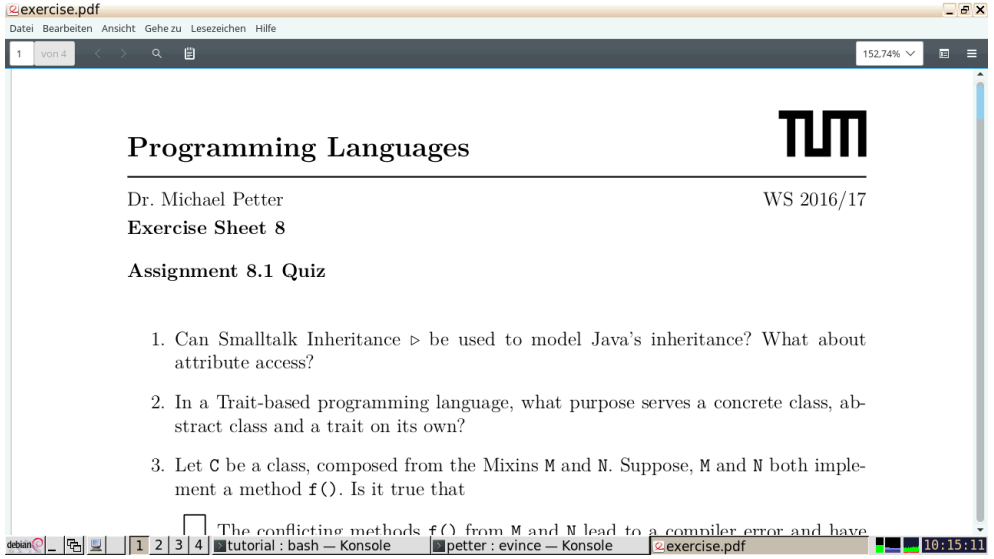
Script **generated by TTT**

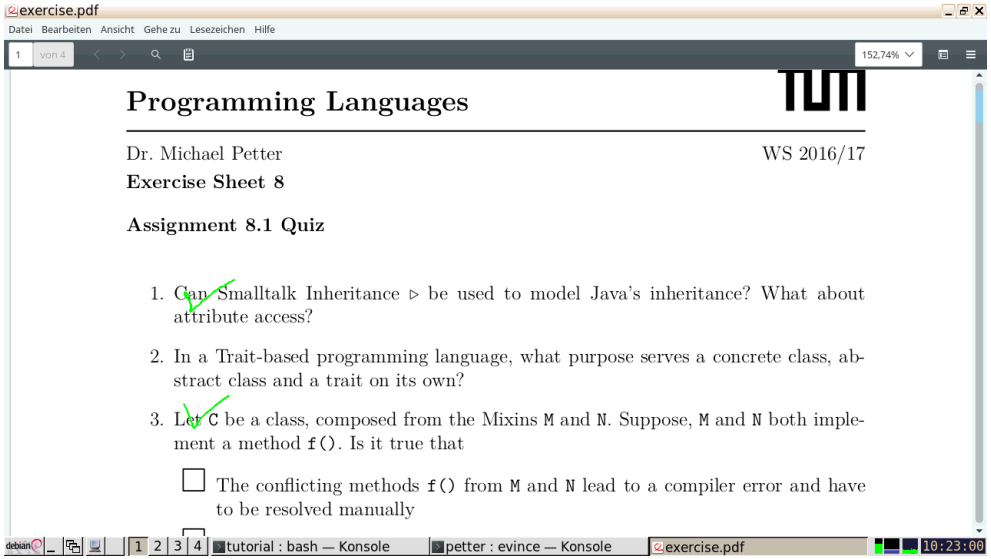
Title: Petter: Programmiersprachen_Uebung (27.01.2017)

