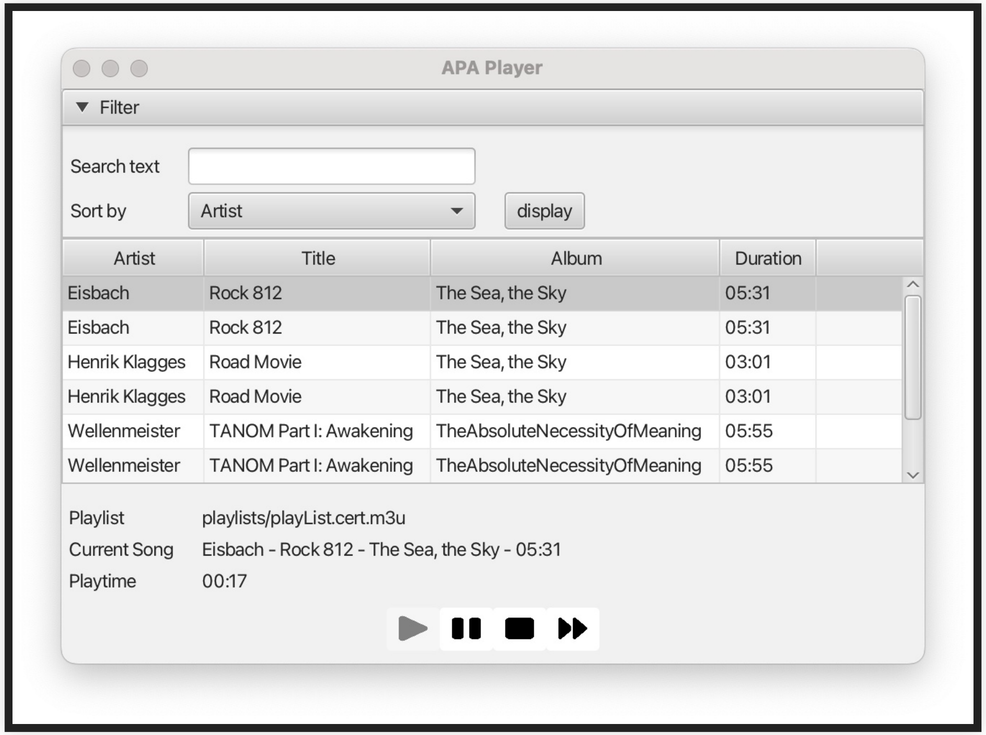
PRACTICAL COURSE ON OBJECT-ORIENTED PROGRAMMING IN JAVA

Let's build a MP3-Player!



Search text

Sort by: Artist

Artist	Title	Album	Duration
Eisbach	Rock 812	The Sea, the Sky	05:31
Eisbach	Rock 812	The Sea, the Sky	05:31
Henrik Klagges	Road Movie	The Sea, the Sky	03:01
Henrik Klagges	Road Movie	The Sea, the Sky	03:01
Wellenmeister	TANOM Part I: Awakening	TheAbsoluteNecessityOfMeaning	05:55
Wellenmeister	TANOM Part I: Awakening	TheAbsoluteNecessityOfMeaning	05:55

Playlist playlists/playList.cert.m3u
Current Song Eisbach - Rock 812 - The Sea, the Sky - 05:31
Playtime 00:17

The aim is to realise a MP3 player using the possibilities of object-oriented programming: Loading audio files, playlists and graphical user interfaces.

Prerequisites

Your personal computer with Java JDK version 13 (at least) and a development environment such as Eclipse. Examples are shown with Eclipse, support for alternative environments is limited!

The practical course is divided into five complex tasks

1. map audio files to classes and process file paths (`ex-06`)
2. implement an inheritance hierarchy for different audio file types (`ex-07`)
3. playlists of audio files (`ex-08`)
4. error handling, sorting and iterating (`ex-09`)
5. graphical user interface (`ex-10`)

Deadlines can be found in the Moodle course room for the APA service

Important: All tasks must be completed on your own to pass the practical course!

Do not adopt any solutions and do not pass on any solutions.

Participants submit their solutions via the THI APA service

Advantage: You can work with the service independently and there are transparent testing criteria.

The APA service learns. Submissions will be checked against submissions already made by other students. Identical submissions can lead to **suspicion of plagiarism**! Suspected plagiarism is assessed manually by the lecturer.

