

django

Übersicht

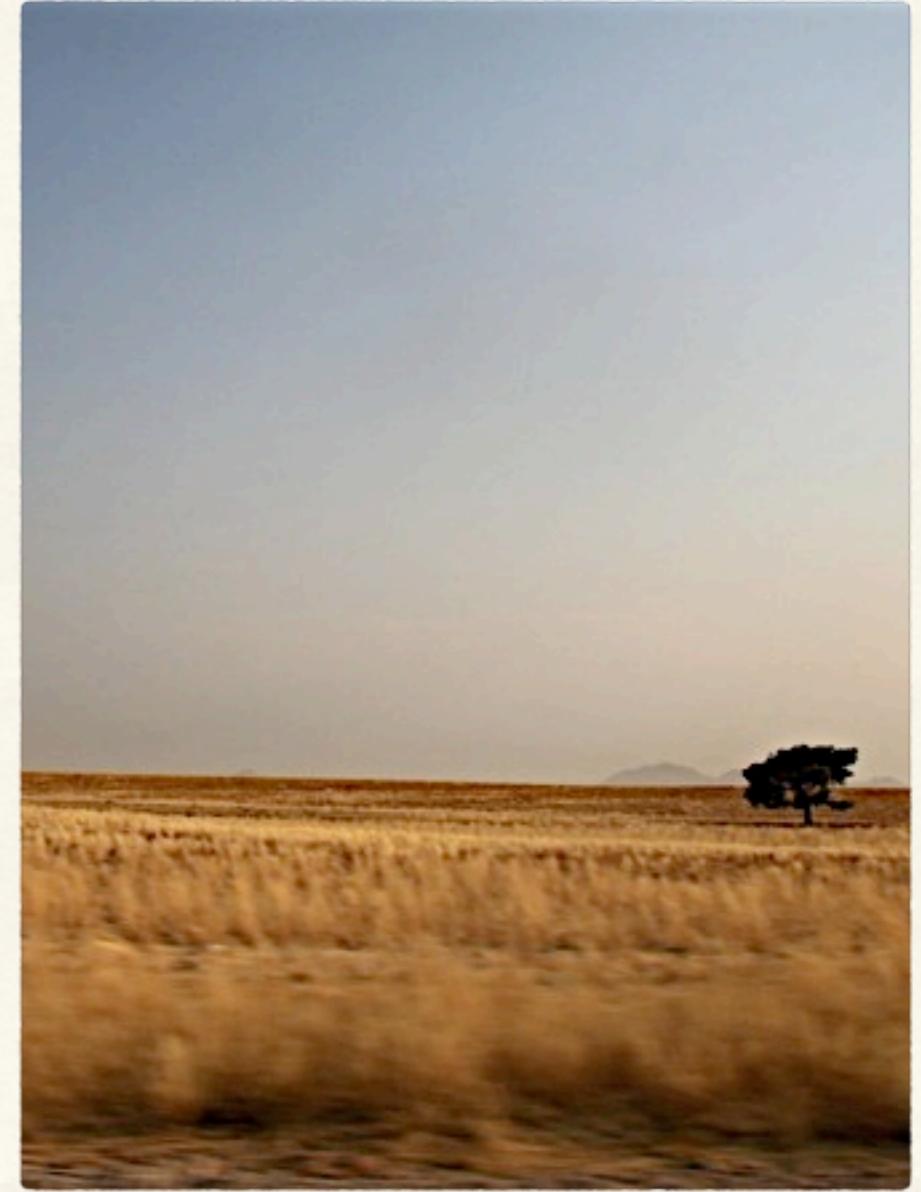
- ▶ Projektdefinition
- ▶ Installation
- ▶ Projektstart
- ▶ Debugging
- ▶ Beispiellösung