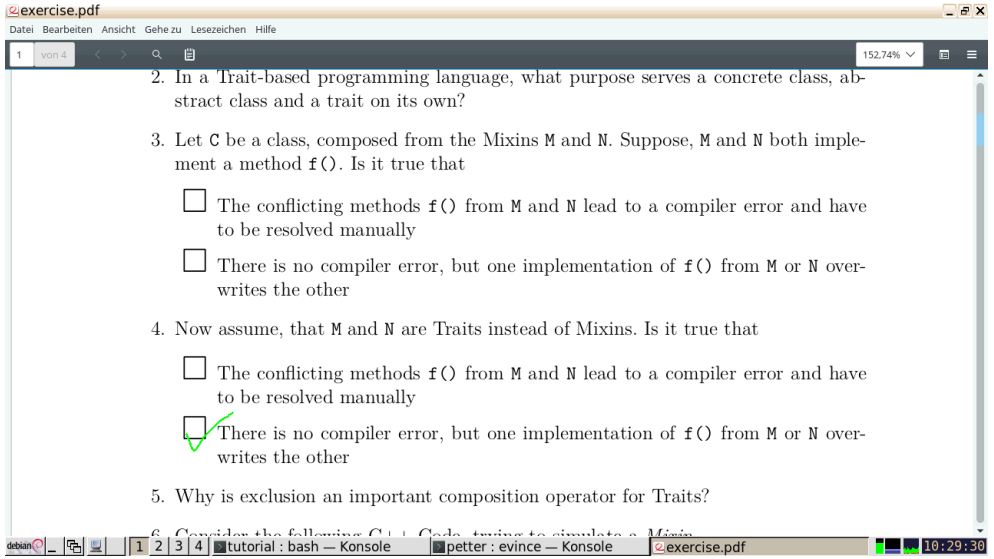
Date: Fri Jan 27 10:15:12 CET 2017

Duration: 98:40 min

Pages: 24







exercise.pdf

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

☐ can not be resolved

7. Given the traits T_1, T_2 and class C:

$$T_1 = \{a \mapsto 1\}$$

$$T_2 = \{a \mapsto 2\}$$

$$C = \langle \emptyset, nil \triangleright (T_1 + T_2) \rangle$$

tutorial : bash — Konsole | petter : evince — Konsole | exercise.pdf | 10:36:30

exercise.pdf

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

$$T_1 = \{a \mapsto 1\}$$

$$T_2 = \{a \mapsto 2\}$$

$$C = \langle \emptyset, nil \triangleright (T_1 + T_2) \rangle$$

What is the value of $(C \triangleright T_1)(a)$?

☐$\top$ ☐1 ☐$\bot$

8. **(Attention: Several answers might be true for this question!)**
$c_1 \sqcup c_2 = c_1 \sqcup c_2$ is true for

- ☐ $c_1 = \{a = 0x1\}, \quad c_2 = \{b = 0x1\}$
- ☐ $c_1 = mixin(c_3)(c_2), \quad c_2 = \{a = 0x1\}, \quad c_3 = \{a = 0x2\}$
- ☐ $c_1 = mixin(c_2)(c_3), \quad c_2 = \{a = 0x1\}, \quad c_3 = \{a = 0x2\}$
- ☑ $c_1 = mixin(c_3)(c_4), \quad c_2 = c_3 \triangleright c_4, \quad c_3 = \{a = 0x1\}, \quad c_4 = \{a = 0x2\}$

Assignment 8.2 Having fun with Mixins

tutorial : bash — Konsole | petter : evince — Konsole | exercise.pdf | 10:45:59

exercise.pdf

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

$$C = \langle \emptyset, nil \triangleright (T_1 + T_2) \rangle$$

What is the value of $(C \triangleright T_1)(a)$?

☐$\top$ ☐1 ☐$\bot$

8. **(Attention: Several answers might be true for this question!)**
$c_1 \sqcup c_2 = c_1 \sqcup c_2$ is true for

- ☐ $c_1 = \{a = 0x1\}, \quad c_2 = \{b = 0x1\}$
- ☐ $c_1 = mixin(c_3)(c_2), \quad c_2 = \{a = 0x1\}, \quad c_3 = \{a = 0x2\}$
- ☐ $c_1 = mixin(c_2)(c_3), \quad c_2 = \{a = 0x1\}, \quad c_3 = \{a = 0x2\}$
- ☐ $c_1 = mixin(c_3)(c_4), \quad c_2 = c_3 \triangleright c_4, \quad c_3 = \{a = 0x1\}, \quad c_4 = \{a = 0x2\}$

Assignment 8.2 Having fun with Mixins

Reconsider the example from the lecture about synchronized file- and socket-streams. The following classes are given:

$$FileStream = \{read = 0x1, write = 0x2\}$$

tutorial : bash — Konsole | petter : evince — Konsole | exercise.pdf | 11:08:22

exercise.pdf

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

Reconsider the example from the lecture about synchronized file- and socket-streams. The following classes are given:

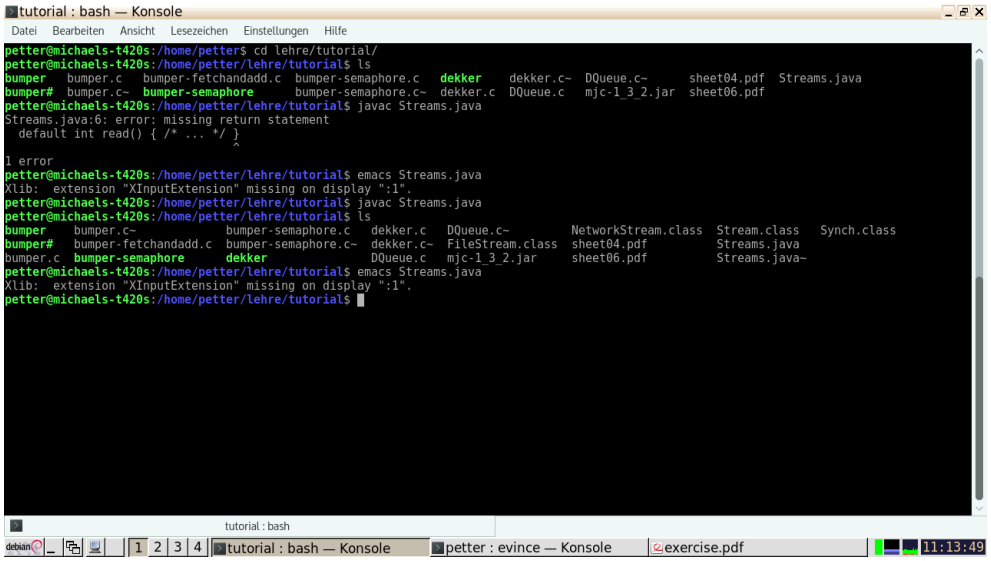
$$FileStream = \{read = 0x1, write = 0x2\}$$
$$SocketStream = \{read = 0x3, write = 0x4\}$$
$$SyncRW = \{read = 0x5, write = 0x6\}$$

