. Uwe Aßmann  
Institut für Software- und  
Multimediatechnik  
Lehrstuhl Softwaretechnologie  
Fakultät für Informatik  
TU Dresden  
Version 21-0.1, 19.04.21

- ▶ PlantUML <https://plantuml.com/de/> ist ein tolles Werkzeug für das Zeichnen und automatisierte Layout von UML-Diagrammen
  - Klassendiagramme <https://plantuml.com/de/class-diagram>
  - Sequenzdiagramme <https://plantuml.com/de/sequence-diagram>
  - Komponentendiagramme (eingeschränkt)  
<https://plantuml.com/de/component-diagram>
  - Nutzfalldiagramme <https://plantuml.com/de/use-case-diagram>
  - Zustandsdiagramme <https://plantuml.com/de/state-diagram>
  - Aktivitätsdiagramme <https://plantuml.com/de/activity-diagram-beta>
  - Objektdiagramme <https://plantuml.com/de/object-diagram>
- ▶ Man erstellt eine Datei `datei.plantuml` in textueller Syntax mit "Tripeln"  
(entspricht einfachen Sätzen wie "X ist ein Y")
- ▶ `plantuml datei.plantuml --> datei.png`



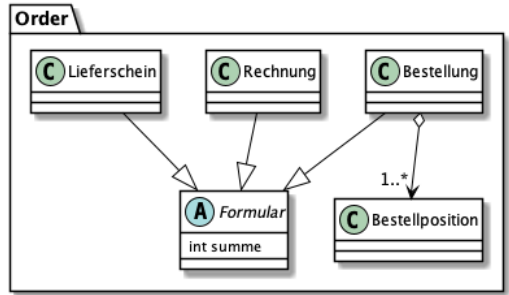
## Nachinstallation mit einem Paketmanager

- ▶ Q: Wie bekommt man `plantuml` auf seinen Rechner?
- ▶ A: mit `brew` oder `apt-get`, den Paketmanagern von Linux/Mac:
- ▶ `brew install plantuml`
- ▶ `sudo apt-get install plantuml`
  
- ▶ Lustig:
- ▶ `plantuml -tutxt diagramm.plantuml`
- ▶ `--` erzeugt utf8text-Diagramm
- ▶ `plantuml -tpdf diagramm.plantuml`
- ▶ `--` erzeugt pdf-Diagramm



```

@startuml
' @author Uwe Assmann
' @version 0.1
' @date 2020-05-30
skinparam class {
  BackgroundColor White
  BorderColor Black
  ArrowColor Black
}
namespace Order {
  abstract class Formular
  Formular : int summe
  Bestellung o--> "1..*" Bestellposition
  Bestellung --|> Formular
  Rechnung --|> Formular
  Lieferschein --|> Formular
}
@enduml
    
```

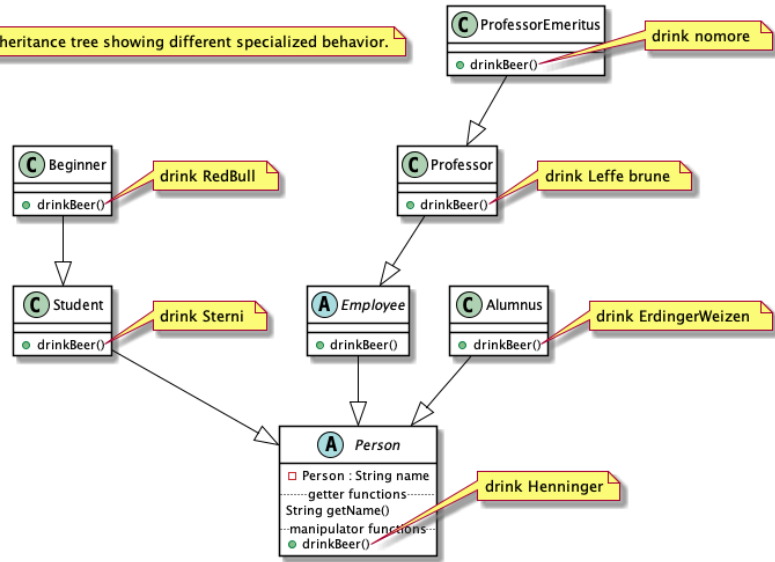


# Personen und Vererbung

plantuml/persons.plantuml

Order

This is a simple class inheritance tree showing different specialized behavior.



```

@startuml
@author Uwe Assmann
@version 0.1
@date 2021-04-12

skinparam class {
    BackgroundColor White
    BorderColor Black
    ArrowColor Black
}

namespace Order {
note "This is a simple class inheritance
tree showing different specialized behavior." as N1
abstract class Person {
    -Person : String name
    .. getter functions ..
    String getName()
    .. manipulator functions ..
    +drinkBeer()
}
note right of Person::drinkBeer()
    drink Henninger
end note
Alumnus --|> Person
class Alumnus {
    +drinkBeer()
}
note right of Alumnus::drinkBeer()
    drink ErdingerWeizen
end note

```

```

Employee --|> Person
abstract class Employee {
    +drinkBeer()
}
Student --|> Person
class Student {
    +drinkBeer()
}
note right of Student::drinkBeer()
    drink Sterni
end note
Professor --|> Employee
class Professor {
    +drinkBeer()
}
note right of Professor::drinkBeer()
    drink Leffe brune
end note
ProfessorEmeritus --|> Professor
class ProfessorEmeritus {
    +drinkBeer()
}
note right of ProfessorEmeritus::drinkBeer()
    drink nomore
end note
Beginner --|> Student
class Beginner {
    +drinkBeer()
}
note right of Beginner::drinkBeer()
    drink RedBull
end note
}
@enduml

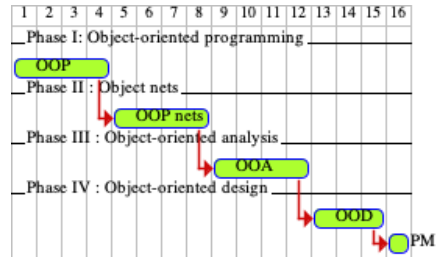
```

# Projektpläne im GANTT Planungsformat (Balkendiagramm)

- ▶ plantuml kann nicht nur UML, sondern auch andere Diagramme
- ▶ GANTT ist wichtig für Projektzeitplanung
- ▶ Achtung: "days" sind hier einfach Zeiteinheiten
- ▶ (Probieren Sie das mal mit "weeks" aus)

```
@startgantt
-- Phase I: Object-oriented programming --
[OOP] lasts 4 days
-- Phase II : Object nets --
[OOP nets] lasts 4 days
[OOP nets] starts at [OOP]'s end
-- Phase III : Object-oriented analysis--
[OOA] lasts 4 days
[OOA] starts at [OOP nets]'s end
-- Phase IV : Object-oriented design--
[OOD] lasts 3 days
[OOD] starts at [OOA]'s end
[PM] lasts 1 day
[PM] starts at [OOD]'s end
@endgantt
```

© Prof. U. Aßmann



# Ende

- ▶ Erklären Sie den Begriff "Domänen-spezifische Sprache". Denken Sie daran, was plantuml von Java unterscheidet.
- ▶ Erklären Sie, ob und warum Sie lieber mit textuellem plantuml oder mit diagrammatischem UML arbeiten.