Software-Engineering Project, Winter 2019/2020

Ralf Hinze,
Annette Bieniusa,
Peter Zeller

AG Programmiersprachen
FB Informatik
TU Kaiserslautern

Participants

- Anna Lieber
- Mahrukh Anwari
- Nithin Vadukkumchery Rajendrakumar
- Sai Sushmitha Kalluri
- Taimoor Bin Khalid
- Tayyaba Seher
- Mohammad Talal Arif
- Vlad-Cristian Constantin

Topic: Persistent data structures

Ephemeral data structures:

- Update modifies in-place
- Original version no longer available

Persistent data structures:

- Old and new version of datastructure available after update

Organization

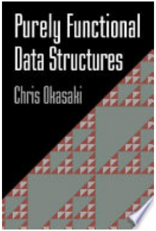
Phase 1:	Reading, Presenting	November 20
Phase 2:	Library Implementation	December 11
Phase 3:	Application Implementation	January 15
Phase 4:	Reflection	February 12

Workload: 8CP ⇒ 200-240 hours ⇒ 15-18 hours/week

Teams of two students.

Languages: Haskell, F#, or Java

Phase 1: Reading, Presenting



Chis Okasaki:
Purely functional data structures

Get it from the library.

Everyone: Read Introduction (Chapters 1 and 2)
Teams of two: Present one topic

	Presentation:
1 Introduction and Lists (Chapter 1 and 2.1)	November 13
2 Search Trees (Chapter 2.2)	November 13
3 Red-black trees (???)	November 20
4 Queues (???)	November 20

Phase 1: Preparation for Phase 2

Build a command-line interface for interacting with priority queues.

```
> empty
$1 = []
> insert 42 $1
$2 = [42]
> insert 7
$3 = [7, 42]
> delete $3
$4 = [42]
> $3
[7, 42]
```

Phase 2: Library Implementation

- Implement different variants of priority queues
- Write automated tests (⇒ cross-testing between teams)

Phase 3: Application

- Build a car traffic simulator
- Discrete time and positions
  - Example: A car with speed $100\frac{km}{h}$ moves 1 place forward every 20 time units.
  -
  - Use priority queue to get next event.
- Bonus: visualization of execution

Phase 4: Reflection

- Review & improve code
- Analyze performance
- . . .

Requirements for command-line interface.

Remember:
Requirements must be precise enough to run tests on other teams.

```
> empty
$1 = []
> insert 42 $1
$2 = [42]
> insert 7
$3 = [7, 42]
> delete $3
$4 = [42]
> $3
[7, 42]
```