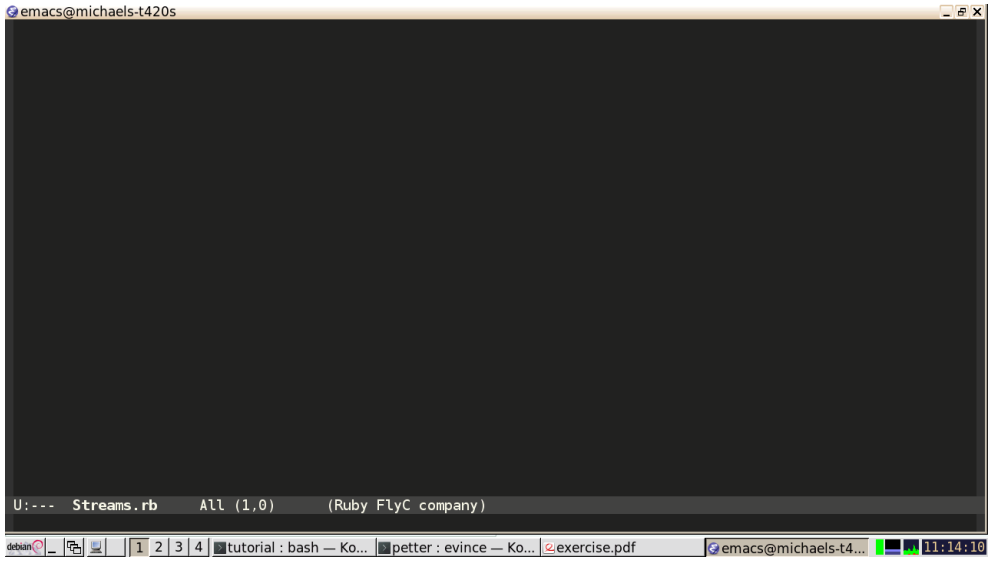
Your task is to come up with a new class *SynchedFileStream* which mixes the class *SyncRW* into the class *FileStream*.