Projekt- definition

Grobübersicht

Klassenübersicht

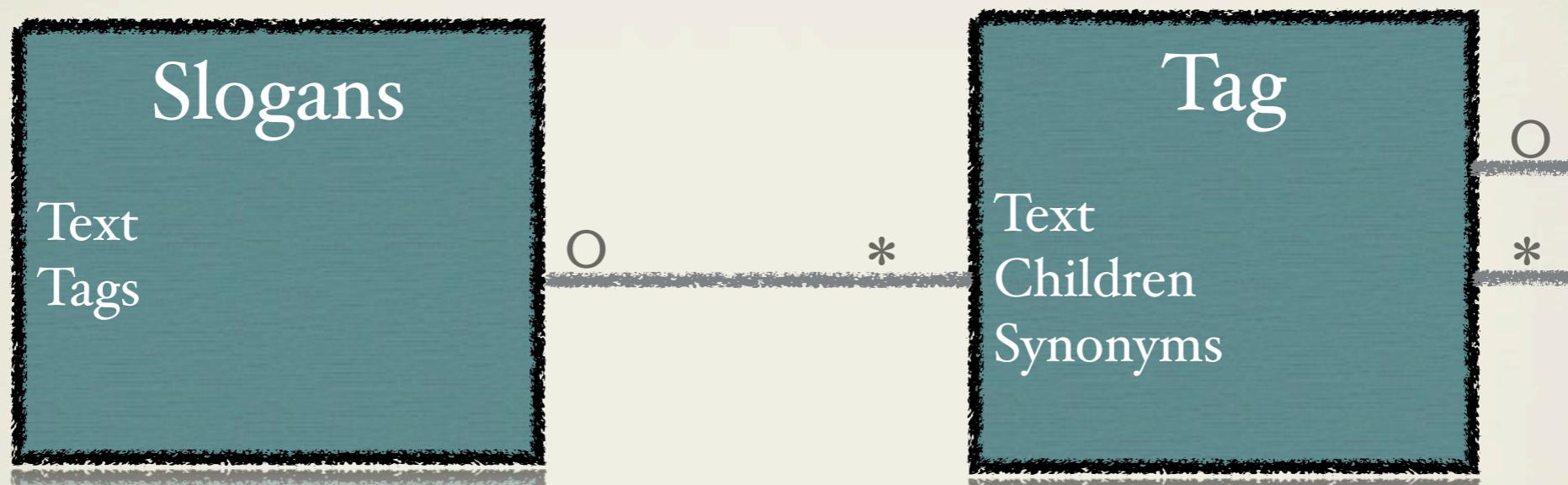
Anforderungen



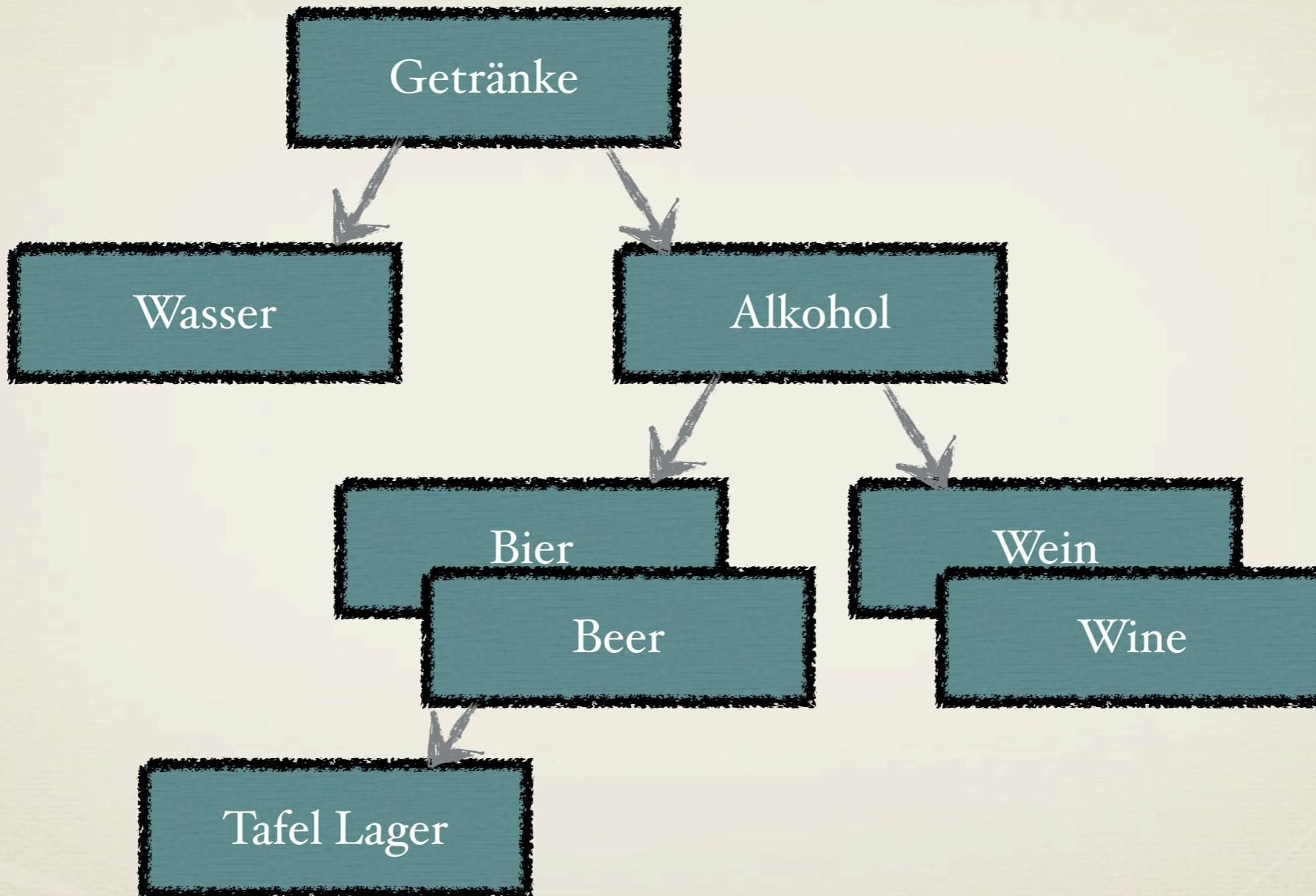
Grobübersicht

- ▶ Sprüche- / Gedichtsammelsurium
- ▶ Sortierung durch Tags möglich
- ▶ Tags sollen auch vererbbar sein