APA SERVICE

All communication is done via email to/from apa@thi.de

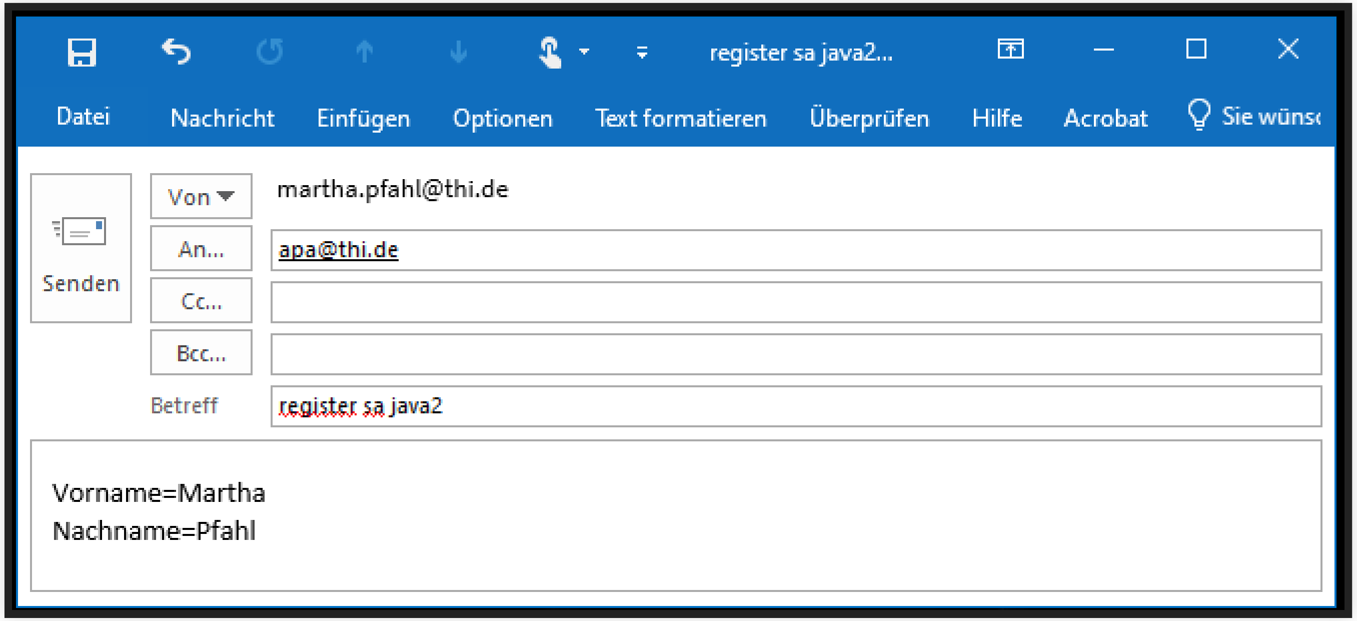
Step 1

At the beginning of the semester, send a one-time registration email to apa@thi.de with the subject “register <lecturer> java2”. As a lecturer, enter the abbreviation of your practical course supervisor, e.g. 'hw' for Prof. Windisch or 'sa' for Prof. Apel.

Please include your first and last name in the email text so that the results of the practical course can be credited to you later. You **must** use the following structure:

```
Vorname=<your first name>
Nachname=<your last name>
```

Example: "Martha Pfahl" is registering for java2 with supervisor Prof. Apel



Von martha.pfahl@thi.de

An... apa@thi.de

Cc...

Bcc...

Betreff register sa java2

Vorname=Martha
Nachname=Pfahl

Step 2

Implement the task, the task description can be found in the Moodle course room for the APA service.

We recommend using Eclipse as your development environment.

Edit and test the task on your local computer. Only when all tests are green is it recommended to submit a solution to the APA service.

Step 3

Send the solution by email to apa@thi.de. The task ID (`ex-06`, `ex-07`, ...) must be used as the subject. The solution is placed in the email as a file attachment, e.g. all Java files that you have implemented yourself.

Important 1: Do not send the test files, the service knows these!

Important 2: Tasks must be delivered in the correct sequence (ex-06, ex-07, ...) and in time.

