Software Language Engineering

Tijs van der Storm

CWI

Centrum Wiskunde & Informatica

university of groningen

- Week 1: Introduction (Chapters 1 & 2)
- Week 2: Concrete syntax (Chapters 6 & 7)
- Week 3: Abstract syntax (Chapters 3 & 4)
- Week 4: Checking (Chapter 9)
- Week 5: Interpretation (Chapters 5 & 8)
- Week 6: Code generation (Chapter 5)
- Week 7: Transformation (Chapters 5 & 12)
- Week 8: Wrap up & grading of lab starts

```
form taxOfficeExample {
  "Did you sell a house in 2010?"
    hasSoldHouse: boolean

  "Did you buy a house in 2010?"
    hasBoughtHouse: boolean

  "Did you enter a loan?"
    hasMaintLoan: boolean

  if (hasSoldHouse) {
    "What was the selling price?"
      sellingPrice: integer
    "Private debts for the sold house:"
      privateDebt: integer
    "Value residue:"
      valueResidue: integer =
        sellingPrice - privateDebt

  }
}
```

- Concrete syntax
- Abstract syntax
- Name resolution
- Type checking
- Interpretation
- Code generation
- Normalization
- Rename refactoring

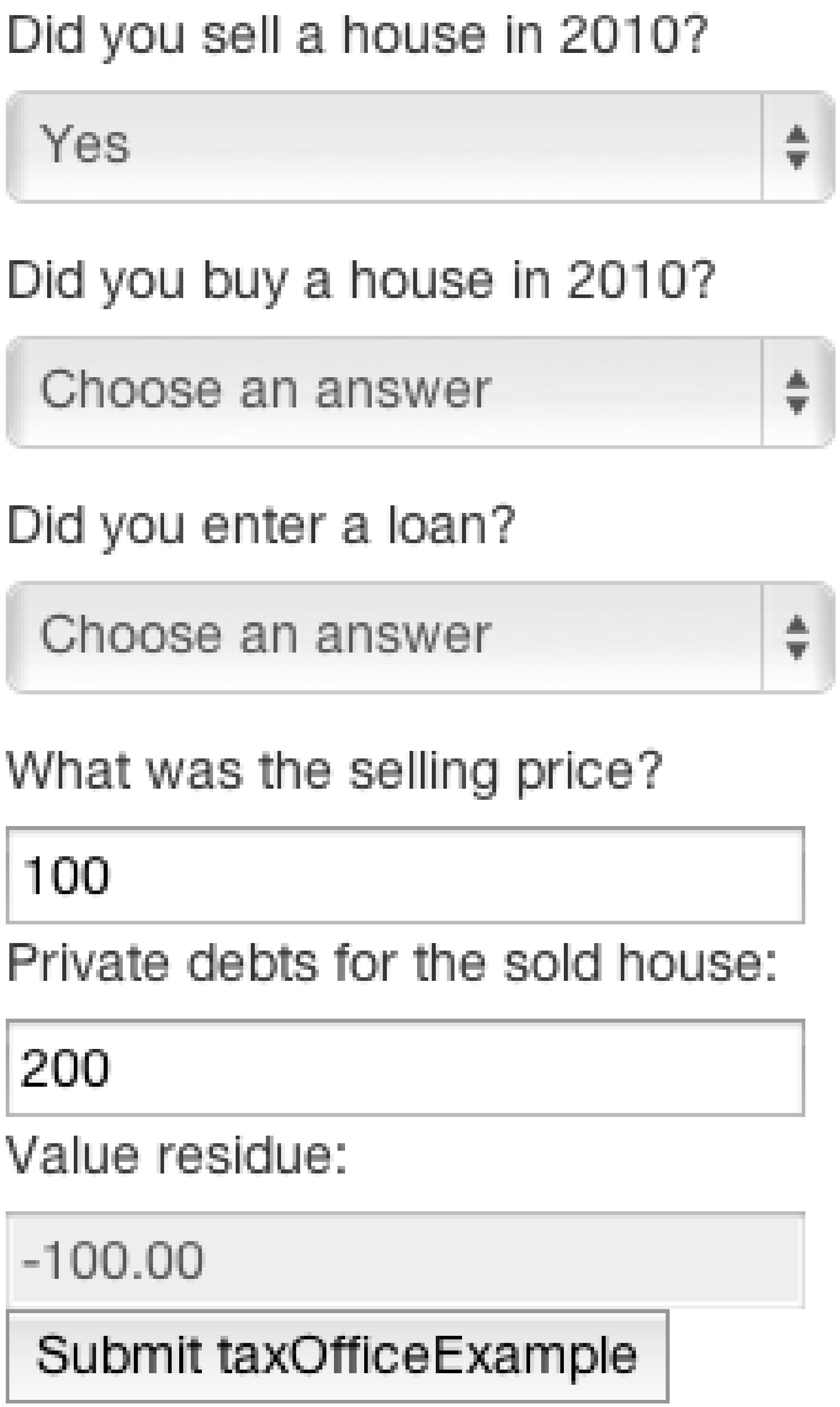
```
form taxOfficeExample {
  "Did you sell a house in 2010?"
    hasSoldHouse: boolean

  "Did you buy a house in 2010?"
    hasBoughtHouse: boolean

  "Did you enter a loan?"
    hasMaintLoan: boolean

  if (hasSoldHouse) {
    "What was the selling price?"
      sellingPrice: integer
    "Private debts for the sold house:"
      privateDebt: integer
    "Value residue:"
      valueResidue: integer =
        sellingPrice - privateDebt

  }
}
```



Did you sell a house in 2010?