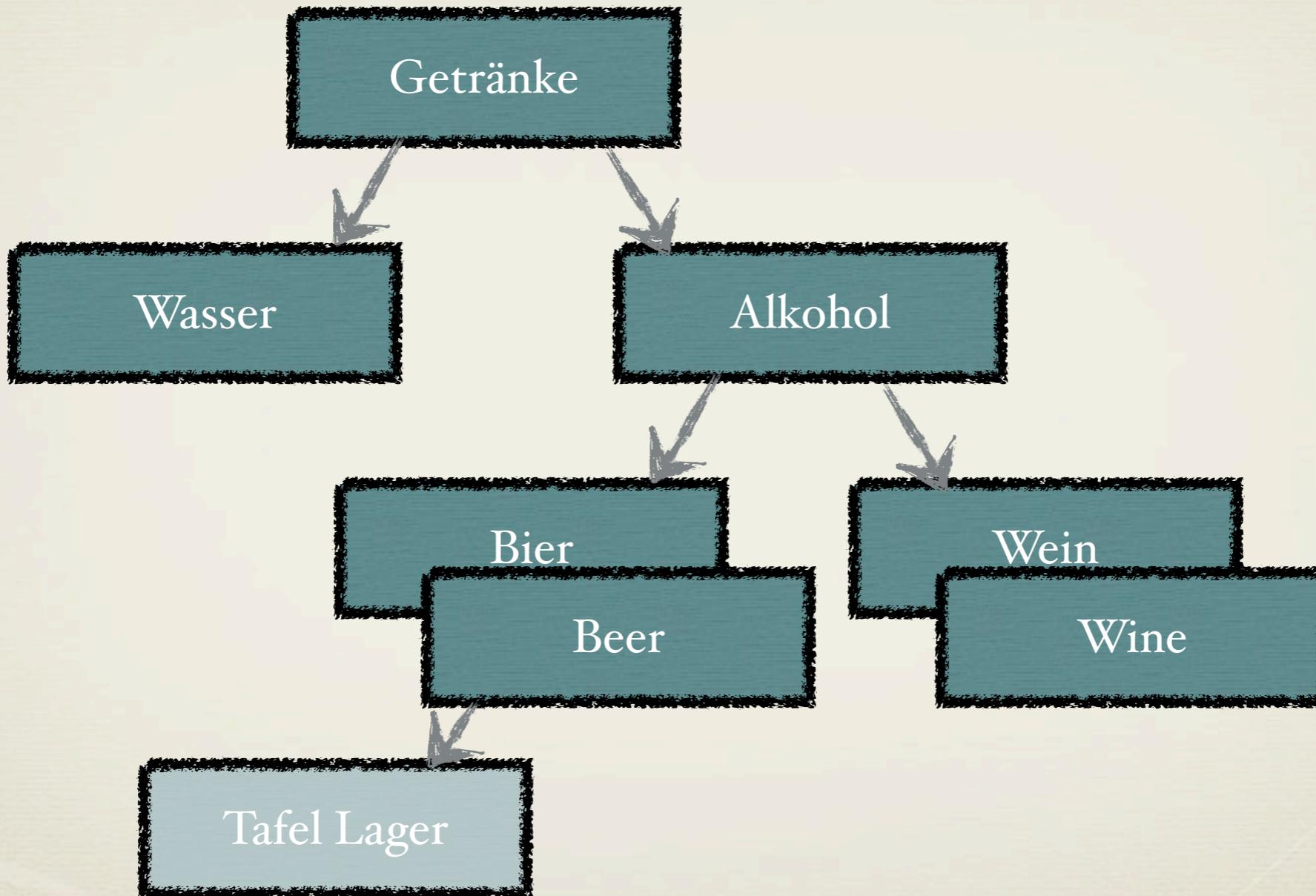
Klassenübersicht



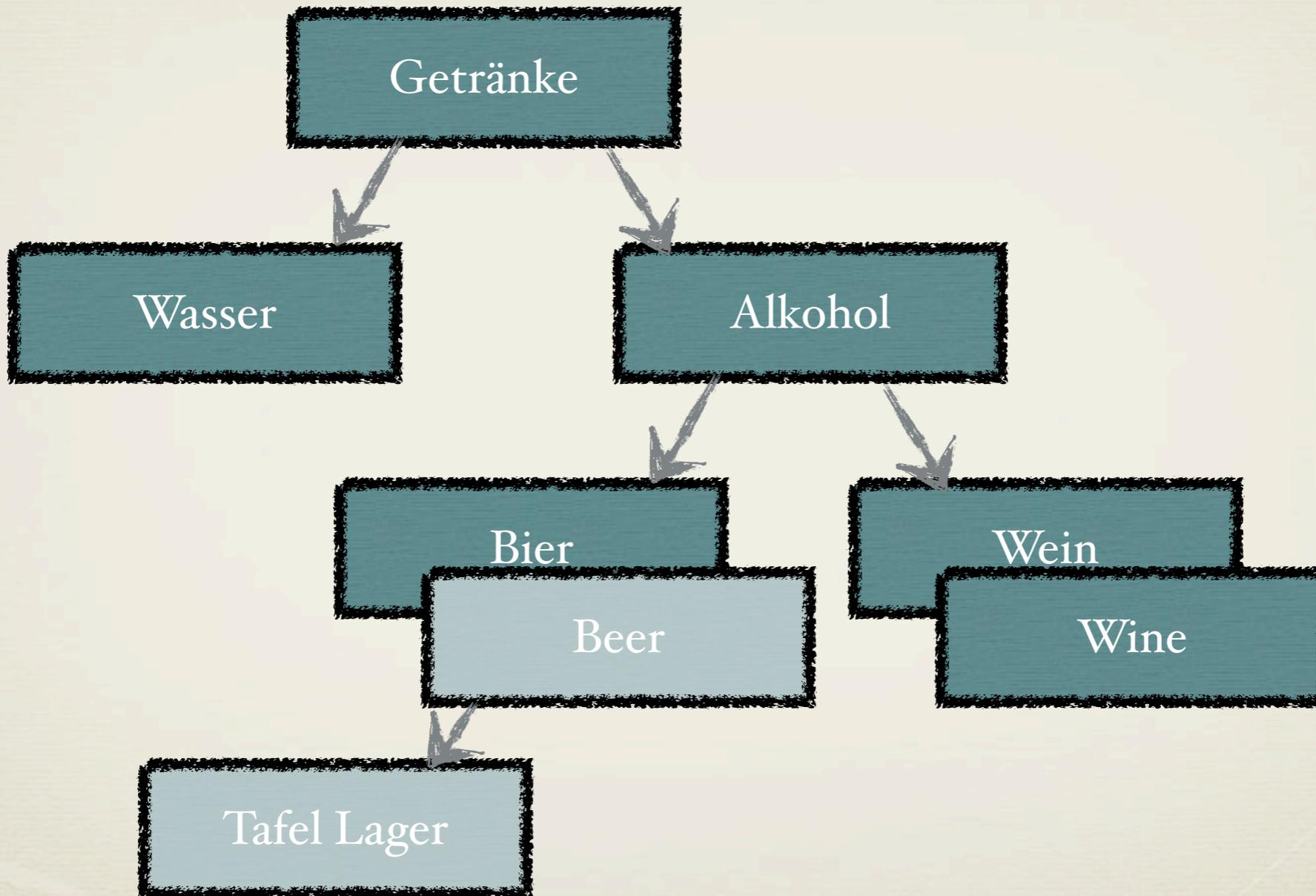
Klassenübersicht



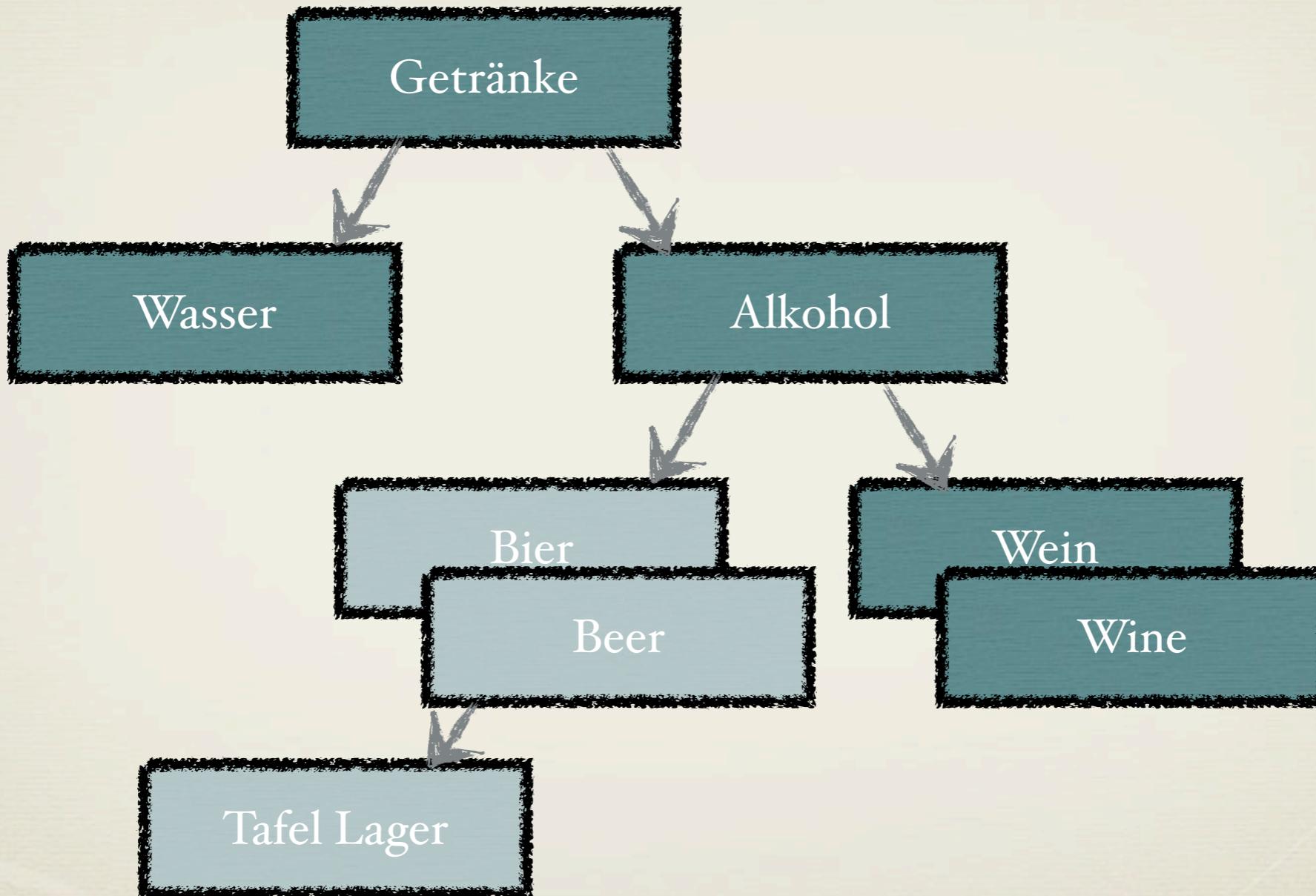
Klassenübersicht



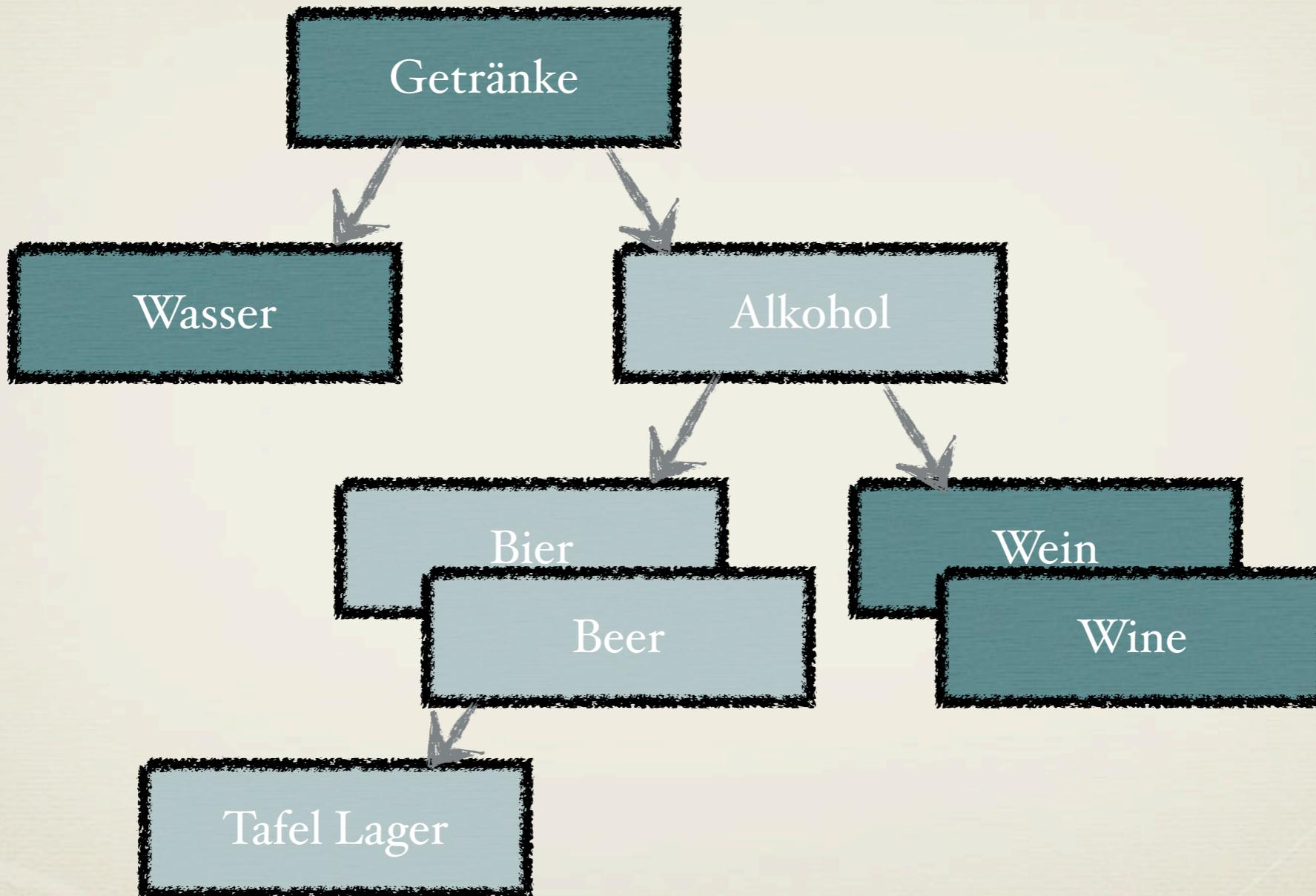
