

Debugging

Bernd Bruegge, Stephan Krusche, Andreas Seitz, Jan Knobloch
Chair for Applied Software Engineering — Faculty of Informatics



Learning goals

- 1) Explain the history of debugging
- 2) Apply debugging in Eclipse

Debugging

- Debugging is the process of finding and resolving of defects that prevent correct operation of computer software or a system.
- **Early days:** debugging was a cumbersome process
- **Nowadays:** Interactive debugging makes it easy



Definitions

- **Compiler:** computer program that transforms source code into object code to create an executable program
- **Debugger:** computer program that is used to test and debug other programs, to inspect variables and program behavior
- **Breakpoint:** intentional stopping or pausing place in a program, put in place for debugging purposes
 - Exception breakpoints
 - Conditional breakpoints
- **Stack trace:** report active method call hierarchy at a certain point of time for a specific thread during the execution to a software program

```
43 // Performs speech recognition on the audio file
44 SyncRecognizeResponse response = speech.syncRecognize(config, audio);
45 List<SpeechRecognitionResult> results = response.getResultsList();
```

```
▼ Thread [main] (Suspended (exception ApiException))
  Throwables.propagateIfInstanceOf(Throwable, Class<X>) line: 70
  ApiExceptions.callAndTranslateApiException(ApiFuture<ResponseT>) line: 58
  UnaryCallable<RequestT,ResponseT>.call(RequestT) line: 282
  SpeechClient.syncRecognize(SyncRecognizeRequest) line: 269
  SpeechClient.syncRecognize(RecognitionConfig, RecognitionAudio) line: 235
  Main.main(String[]) line: 44
```


Debugging

Bernd Bruegge, Stephan Krusche, Andreas Seitz, Jan Knobloch
Chair for Applied Software Engineering — Faculty of Informatics

