

valid, 64 possible configurations

- ▼ ☒ Root
 - ☒ Base
 - ☐ ColoredMessages
 - ☐ Authentication
 - ☐ ChatLog
 - ☐ SoundOutput
 - > ☐ Encryption
 - ▼ ☒ UserInterface
 - ☐ GUI
 - ☒ CommandLine

Terminal - javac /home/bintzandt/Uni/SoftwareProductLines/ChatHouse/src

bintzandt@Trinity ~/U/S/C/src (master)> javac -d out **/*.java && java -cp out Client localhost 8080

Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Logging to /home/bintzandt/spl_client.log
Connected to the chat server

Reader started

Writer started

[Server]: Welcome anonymous, no other users are currently connected

[anonymous]: hi

[anonymous]: I am anonymous

[anonymous]:

Terminal - javac /home/bintzandt/Uni/SoftwareProductLines/ChatHouse/src

bintzandt@Trinity ~/U/S/C/src (master)> javac -d out **/*.java && java -cp out Server 8080

Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Logging to /home/bintzandt/spl_server.log
Server is listening on port 8080
New user connected!

a) Evolve your chat-product line towards feature-oriented programming.*

Hints:

- I recommend using FeatureIDE with FeatureHouse, which on my system works without problems. See <https://github.com/FeatureIDE/FeatureIDE/wiki/Tutorial>. Using AHEAD instead of FeatureHouse I encounter a bug, even on the simple program from the tutorial.
- Start with the base program and extend it with the features from the previous assignments.

* Modified version for smaller groups:
3-person groups may omit support for logging.
2-person groups may omit support for logging and authentication.

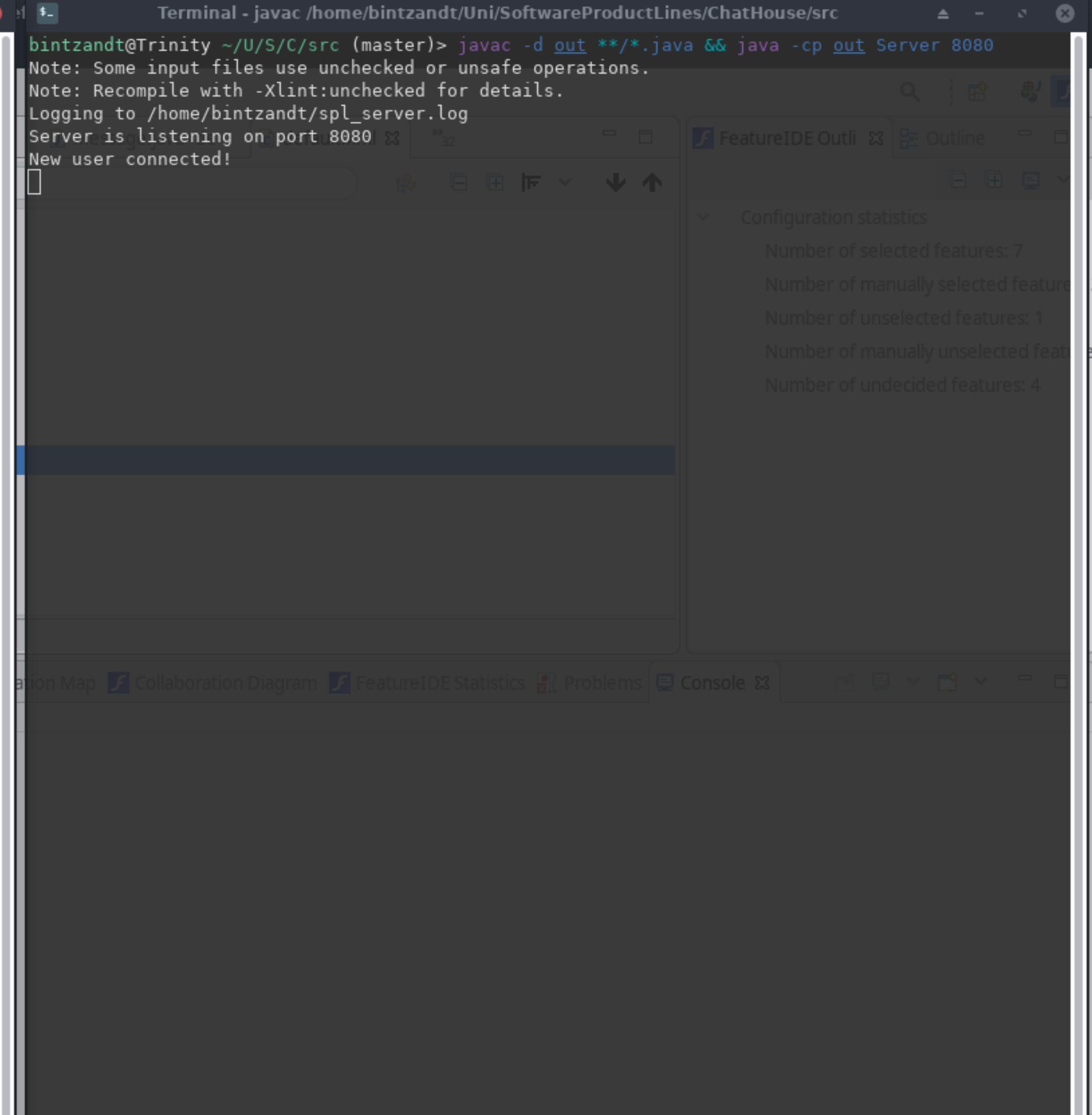
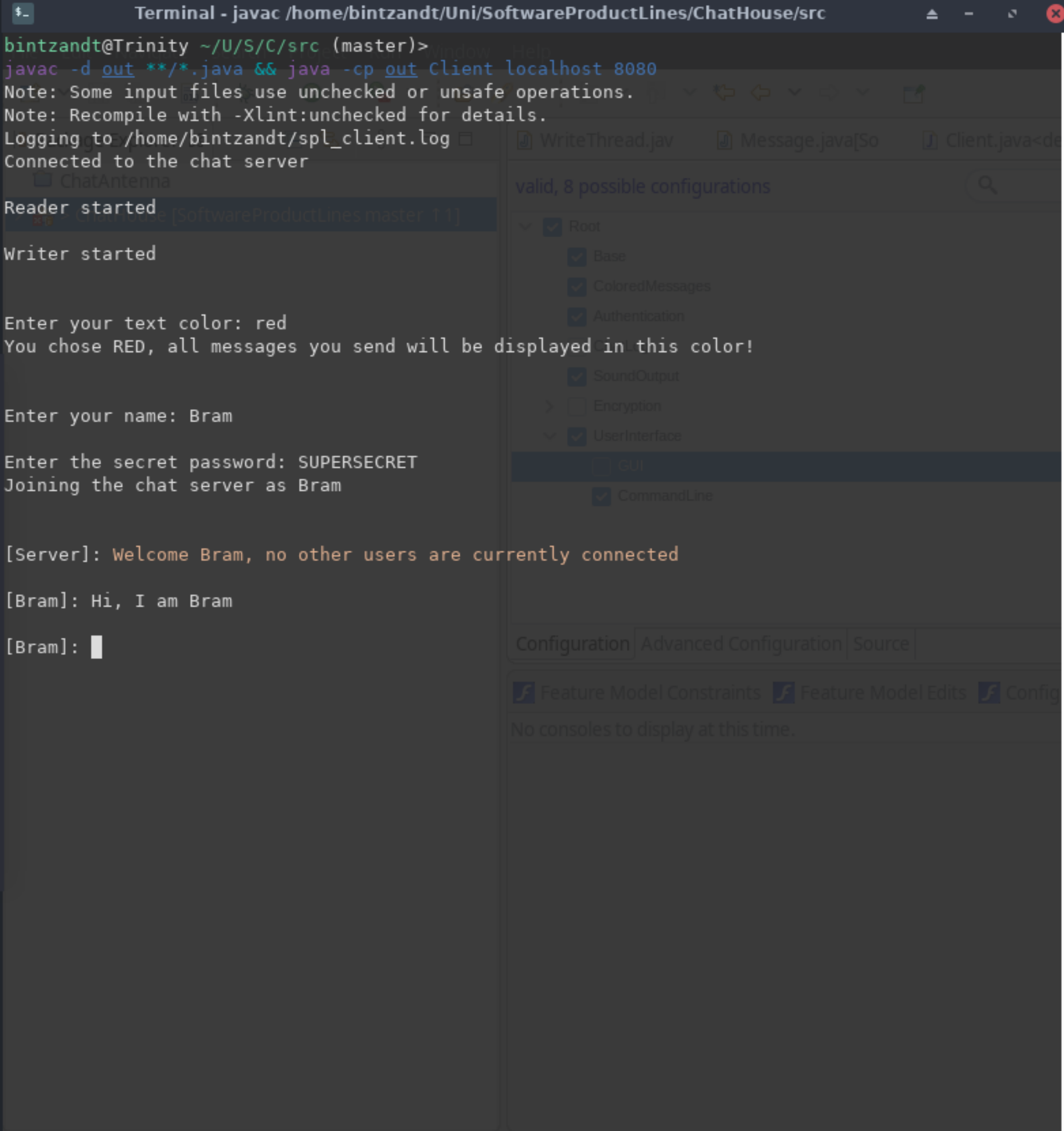
b) After finishing task (a), add a new feature: sound output. This feature plays a sound on every received and sent message in the client. A simple solution could just play beep sound, using `Toolkit.getDefaultToolkit().beep();`