Klassenübersicht



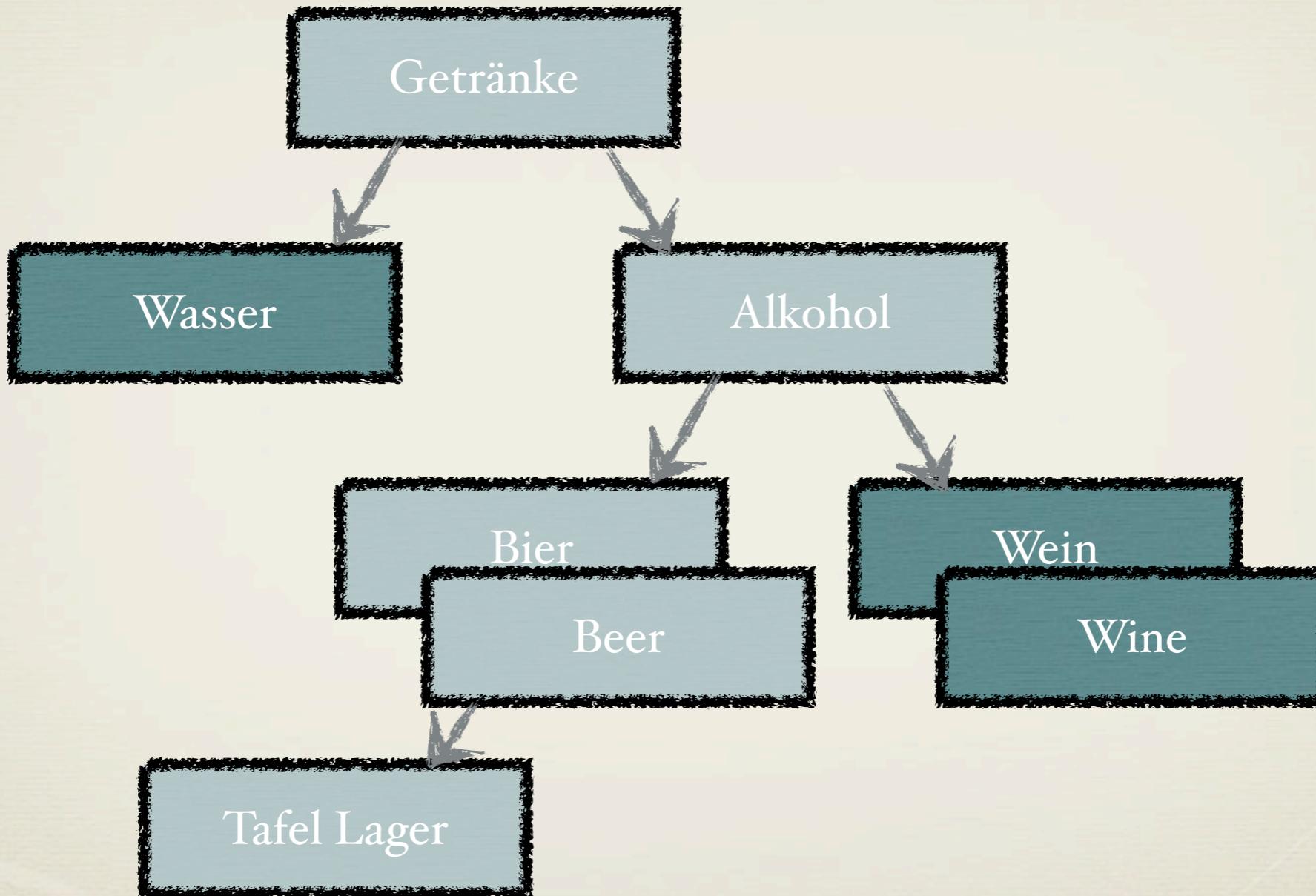
Klassenübersicht



Klassenübersicht



Klassenübersicht



Anforderungen

- ▶ /slogan zeigt zufälligen Spruch
- ▶ /slogan/<id> zeigt Spruch mit Id = <id>
- ▶ /tag zeigt alle Tags
- ▶ /tag/<id> zeigt den Tag, plus Kinder/Synonyme und alle sonstigen Nachfahren des Tags
- ▶ /tag/get?... soll nach Sprüchen mit bestimmten Tags suchen lasen