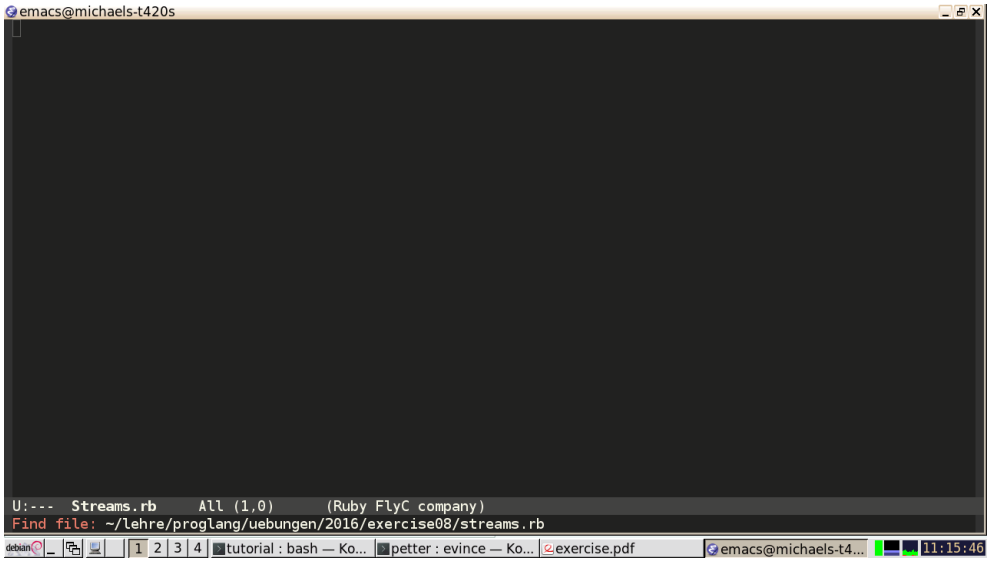
Assignment 8.3 Mixins Ruby

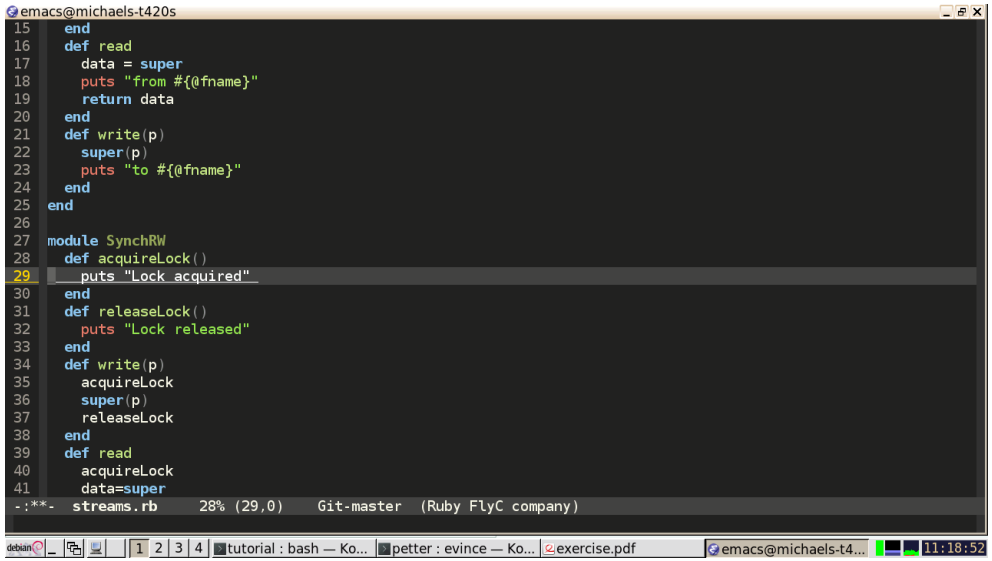
Implement the *Stream Wrapper* scenario from the lecture based on Ruby Mixins

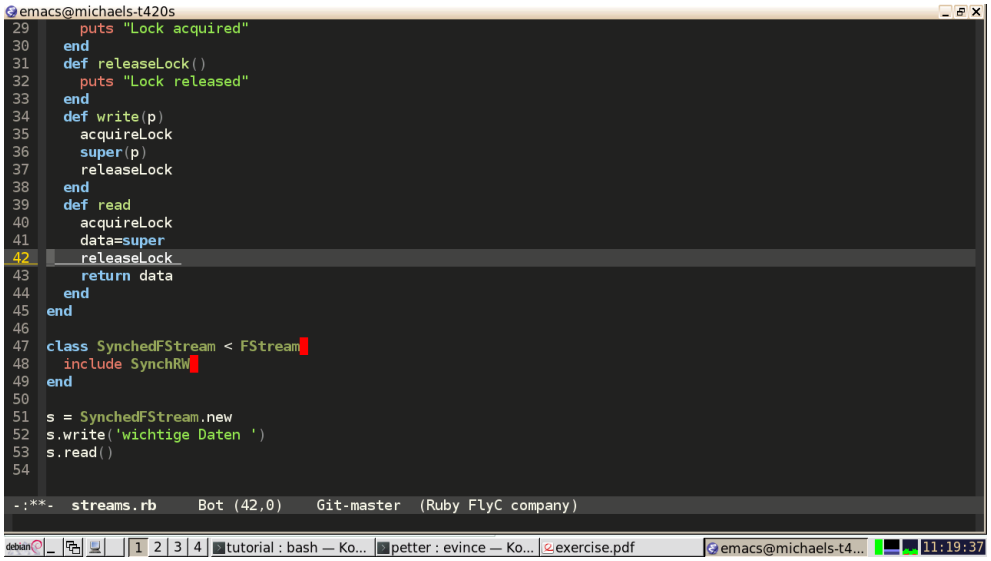
tutorial : bash — Konsole | petter : evince — Konsole | exercise.pdf | 11:12:56