c) Generate and manually test a few (at least three) variants.

Submit a zip archive with the source code of your implementation, and a test report PDF. The test report should include for each tested variant: a) the used feature selection, and b) a screenshot of the resulting system when executed.

valid, 8 possible configurations

- ▼ ☒ Root
 - ☒ Base
 - ☒ ColoredMessages
 - ☒ Authentication
 - ☐ ChatLog
 - ☒ SoundOutput
 - > ☐ Encryption
 - ▼ ☒ UserInterface
 - ☐ GUI
 - ☒ CommandLine



valid, 2 possible configurations

- ▼ ☒ Root
 - ☒ Base
 - ☐ ColoredMessages
 - ☒ Authentication
 - ☐ ChatLog
 - ☒ SoundOutput
 - ▼ ☒ Encryption
 - ☒ ROT13
 - ☒ Reverse
 - ▼ ☒ UserInterface
 - ☒ GUI
 - ☐ CommandLine

```
bintzandt@Trinity ~/U/S/C/src (master)> javac -d out **/*.java && java -cp out Client localhost 8080
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Logging to /home/bintzandt/spl_client.log
```

Chat Frame

```
Connected to the chat server
Reader started
Writer started
Enter your name: Bram
Enter the secret password: SUPERSECRET
Joining the chat server as Bram
[Server]: Welcome Bram, no other users are currently connected
[Bram]: Hoi
```

[Bram]:

```
bintzandt@Trinity ~/U/S/C/src (master)> javac -d out **/*.java && java -cp out Server 8080
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Logging to /home/bintzandt/spl_server.log
Server is listening on port 8080
New user connected!
```

FeatureIDE Outli



Configuration statistics

Number of selected features: 9
Number of manually selected feature
Number of unselected features: 2
Number of manually unselected fea
Number of undecided features: 1

No consoles to display at this time.