Installation

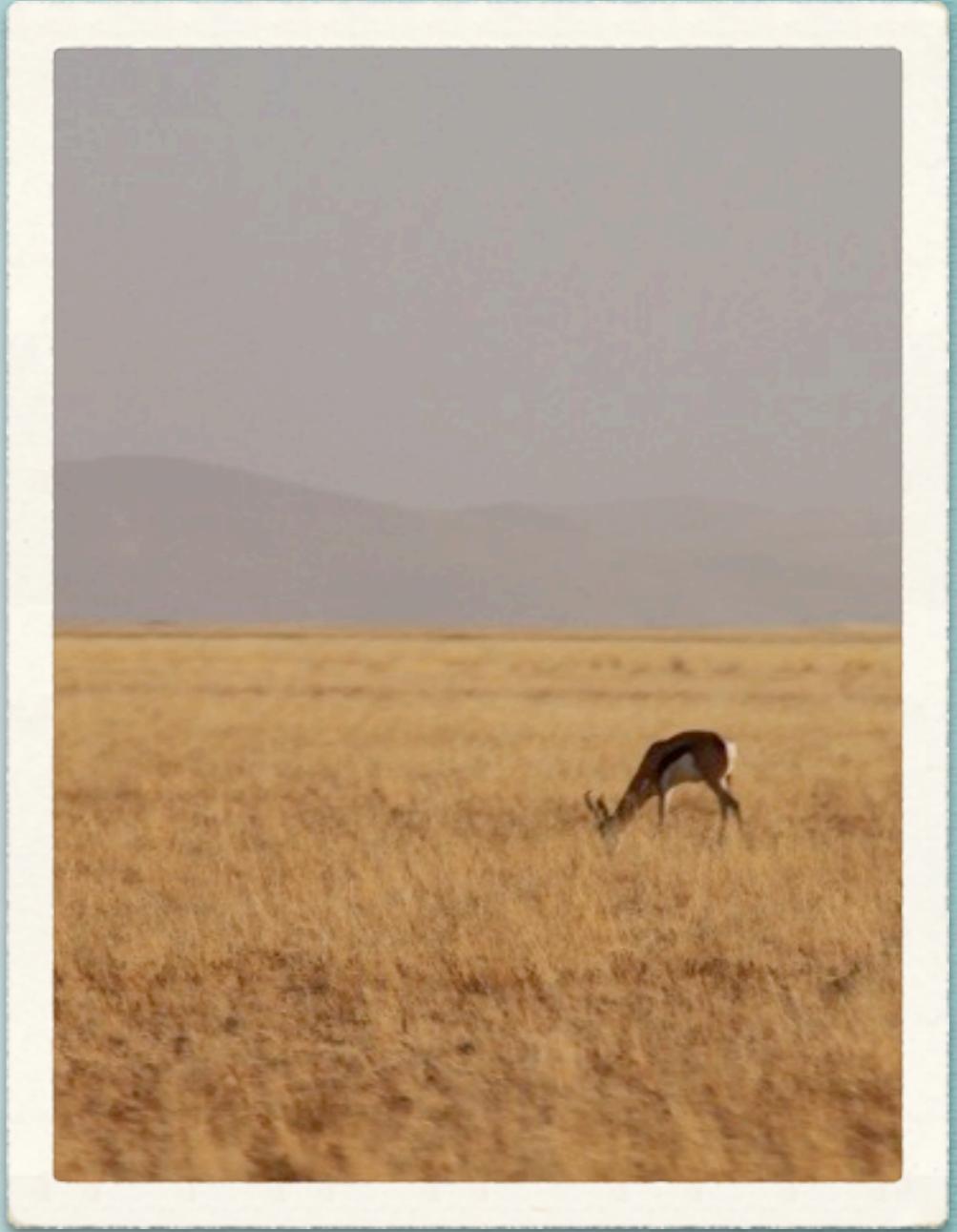
download .tar.gz

<http://djangoproject.com/download>

extract and install

```
tar xzvf Django-1.1.1.tar.gz  
cd Django-1.1.1  
sudo python setup.py install
```

Projektstart



Projektstart

erstelle ein Projekt

django-admin.py startproject truemen

Dateistruktur

```
truemen/  
    __init__.py  
    manage.py  
    settings.py  
    urls.py
```

Konfiguration

settings.py

```
DATABASE_ENGINE = 'sqlite3'      # 'postgresql_psycopg2', 'postgresql',
                                 # 'mysql', 'sqlite3' or 'oracle'.
                                 # Or path to database file if using sqlite3.
DATABASE_NAME = 'dbfile'        # Not used with sqlite3.
DATABASE_USER = ''              # Not used with sqlite3.
DATABASE_PASSWORD = ''          # Set to empty string for localhost.
                                 # Set to empty string for default.
```

Applikation

erstelle ein Projekt

django-admin.py startapp slogans

Dateistruktur

```
slogans/  
    __init__.py  
    models.py  
    tests.py  
    views.py
```

Django Objekte

models.py

```
from django.db import models

# Create your models here.
class Slogan(models.Model):
    text = models.TextField()
    tags = models.ManyToManyField('Tag', blank=True, null=True)

    def __unicode__(self):
        desc = self.text[:17]
        if len(self.text) > 17:
            desc += '...'
        return desc
```

Django Objekte

models.py (Fortsetzung)

```
class Tag(models.Model):
    text = models.CharField(max_length=200)
    children = models.ManyToManyField('self', blank=True, null=True,
symmetrical=False)
    synonyms = models.ManyToManyField('self', blank=True, null=True)

    def __unicode__(self):
        return self.text.replace(' ', '_')
```

Django Objekte

aktualisier die DB Tabellen

```
python manage.py syncdb
```

Django Objekt Handling

```
python manage.py shell
```

```
>>> from truemen.slogans.models import Slogan, Tag  
>>> t = Tag(text='Bier')  
>>> t.save()  
>>> t.id  
1  
...  
>>> tags = Tag.objects.all()  
>>> tags.filter(id=1)
```

Mehr Infos

<http://docs.djangoproject.com/en/dev/intro/tutorial01>

<http://docs.djangoproject.com/en/dev/topics/db/queries/#topics-db-queries>

Admin Interface

urls.py

```
from django.conf.urls.defaults import *

# Uncomment the next two lines to enable the admin:
from django.contrib import admin
admin.autodiscover()

urlpatterns = patterns('',
    # Example:
    # (r'^truemen/', include('truemen.foo.urls')),
    #[...]

    # Uncomment the next line to enable the admin:
    (r'^admin/', include(admin.site.urls)),
)
```

Admin Interface

urls.py

```
INSTALLED_APPS = (  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.sites',  
    'django.contrib.admin',  
)
```

Admin Interface

Starte den Server

`django-admin.py startapp slogans`

Link zum Admininterface

<http://localhost:8000/admin>

Admin Interface

Im slogans Ordner muss folgende Datei angelegt werden damit die Slogans und Tags über das Admininterface bearbeitet werden können.

admin.py

```
from truemen.slogans.models import Slogan, Tag
from django.contrib import admin

admin.site.register(Slogan)
admin.site.register(Tag)
```

Views

urls.py

```
from django.conf.urls.defaults import *

# Uncomment the next two lines to enable the admin:
from django.contrib import admin
admin.autodiscover()

urlpatterns = patterns('',
    # Example:
    # (r'^truemen/', include('truemen.foo.urls')),
    #[ ... ]

    # Uncomment the next line to enable the admin:
    (r'^admin/', include(admin.site.urls)),
    (r'^tag/$', 'truemen.slogans.views.taglist'),
    (r'^tag/(?P<tag_id>\d+)$', 'truemen.slogans.views.tag_detail'),
)
```

Views

views.py

```
from truemen.slogans.models import Tag, Slogan
from django.shortcuts import render_to_response, get_object_or_404

# Create your views here.
def taglist(request):
    tag_list = Tag.objects.all()
    return render_to_response('slogans/taglist.html', {'tag_list': tag_list})

def tag_detail(request, tag_id):
    tag = get_object_or_404(Tag, pk=tag_id)

    return render_to_response('slogans/tag_detail.html', {'tag': tag,
    'child_list':tag.getAllDescentent()} )
```

Views

views.py

```
from truemen.slogans.models import Tag, Slogan
from django.shortcuts import render_to_response, get_object_or_404

# Create your views here.
def taglist(request):
    tag_list = Tag.objects.all()
    return render_to_response('slogans/taglist.html', {'tag_list': tag_list})

def tag_detail(request, tag_id):
    tag = get_object_or_404(Tag, pk=tag_id)

    return render_to_response('slogans/tag_detail.html', {'tag': tag,
    'child_list':tag.getAllDescentent()})
```

Admin Interface

Es muss noch der absolute Pfad zu den Templates angegeben werden.

settings.py

```
TEMPLATE_DIRS = (
    # Put strings here, like "/home/html/django_templates" or "C:/www/django/"
    # Always use forward slashes, even on Windows.
    # Don't forget to use absolute paths, not relative paths.
    '/Users/olafgladis/workspace/templates',
)
```