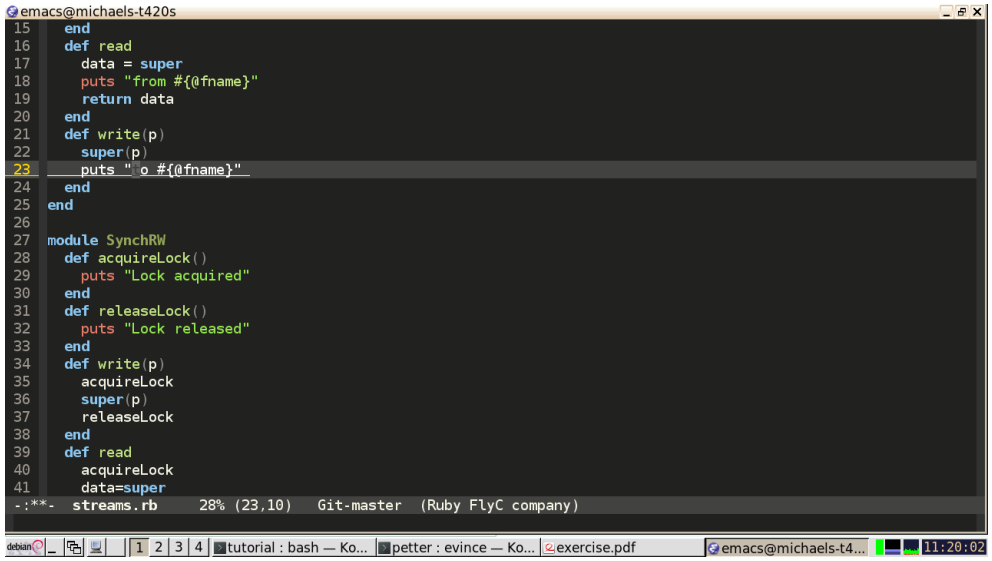

tutorial : bash — Konsole
Datei Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe
petter@michaels-t420s:/home/petter$ cd lehre/tutorial/
petter@michaels-t420s:/home/petter/lehre/tutorial$ ls
bumper bumper.c bumper-fetchandadd.c bumper-semaphore.c dekker dekker.c~ DQueue.c~ sheet04.pdf Streams.java
bumper# bumper.c~ bumper-semaphore bumper-semaphore.c~ dekker.c DQueue.c mjc-1_3_2.jar sheet06.pdf
petter@michaels-t420s:/home/petter/lehre/tutorial$ javac Streams.java
Streams.java:6: error: missing return statement
default int read() { /* ... */ }
^
1 error
petter@michaels-t420s:/home/petter/lehre/tutorial$ emacs Streams.java
Xlib: extension "XInputExtension" missing on display ":1".
petter@michaels-t420s:/home/petter/lehre/tutorial$ javac Streams.java
petter@michaels-t420s:/home/petter/lehre/tutorial$ ls
bumper bumper.c~ bumper-semaphore.c dekker.c DQueue.c~ NetworkStream.class Stream.class Synch.class
bumper# bumper-fetchandadd.c bumper-semaphore.c~ dekker.c~ FileStream.class sheet04.pdf Streams.java
bumper.c bumper-semaphore dekker DQueue.c mjc-1_3_2.jar sheet06.pdf Streams.java~
petter@michaels-t420s:/home/petter/lehre/tutorial$ emacs Streams.java
Xlib: extension "XInputExtension" missing on display ":1".
petter@michaels-t420s:/home/petter/lehre/tutorial$
tutorial : bash
tutorial : bash — Konsole
petter : evince — Konsole
exercise.pdf
11:13:49

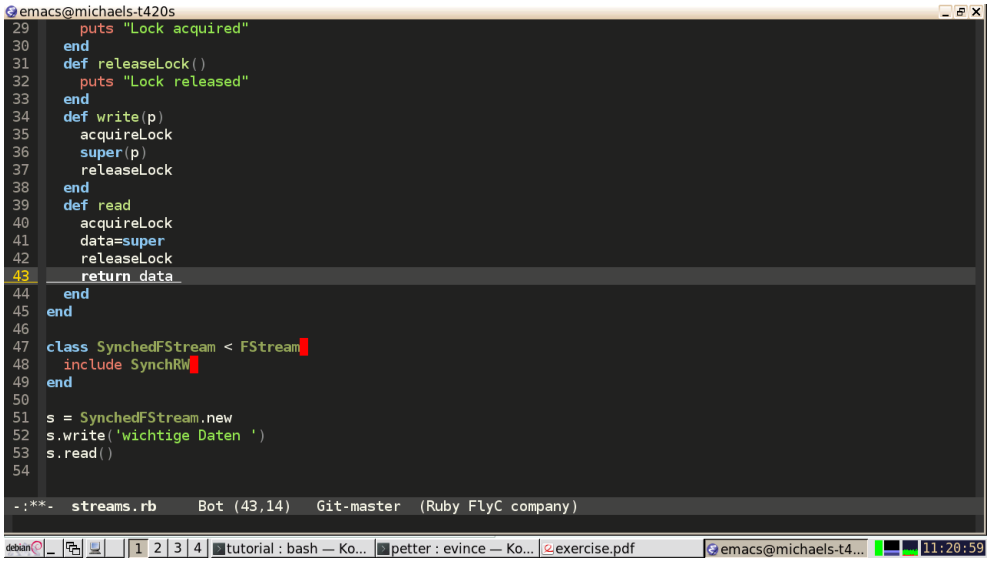

emacs@michaels-t420s
U:--- Streams.rb All (1,0) (Ruby FlyC company)
tutorial : bash — Ko...
petter : evince — Ko...
exercise.pdf
emacs@michaels-t4...
11:14:10

