Homework

Dirk Riehle, FAU Erlangen

ADAP A03

Agenda

Programming homework

1. Running example
2. Homework setup
3. Homework list

Design homework

4. To be defined

1. Running Example

Homogenous Names

A homogenous name is a sequence of same-type string components

Examples of homogenous names

- File names e.g. “/usr/bin/tool” or “dirkr\tools\vscode”
  - The file name “C:\user\dirkr\tools” is not homogenous
- Domain names e.g. “uni.de” or “oss.cs.fau.de”
  - The URL “https://moo.uni1.de/login” is not homogenous

The programming homework is to implement homogenous name classes

The Delimiter Character

When printing a homogenous name, name components are to be separated

The separation is performed by using the name's delimiter character, e.g.

- '.' for domain names
- '/' for file names

The printed domain name oss.cs.fau.de uses '.' to separate four components

The Escape Character

An escape character marks the following character as to-use verbatim [1]

Escape characters allow the use of special characters when printing names

This homework only knows two special characters: Delimiter and escape

The escape character is fixed ('\'), the delimiter character is not

[1] See https://en.wikipedia.org/wiki/Escape_character

2. Homework Setup

Homework Setup

The source homework repository is https://github.com/riehlegroup/adap-names

- Not to be confused with the adap-course repository for the slides

Fork this repository to your own account and set the repository to public

- Your repository is where you publish your own homework
- Commit and push to your repo by the homework deadline
- We use the timestamp; no need to tag or version your code
- Do not send pull requests back to the original repository

Your Work Setup

Clone your (forked) GitHub repository to your local machine

Install all needed dependencies

Example

```
git clone https://github.com/<username>/adap-names
cd adap-names
npm install
```

Test Your Work Setup

Ensure that the project builds

```
npm run build
```

Run the public tests

```
npm run test # for all exercises
npm run test:b01 # for exercise B01
```

Check the results

```
Test Files  1 failed (1)
     Tests  3 failed | 1 passed (4)
```

Automated tests

Public tests

- Are included in adap-names and are visible to you
- Can be run by you to test your homework code

Private tests

- Are not part of adap-names and are hence not visible to you
- Cannot be run by you but are used by us to test your work

Important files / directories

`/src`

- Contains the files you need for the homework, e.g.
  - src/adap-b01/names/Name.ts
- We grade only the work done in the respective subfolder
  - Changes to any other files be used in grading (we ignore them so don’t rely on them)

`/test`

- Contains public test files, e.g.
  - test/adap-b01/names/Name.test.ts

Homework Submission

Commit and push your homework before the next lecture

- Ensure your repository is synced
- Your repository must be public
- We grade only the default branch
- Late uploads are not graded

Enable GitHub Actions if you want feedback from our public tests

- This is a preview and does not necessarily reflect your score
- It should provide the same results to running tests locally

Homework Grading

All tests (public and private) are run on our infrastructure

Our repository setup probably differs from yours

We use only files from the *src* directory

We cannot guarantee homework feedback before the holidays

Helpful Resources

How to fork a repository

TypeScript for the New Programmer

TypeScript for Java/C# Programmers

The TypeScript Handbook

3. Homework List

Homework A01 – Introduction

Fill out the following form to provide us with your repository information

- https://forms.gle/Hu85W6VFrtJPp4kc8

Homework B01 – Method Types and Properties

- Implement the adap-b01 Name class as provided
- Use string[] as internal representation of name
- Annotate each method with its method type; example

```
// @methodtype get-method
public getX(): number {
    return this.x;
}
```

- Commit homework by deadline to homework repository

Homework B02 – Class and Interface Design

- Split Name class into Name interface and StringArrayName class
    - StringArrayName uses a string[] as the internal representation of a name
- Add a StringName class that represents a name as a single string
- Ensure that Name instances can be used interchangeably
- Adapt your previous work to this homework as you see fit
- Commit homework by deadline to homework repository

Homework B03 – Subtyping and Inheritance

- Extract AbstractName superclass from StringName and StringArrayName
  - Identify and implement the narrow (minimal) inheritance interface
  - Move as much as you sensibly can into the AbstractName class
- Adapt your previous work to this homework as you see fit
- Commit homework by deadline to homework repository

Homework B04 – Design by Contract

- Identify the names contracts from lecture and documentation
  - Implement preconditions, postconditions, and class invariants
  - Create corresponding component tests for the contract
- Identify the files contracts from lecture and documentation
  - Implement the corresponding preconditions
- Use the exception classes from common as explained in class
- Adapt your previous work to this homework as you see fit
- Commit homework by deadline to homework repository

Homework B05 – Error and Exception Handling

- Implement findNodes() for the Node class hierarchy
  - View a file system as a service with the root node as its interface
  - View the names as just a component used within the file system
- Make the buggy file setup test work as intended
  - Please see the injected fault in BuggyFile.ts
  - No need to complete the file implementations
- Adapt your previous work to this homework as you see fit
- Commit homework by deadline to homework repository

Homework B06 – Value Objects

- Turn Name into a value type, including its implementations
  - Make all value objects **immutable objects** (no need for sharing)
  - You have to adjust the interfaces yourself (no template code provided)
- Ensure implementations correctly fulfill the equality contract
- Adapt your previous work to this homework as you see fit
- Commit homework by deadline to homework repository

Thank you! Any questions?

adap-team@group.riehle.org

https://oss.cs.fau.de

Legal Notices