

University of Tartu
Institute of Computer Science


Jüri Kiho

***Amadeus_AlgJava:
a Java Programming Environment for Beginners***

Jüri Lutsius

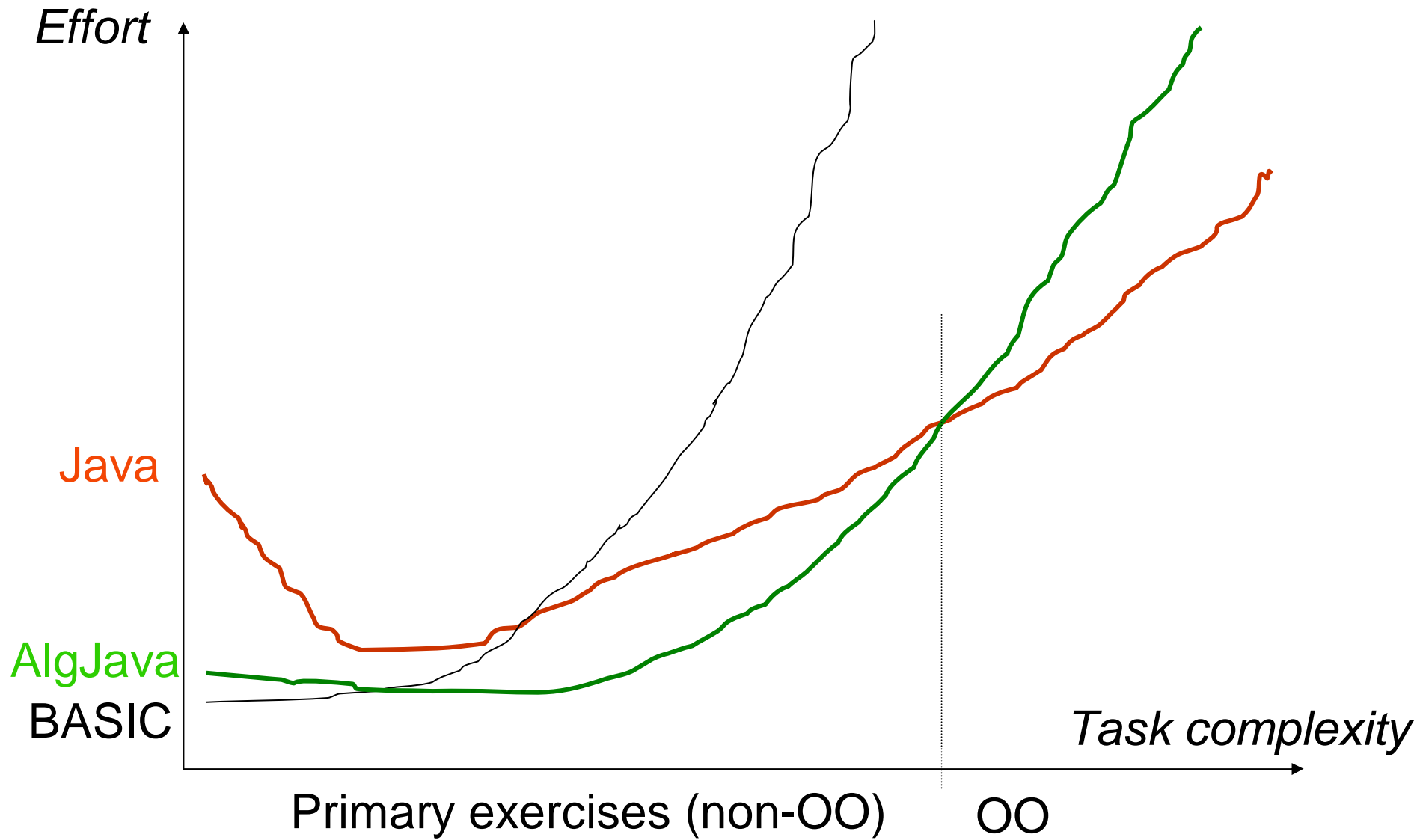
Converter Generation by Amadeus

Second Lübeck-Tartu Workshop
on Informatics September 25-27,
2006

- Amadeus_AlgJava
 - **Amadeus**
 - Converter Generation
- 
- ```
graph LR; A[Amadeus] -.-> B[Amadeus_AlgJava]; A -.-> C[Converter Generation];
```

Which programming language should be used to teach beginners ? [In UT: 392 students in fall semester 2006]

|         | Start | Perspective |
|---------|-------|-------------|
| BASIC   | Easy  | Poor        |
| Python  | Easy  | ?           |
| Java    | Hard  | Good        |
| AlgJava | Easy  | Good        |



# AlgJava

(*Alg-* in Estonian – *Basic, Primary*)

**Algorithmic Java:** methods – yes, own objects – no  
Simplicity. Closeness to Java

## Amadeus\_AlgJava \*

Programming environment for AlgJava

Semigraphical program representation

<http://www.cs.ut.ee/~kiho/AlgJavaHome.html>

\* Developed in year 2006 at the University of Tartu

Contributors: Helle Hein, Marina Issakova, Vambola Leping, Reimo Palm, Ahti Peder