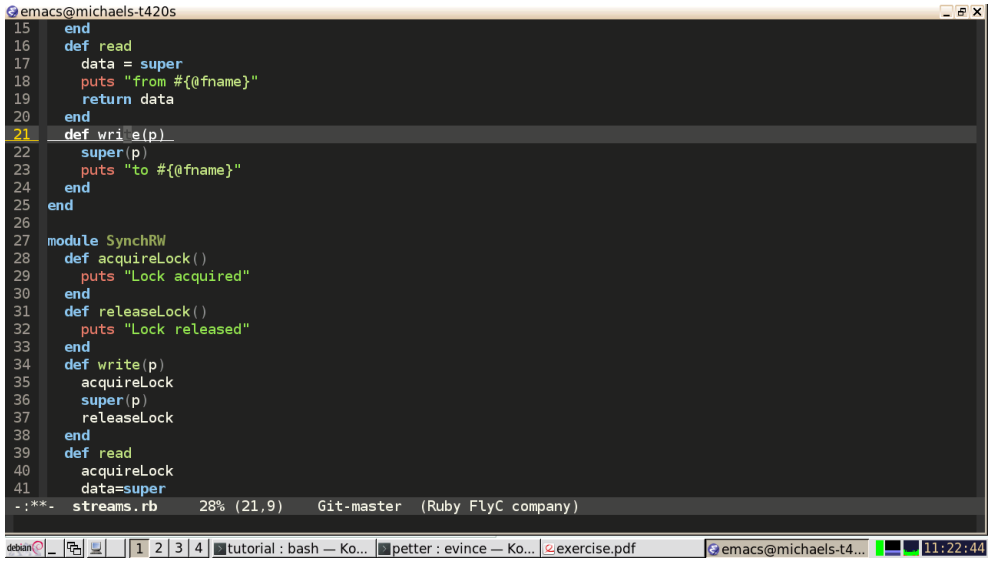

emacs@michaels-t420s
U:--- Streams.rb All (1,0) (Ruby FlyC company)
Find file: ~/lehre/proglang/uebungen/2016/exercise08/streams.rb
tutorial : bash — Ko...
petter : evince — Ko...
exercise.pdf
emacs@michaels-t4...
11:15:46


emacs@michaels-t420s
15 end
16 def read
17 data = super
18 puts "from #{@fname}"
19 return data
20 end
21 def write(p)
22 super(p)
23 puts "to #{@fname}"
24 end
25 end
26
27 module SynchRW
28 def acquireLock()
29 puts "Lock acquired"
30 end
31 def releaseLock()
32 puts "Lock released"
33 end
34 def write(p)
35 acquireLock
36 super(p)
37 releaseLock
38 end
39 def read
40 acquireLock
41 data=super
-:**- streams.rb 28% (29,0) Git-master (Ruby FlyC company)
tutorial : bash — Ko...
petter : evince — Ko...
exercise.pdf
emacs@michaels-t4...
11:18:52


emacs@michaels-t420s
29 puts "Lock acquired"
30 end
31 def releaseLock()
32 puts "Lock released"
33 end
34 def write(p)
35 acquireLock
36 super(p)
37 releaseLock
38 end
39 def read
40 acquireLock
41 data=super
42 releaseLock
43 return data
44 end
45 end
46
47 class SynchedFStream < FStream
48 include SynchRW
49 end
50
51 s = SynchedFStream.new
52 s.write('wichtige Daten ')
53 s.read()
54
-:**- streams.rb Bot (42,0) Git-master (Ruby FlyC company)
1 2 3 4 tutorial : bash — Ko... petter : evince — Ko... exercise.pdf emacs@michaels-t4... 11:19:37


emacs@michaels-t420s
15 end
16 def read
17 data = super
18 puts "from #{@fname}"
19 return data
20 end
21 def write(p)
22 super(p)
23 puts " o #{@fname}"
24 end
25 end
26
27 module SynchRW
28 def acquireLock()
29 puts "Lock acquired"
30 end
31 def releaseLock()
32 puts "Lock released"
33 end
34 def write(p)
35 acquireLock
36 super(p)
37 releaseLock
38 end
39 def read
40 acquireLock
41 data=super
-:**- streams.rb 28% (23,10) Git-master (Ruby FlyC company)
1 2 3 4 tutorial : bash — Ko... petter : evince — Ko... exercise.pdf emacs@michaels-t4... 11:20:02


emacs@michaels-t420s
29 puts "Lock acquired"
30 end
31 def releaseLock()
32 puts "Lock released"
33 end
34 def write(p)
35 acquireLock
36 super(p)
37 releaseLock
38 end
39 def read
40 acquireLock
41 data=super
42 releaseLock
43 return data
44 end
45 end
46
47 class SynchedFStream < FStream
48 include SynchRW
49 end
50
51 s = SynchedFStream.new
52 s.write('wichtige Daten ')
53 s.read()
54
-:**- streams.rb Bot (43,14) Git-master (Ruby FlyC company)
1 2 3 4 tutorial : bash — Ko... petter : evince — Ko... exercise.pdf emacs@michaels-t4... 11:20:59


emacs@michaels-t420s
15 end
16 def read
17 data = super
18 puts "from #{@fname}"
19 return data
20 end
21 def wri e(p)
22 super(p)
23 puts "to #{@fname}"
24 end
25 end
26
27 module SynchRW
28 def acquireLock()
29 puts "Lock acquired"
30 end
31 def releaseLock()
32 puts "Lock released"
33 end
34 def write(p)
35 acquireLock
36 super(p)
37 releaseLock
38 end
39 def read
40 acquireLock
41 data=super
-:**- streams.rb 28% (21,9) Git-master (Ruby FlyC company)
1 2 3 4 tutorial : bash — Ko... petter : evince — Ko... exercise.pdf emacs@michaels-t4... 11:22:44

exercise.pdf

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

2 von 4 152,74%

Reconsider the example from the lecture about synchronized file- and socket-streams. The following classes are given:

$$FileStream = \{read = 0x1, write = 0x2\}$$
$$SocketStream = \{read = 0x3, write = 0x4\}$$
$$SyncRW = \{read = 0x5, write = 0x6\}$$

