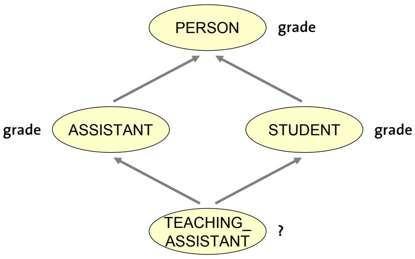
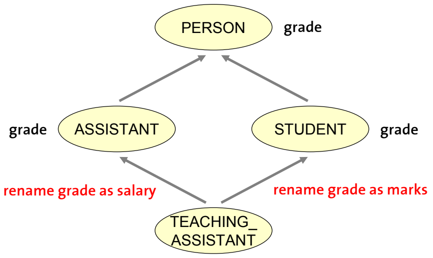
# Exercise Session Informatik III

11. Principles of Inheritance

## Repeated Inheritance



## How to Keep Both Versions

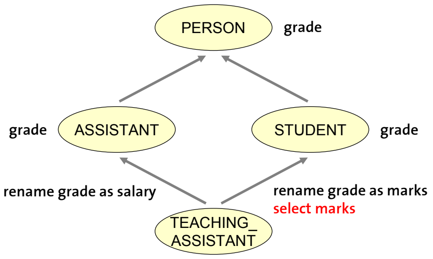


## What Happens When...

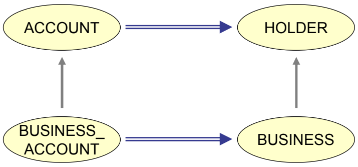
```
class TEST create
   make
feature
   make is
      local
         p: PERSON
      do
         create {TEACHING_ASSISTANT} p
         p.grade
      end
end -- class TEST
```

Polymorphic call to feature **grade**. What should be executed, **salary** or **marks**?

## The Solution: select

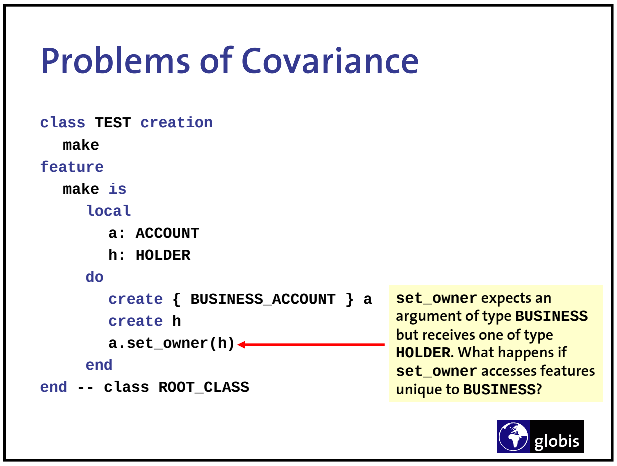


## Covariance



```
class ACCOUNT feature
  set_owner(o: HOLDER) is
    do
      ...
    end
end -- class ACCOUNT
```

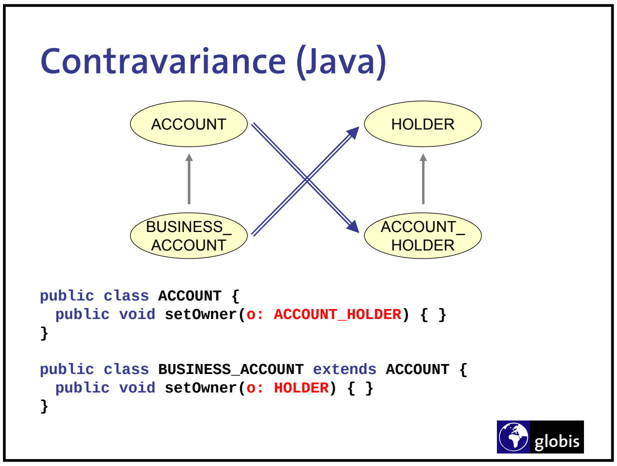
```
class BUSINESS_ACCOUNT
inherit ACCOUNT
  redefine set_owner end
feature
  set_owner(o: BUSINESS) is
    do
      ...
    end
end -- class ACCOUNT
```



# Problems of Covariance

```
class TEST creation
   make
feature
   make is
      local
         a: ACCOUNT
         h: HOLDER
      do
         create { BUSINESS_ACCOUNT } a
         create h
         a.set_owner(h)
      end
end -- class ROOT_CLASS
```

`set_owner` expects an argument of type **BUSINESS** but receives one of type **HOLDER**. What happens if `set_owner` accesses features unique to **BUSINESS**?



# Contravariance (Java)

```
public class ACCOUNT {
  public void setOwner(o: ACCOUNT_HOLDER) { }
}

public class BUSINESS_ACCOUNT extends ACCOUNT {
  public void setOwner(o: HOLDER) { }
}
```

# That's all folks!

**Bis zum nächsten Mal:**

- Task 3 der Übung fertigstellen
- Fragen für Prof. N. Wirth