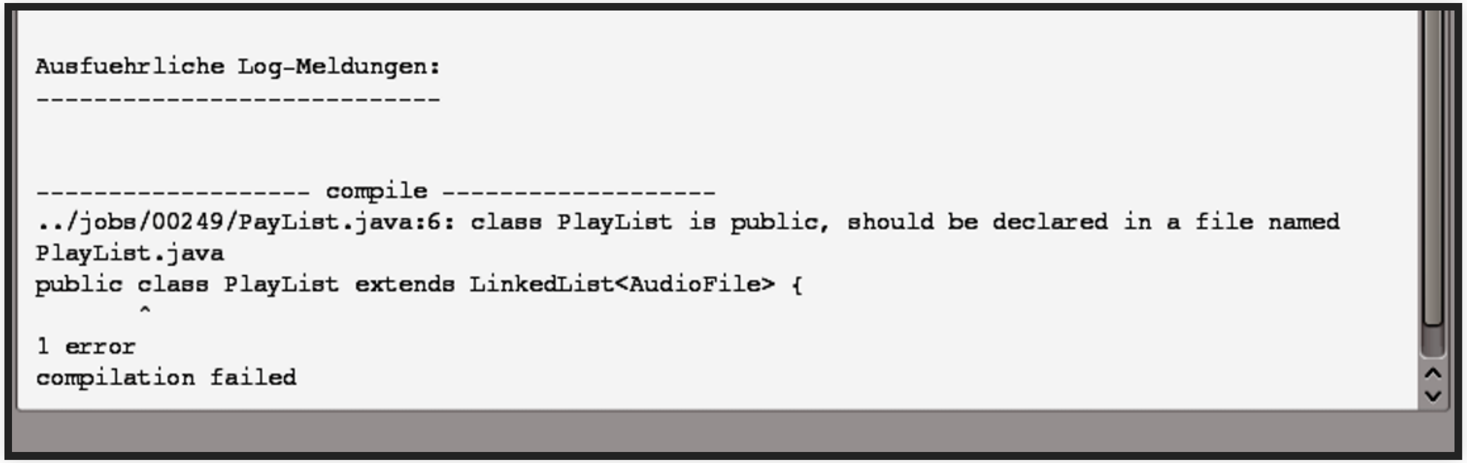
Step 4

Wait until the APA service responds. Possible answers:

- *Compiler error:* something is wrong with the code - make sure there are no syntactic errors in your implementation
- *Test error:* something is wrong with the implementation - You can find the tests in Moodle and run them locally - only if they pass all tests on your local computer it be assumed that they will also pass all tests on the APA server!
- *Formatting errors:* note that the APA service to a limited extent checks the formatting, e.g. indentation using 4 spaces
- *Deadline error:* **the deadlines must be adhered to! Late submissions will result in failure of the practical course.**
- *Suspicion of plagiarism:* A manual check is carried out by the lecturer