Views

tag_list.html

```
<h1>Taglist</h1>

{%
    if tag_list %}

        {% for tag in tag_list %}
            <a href='{{tag.id}}'>{{ tag.text }}</a><br/>
        {% endfor %}

    {% else %}
        <p>No tags are available.</p>
    {% endif %}
}
```

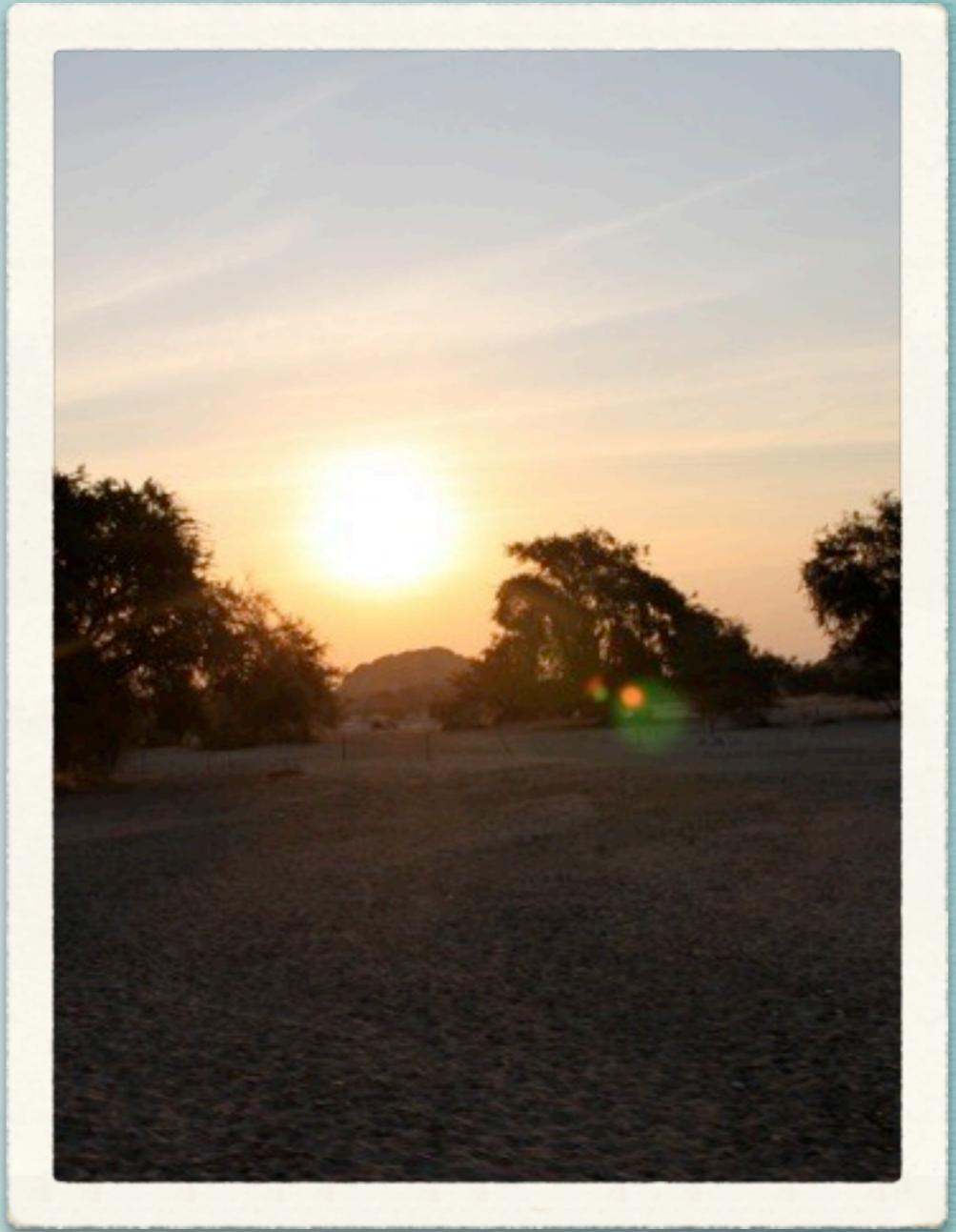
Erstelle die letzten Views

Viel Erfolg!!!