Your task is to come up with a new class $SynchedFileStream$ which mixes the class $SyncRW$ into the class $FileStream$.

Assignment 8.3 Mixins Ruby

Implement the *Stream Wrapper* scenario from the lecture based on Ruby Mixins

1 2 3 4 tutorial : bash — Ko... petter : evince — Ko... exercise.pdf emacs@michaels-t4... 11:23:53

exercise.pdf

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

3 von 4 127,29%

Assignment 8.4 Implementation differences: Traits vs. Mixins

A next mainstream implementation of traits comes with the Java version 8.

- Implement a solution for the *Stream Wrapper* problem. You may use the following code:

```
interface Stream {
  int read();
}

interface FileStream extends Stream {
  default int read() { /* ... */ }
}

interface NetworkStream extends Stream {
  default int read() { /* ... */ }
}

interface Synch {
  default void acquireLock() { /* ... */ }
  default void releaseLock() { /* ... */ }
}
```

1 2 3 4 tutorial : bash — Ko... petter : evince — Ko... exercise.pdf emacs@michaels-t4... 11:24:15

exercise.pdf

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

3 von 4 127,29%

Assignment 8.4 Implementation differences: Traits vs. Mixins

A next mainstream implementation of traits comes with the Java version 8.

- Implement a solution for the *Stream Wrapper* problem. You may use the following code:

```
interface Stream {
  int read();
}

interface FileStream extends Stream {
  default int read() { /* ... */ }
}

interface NetworkStream extends Stream {
  default int read() { /* ... */ }
}

interface Synch {
  default void acquireLock() { /* ... */ }
  default void releaseLock() { /* ... */ }
}
```

- Compare your solution to the one based on Mixins from the last exercise sheet. What are the differences? Which one is more flexible w.r.t. software engineering

1 2 3 4 tutorial : bash — Ko... petter : evince — Ko... exercise.pdf emacs@michaels-t4... 11:24:55

exercise.pdf

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

3 von 4 127,29%

Assignment 8.4 Implementation differences: Traits vs. Mixins

A next mainstream implementation of traits comes with the Java version 8.

- Implement a solution for the *Stream Wrapper* problem. You may use the following code:

```
interface Stream {
  int read();
}

interface FileStream extends Stream {
  default int read() { /* ... */ }
}

interface NetworkStream extends Stream {
  default int read() { /* ... */ }
}

interface Synch {
  default void acquireLock() { /* ... */ }
  default void releaseLock() { /* ... */ }
}
```

- Compare your solution to the one based on Mixins from the last exercise sheet. What are the differences? Which one is more flexible w.r.t. software engineering

1 2 3 4 tutorial : bash — Ko... petter : evince — Ko... exercise.pdf emacs@michaels-t4... 11:26:18

tutorial : bash — Konsole

Datei Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe

```
petter@michaels-t420s:/home/petter$ cd lehre/tutorial/
petter@michaels-t420s:/home/petter/lehre/tutorial$ ls
bumper   bumper.c   bumper-fetchandadd.c  bumper-semaphore.c   dekker    dekker.c~  DQueue.c~      sheet04.pdf  Streams.java
bumper#  bumper.c~  bumper-semaphore      bumper-semaphore.c~  dekker.c  DQueue.c   mjc-1_3_2.jar  sheet06.pdf
petter@michaels-t420s:/home/petter/lehre/tutorial$ javac Streams.java
Streams.java:6: error: missing return statement
  default int read() { /* ... */ }
                     ^
1 error
petter@michaels-t420s:/home/petter/lehre/tutorial$ emacs Streams.java
Xlib:  extension "XInputExtension" missing on display ":1".
petter@michaels-t420s:/home/petter/lehre/tutorial$ javac Streams.java
petter@michaels-t420s:/home/petter/lehre/tutorial$ ls
bumper   bumper.c~            bumper-semaphore.c   dekker.c   DQueue.c~         NetworkStream.class  Stream.class   Synch.class
bumper#  bumper-fetchandadd.c  bumper-semaphore.c~  dekker.c~  FileStream.class  sheet04.pdf          Streams.java
bumper.c  bumper-semaphore     dekker               DQueue.c   mjc-1_3_2.jar     sheet06.pdf          Streams.java~
petter@michaels-t420s:/home/petter/lehre/tutorial$ emacs Streams.java
Xlib:  extension "XInputExtension" missing on display ":1".
petter@michaels-t420s:/home/petter/lehre/tutorial$ emacs Streams.rb &
[1] 2942
petter@michaels-t420s:/home/petter/lehre/tutorial$ Xlib:  extension "XInputExtension" missing on display ":1".
```