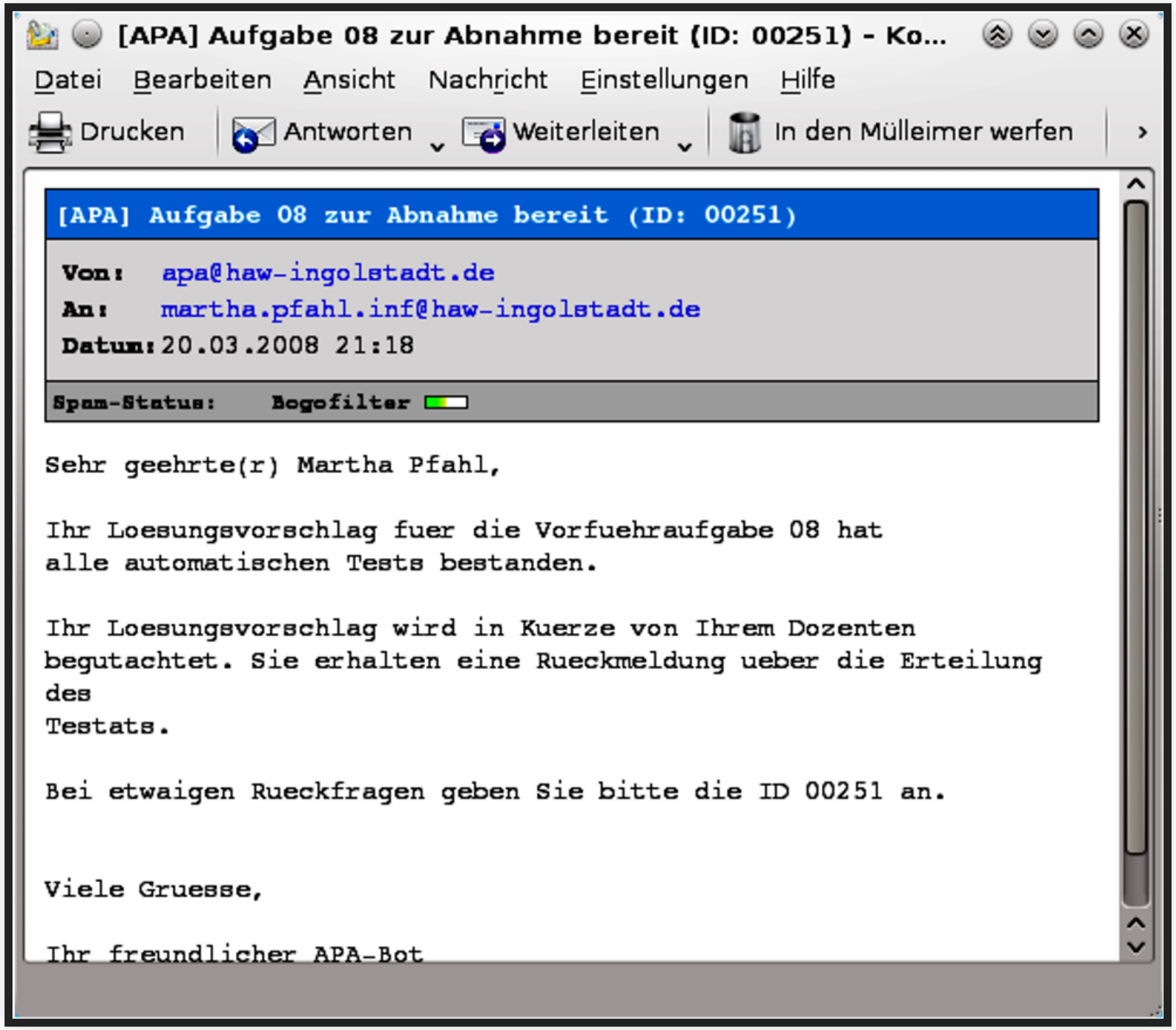
... otherwise you will receive a confirmation that everything is OK!

Example Compiler Error



```
Ausfuehrliche Log-Meldungen:
----------------------------



------------------- compile -------------------
../jobs/00249/PayList.java:6: class PlayList is public, should be declared in a file named
PlayList.java
public class PlayList extends LinkedList<AudioFile> {
       ^
1 error
compilation failed
```