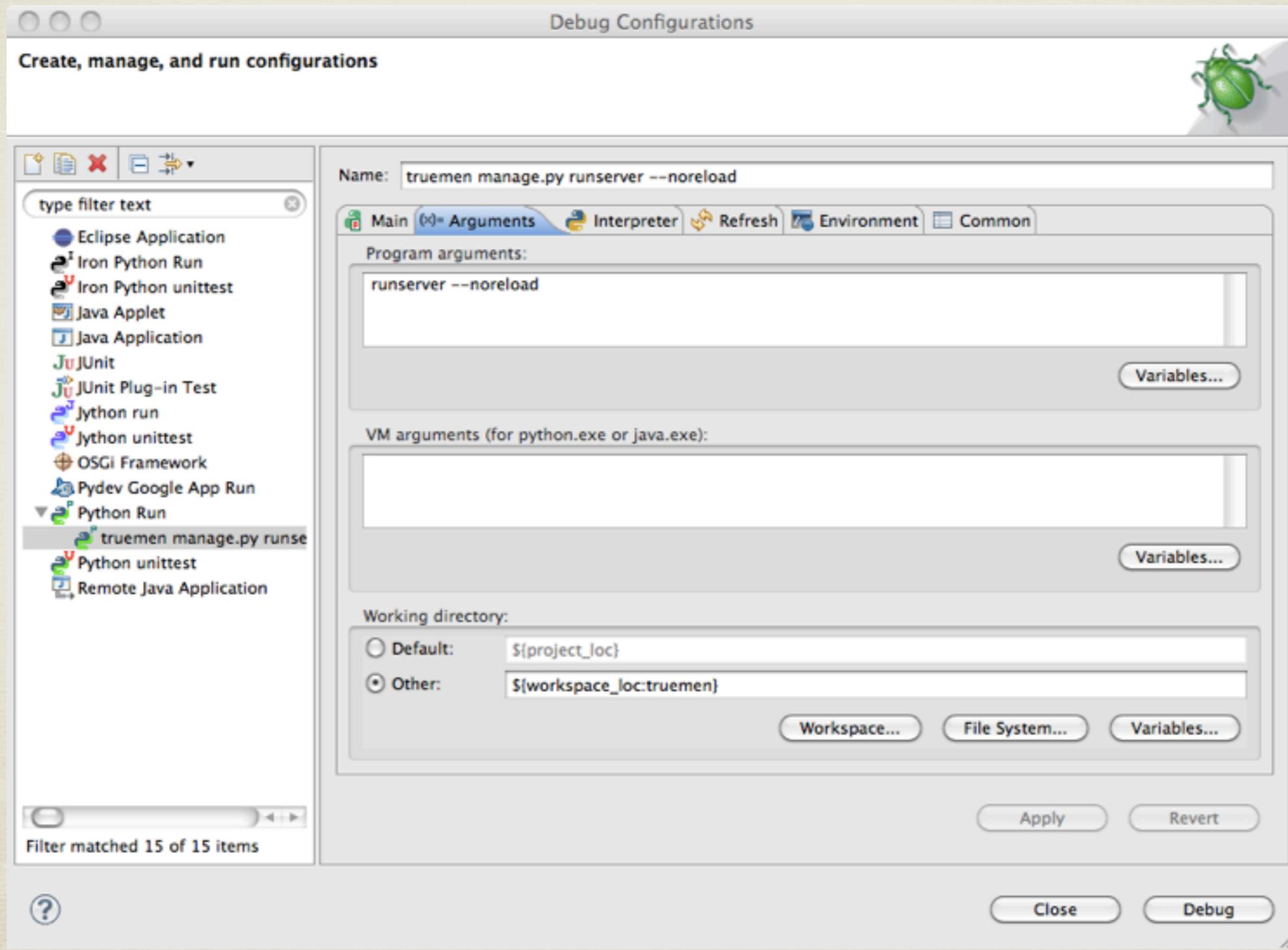
Debugging

pydev

pyCrust Prototyping



pydev



pydev

- ▶ in Eclipse breakpoints setzen
- ▶ Server in Debug-Modus starten
- ▶ Mit dem Browser Seite aufrufen
- ▶ jetzt sollte Eclipse beim auftreffen des breakpoints in den Debug-View wechseln

pyCrust Prototyping

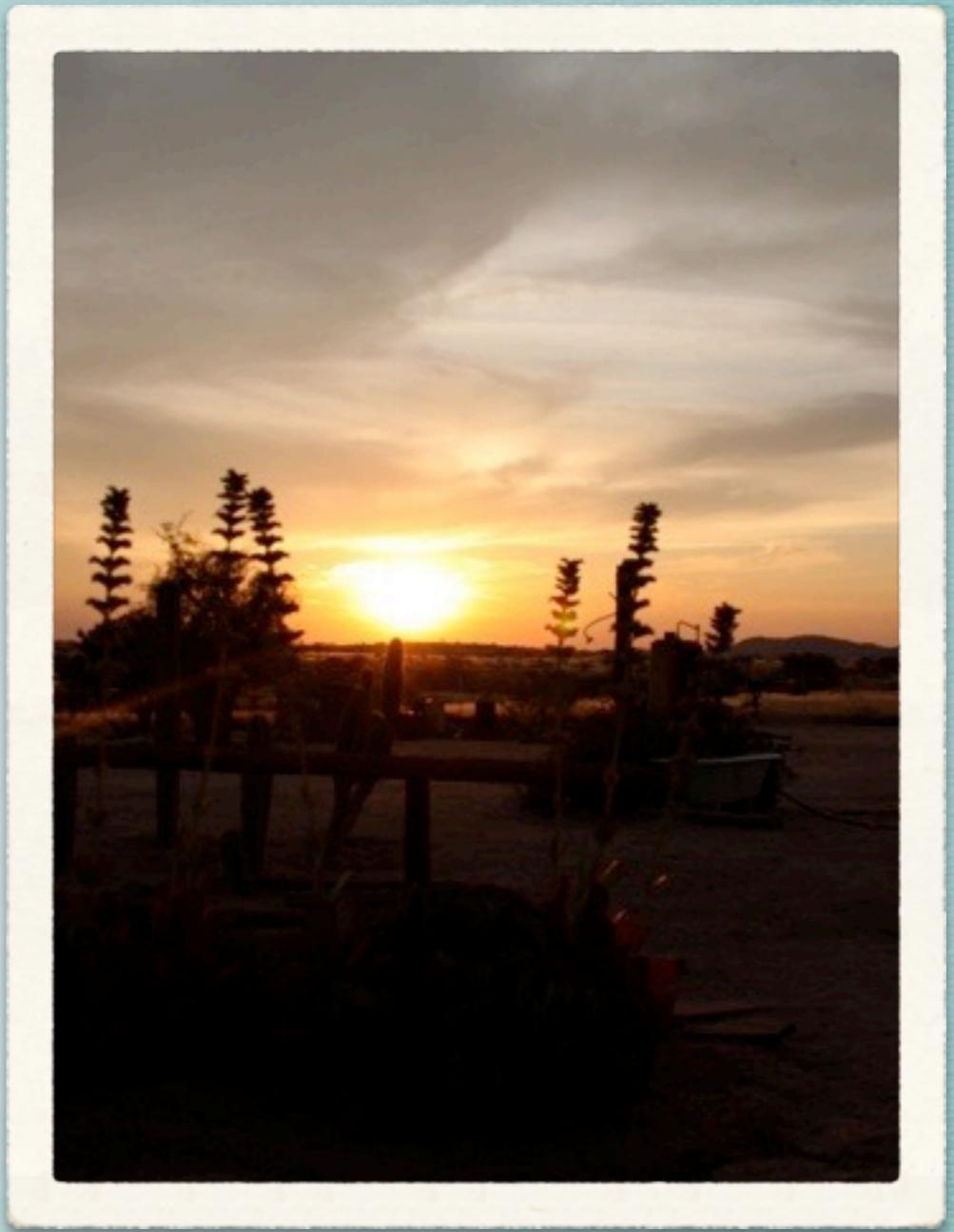
Vorbereitung

```
cd <WORKSPACE>
export PYTHONPATH = $PYTHONPATH:$CWD
export DJANGO_SETTINGS_MODULE=<PROJECY>.settings
pocrust
```

Sinnvolle erste Schritte

```
import <PROJECT>.settings
from <PROJECT>.<APP>.models import <MODEL1>, ...
```

Beispiel Lösung



Beispiel Lösung

Die Dateien können als .zip Datei in Moodle
runtergeladen werden