tutorial : bash

debian 1 2 3 4 tutorial : bash — Ko... petter : evince — Ko... exercise.pdf emacs@michaels-t4... 11:26:36

emacs@michaels-t420s

```
17
18 class SynchFileStream implements FileStream,Synch {
19     public int read(){
20         acquireLock();
21         int read = FileStream.super.read();
22         releaseLock();
23         return read;
24     }
25 }
26
27 interface SynchWrapper extends Synch {
28 default int read(){
29     acquireLock();
30     re
31 }}
```

-:**- Streams.java   Bot (30,6)   (Java/l FlyC- company Abbrev)

debian 1 2 3 4 tutorial : bash — Ko... petter : evince — Ko... exercise.pdf emacs@michaels-t4... 11:34:19

exercise.pdf

Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

3 von 4 127,29%

Assignment 8.4 Implementation differences: Traits vs. Mixins

A next mainstream implementation of traits comes with the Java version 8.

- Implement a solution for the *Stream Wrapper* problem. You may use the following code:

```
interface Stream {
  int read();
}

interface FileStream extends Stream {
  default int read() { /* ... */ }
}

interface NetworkStream extends Stream {
  default int read() { /* ... */ }
}

interface Synch {
  default void acquireLock() { /* ... */ }
  default void releaseLock() { /* ... */ }
}
```

- Compare your solution to the one based on Mixins from the last exercise sheet. What are the differences? Which one is more flexible w.r.t. software engineering

debian 1 2 3 4 tutorial : bash — Ko... petter : evince — Ko... exercise.pdf emacs@michaels-t4... 11:39:12

exercise.pdf

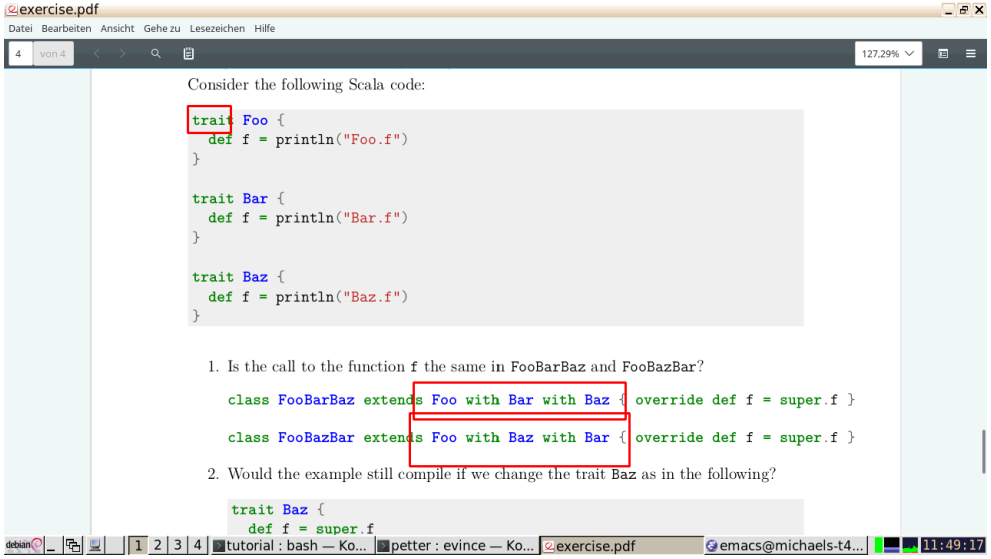
Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe

3 von 4 127,29%

```
    public abstract void f();
}
interface K {
  default void g() { System.out.println("G"); };
}
interface I extends K {
  default void f() { System.out.println("I"); };
}
interface J extends K {
  default void f() { System.out.println("J"); };
}
public class C extends A implements I,J {
}
```

1. How many errors are in this code? Elaborate extensively on the causes!
2. Fix the code, while conserving the given inheritance relation, so that the whole program compiles!

debian 1 2 3 4 tutorial : bash — Ko... petter : evince — Ko... exercise.pdf emacs@michaels-t4... 11:48:49


exercise.pdf
Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe
4 von 4
127.29%
Consider the following Scala code:
trait Foo {
def f = println("Foo.f")
}
trait Bar {
def f = println("Bar.f")
}
trait Baz {
def f = println("Baz.f")
}
1. Is the call to the function f the same in FooBarBaz and FooBazBar?
class FooBarBaz extends Foo with Bar with Baz { override def f = super.f }
class FooBazBar extends Foo with Baz with Bar { override def f = super.f }
2. Would the example still compile if we change the trait Baz as in the following?
trait Baz {
def f = super.f
tutorial : bash — Ko...
petter : evince — Ko...
exercise.pdf
emacs@michaels-t4...
11:49:17