Example Successful Submittance



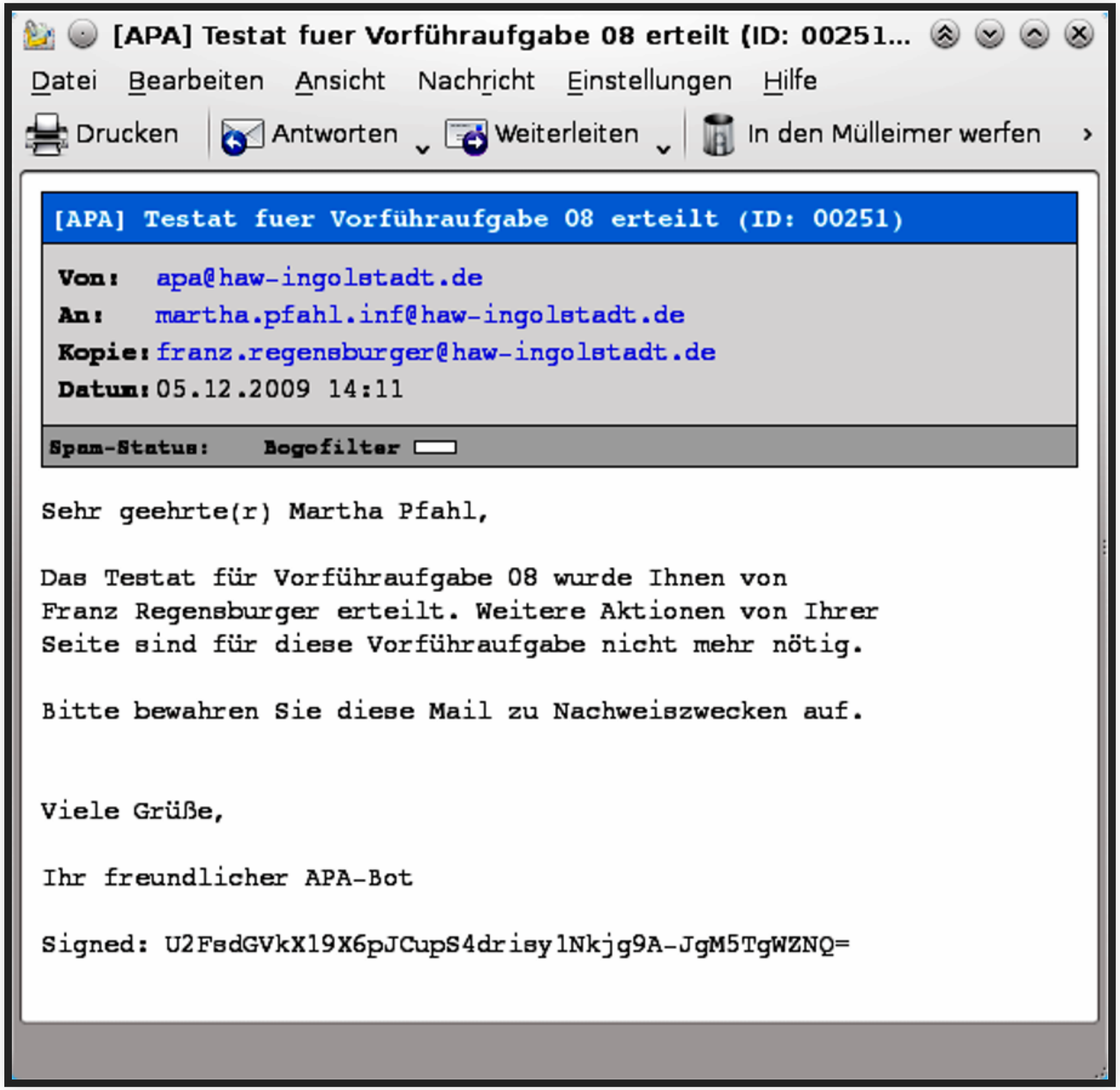
[APA] Aufgabe 08 zur Abnahme bereit (ID: 00251)

Von: apa@haw-ingolstadt.de
An: martha.pfahl.inf@haw-ingolstadt.de
Datum: 20.03.2008 21:18

Spam-Status: **Bogofilter**

```
Sehr geehrte(r) Martha Pfahl,

Ihr Loesungsvorschlag fuer die Vorfuehraufgabe 08 hat
alle automatischen Tests bestanden.

Ihr Loesungsvorschlag wird in Kuerze von Ihrem Dozenten
begutachtet. Sie erhalten eine Rueckmeldung ueber die Erteilung
des
Testats.

Bei etwaigen Rueckfragen geben Sie bitte die ID 00251 an.


Viele Gruesse,

Ihr freundlicher APA-Bot
```

Example Confirmation



[APA] Testat fuer Vorführaufgabe 08 erteilt (ID: 00251)

Von: apa@haw-ingolstadt.de
An: martha.pfahl.inf@haw-ingolstadt.de
Kopie: franz.regensburger@haw-ingolstadt.de
Datum: 05.12.2009 14:11

Spam-Status: **Bogofilter** ▭

Sehr geehrte(r) Martha Pfahl,

Das Testat für Vorführaufgabe 08 wurde Ihnen von
Franz Regensburger erteilt. Weitere Aktionen von Ihrer
Seite sind für diese Vorführaufgabe nicht mehr nötig.

Bitte bewahren Sie diese Mail zu Nachweiszwecken auf.

Viele Grüße,

Ihr freundlicher APA-Bot

Signed: U2FsdGVkX19X6pJCupS4drisy1Nkjg9A-JgM5TgWZNQ=

Step 5

On to the next task and continue with step 3...

... otherwise you have passed the practical course!

PRACTICAL COURSE SESSIONS

- Presentation of the tasks, clarification of ambiguities
- Exercises on various topics from the script on request
- Individual support and help
    - This can be provided ad hoc during the practical course session
    - or by e-mail