Yes

Did you buy a house in 2010?

Choose an answer

Did you enter a loan?

Choose an answer

What was the selling price?

100

Private debts for the sold house:

200

Value residue:

-100.00

Some action points

- Send link to github repo + full names + student nr of team
- Send a screenshot of a running questionnaire :)

Checking the lab

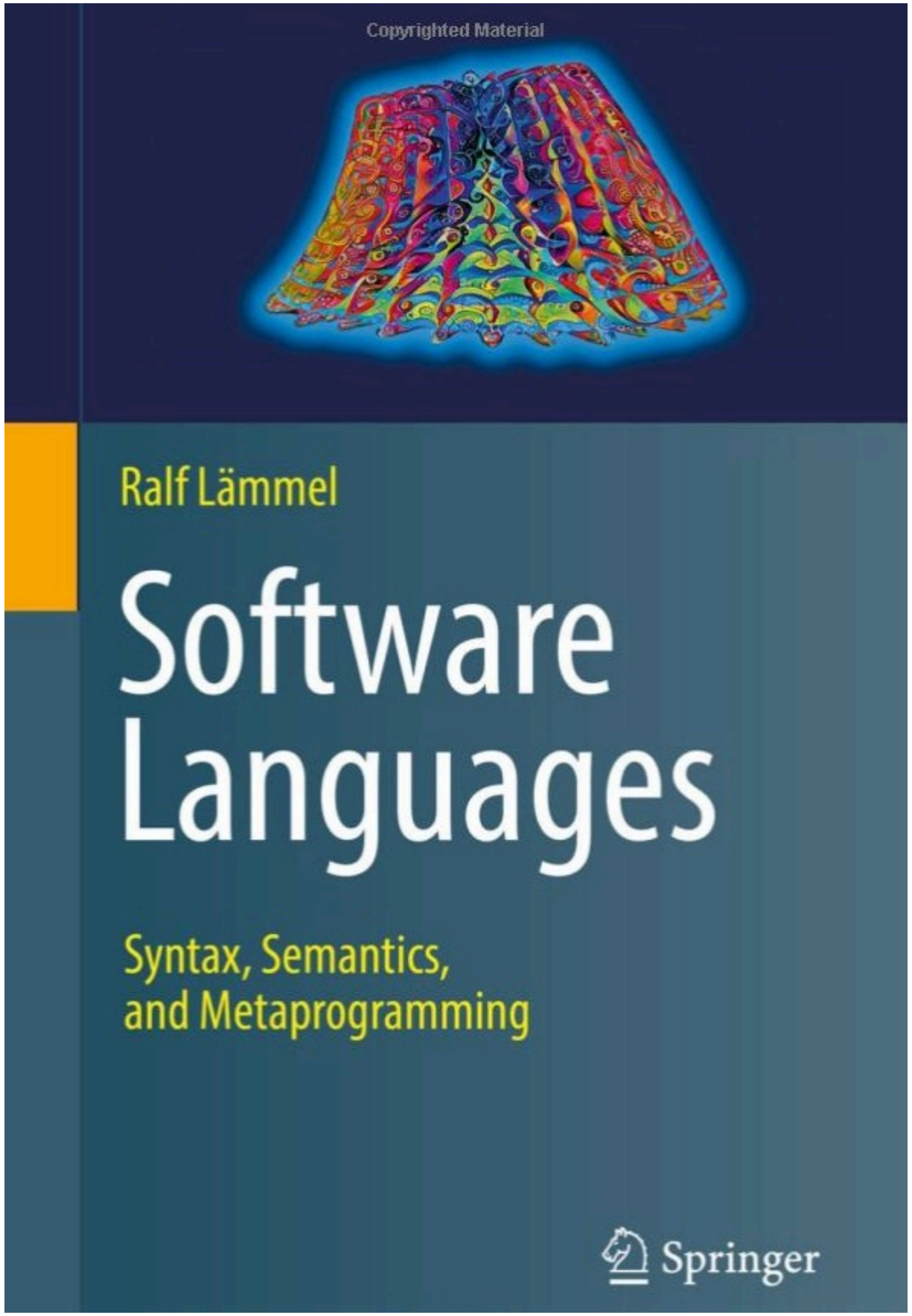
- Example QL program: running
- Introduce error and marking
- Hyperlinking works
- Interpreter demo
- Brief survey of the code

Code quality

- Please, please, please, clean up code
  - no commented out stuff, dead code, failed trials etc.
- Consistent layout and indentation
- Communicating code over comments
- Otherwise I’ll be grumpy ;)

Exam

- Fri 25/1 14:00–17:00
- MartiniPlaza



Copyrighted Material

Ralf Lämmel

Software Languages

Syntax, Semantics, and Metaprogramming

Springer

Ralf Lämmel

+ slides

Example questions

- Define a grammar for… <well-known language x>
- Define abstract syntax ADT for <some grammar>
- Categorize <X> regarding <Y>
- Extend QL with a construct <X>, and elaborate
- Draw name graph on top of program <X>
- Explain difference between <X> and <Y>

Example questions

- Explain drawback of <X> over <Y>
- Mention *n* concrete examples of <X>
- Provide a definition for <X>
- Identify *n* errors in the following program
- Sketch an interpreter for language <X>
- Sketch the basic architecture of <X>