Supported by Estonian Information Technology Foundation / Tiger University

Project duration: June 1 – November 30, 2006

Second Lübeck-Tartu Workshop  
on Informatics September 25-27,  
2006

**Patrick Jordan.**

*A Very Quick Comparison of Popular Languages  
for Teaching Computer Programming.*

<http://www.ariel.com.au/a/teaching-programming.html> (25.09.06)

BASIC, C, **Java and Python** + AlgJava

Task of reading two numbers from the user,  
adding them together and printing out the result

- How long did it take to write and debug the code
- How many things does a student need to understand in order to write this code

## Java

```
import java.io.*;
 public class Addup {
 static public void main(String args[]) {
 InputStreamReader stdin = new InputStreamReader(System.in);
 BufferedReader console = new BufferedReader(stdin);
 int i1 = 0, i2 = 0;
 String s1,s2;
 try { s1 = console.readLine();
 i1 = Integer.parseInt(s1);
 s2 = console.readLine();
 i2 = Integer.parseInt(s2); }
 catch(IOException ioex) {
 System.out.println("Input error"); System.exit(1);
 }
 catch(NumberFormatException nfex) {
 System.out.println("\"" + nfex.getMessage() +
 "\" is not numeric");
 System.exit(1); }
 System.out.println(i1 + " + " + i2 + " = " + (i1+i2));
 System.exit(0); } }
```

```
%> javac Addup.java
```

```
%> java Addup
```

**Time to write:**19 minutes!

```
import java.io.*;
public class Addup {
 static public void main(String args[]) {
 InputStreamReader stdin = new InputStreamReader(System.in);
 BufferedReader console = new BufferedReader(stdin);
 int a = 0, b = 0;
 try {
 a = Integer.parseInt(console.readLine());
 b = Integer.parseInt(console.readLine()); }
 catch(IOException e) { }
 System.out.println(a + " + " + b + " = " + (a+b));
 }
}
```



## Python

```
import sys
a = sys.stdin.readline()
b = sys.stdin.readline()
c = int(a) + int(b)
print c
```

*%> python add.py*

**Time to write:**  
about one minute

## AlgJava

```
>, readln
int a = Integer.parseInt(readln());
int b = Integer.parseInt(readln());
> inspect :a :b :a + b
```

*Ctrl+s*

## Java

```
import java.io.*;
public class Addup {
 static public void main(String args[]) {
 InputStreamReader stdin = new InputStreamReader(System.in);
 BufferedReader console = new BufferedReader(stdin);
 int a = 0, b = 0;
 try {
 a = Integer.parseInt(console.readLine());
 b = Integer.parseInt(console.readLine());
 } catch(IOException e) { }
 System.out.println(a + " + " + b + " = " + (a+b));
 }
}
```

## Python

```
import sys
a = sys.stdin.readline()
b = sys.stdin.readline()
c = int(a) + int(b)
print c
```

*%> python add.py*

**Time to write:**  
about one minute

## AlgJava

**readln**

```
int a = Integer.parseInt(readln());
int b = Integer.parseInt(readln());
println(a + " + " + b + " = " + (a+b));
```

*Ctrl+s*

## Java

```
import java.io.*;
public class Addup {
 static public void main(String args[]) {
 InputStreamReader stdin = new InputStreamReader(System.in);
 BufferedReader console = new BufferedReader(stdin);
 int a = 0, b = 0;
 try {
 a = Integer.parseInt(console.readLine());
 b = Integer.parseInt(console.readLine());
 } catch(IOException e) { }
 System.out.println(a + " + " + b + " = " + (a+b));
 }
}
```

# Things to explain

## Java

import, classes, semicolons, braces  
public, static, void, String, main, args[ ]  
InputStreamReader, BufferedReader, System.in  
variables, types  
try, catch, exceptions, readLine, parseInt  
System.out.println, compiling, running

## Python

import .....  
variables .....  
sys.stdin .....  
readline ..... (reads a string) .....  
int ..... (converts a string to an integer) .....  
print .....

## AlgJava

plug-ins  
variables  
readln  
Integer.parseInt  
inspect (or println)

# **Amadeus\_AlgJava: demo**

Second Lübeck-Tartu Workshop  
on Informatics September 25-27,  
2006

# AlgJava/Amadeus\_AlgJava

## Benefits

Very quick start  
Mother language (EST RUS ENG)  
All (non OO) Java elements  
Semigraphic representation  
Structure support  
Demos  
Macros  
Immediate screen graphics  
Easy text files  
Method readln()  
Counter-loop \*  $i = a .. b$   
Break points and inspection  
Clear project definition  
Minimal user interface

## Possible drawbacks

Extra effort to learn editing  
Reluctant return to plain texts  
  
Missing in Java  
Missing in Java  
Missing in Java  
Missing in Java  
Uncommon

# AMADEUS

Second Lübeck-Tartu Workshop  
on Informatics September 25-27,  
2006

# Skm-text structure

-Start of sketch

[**SKETCH HEAD** -- primitive]

-Start of branch (1)

[**BRANCH HEAD** -- primitive]

**MEMBER** -- primitive or sub-sketch

**MEMBER** -- primitive or sub-sketch

...

-Start of branch (2)

[**BRANCH HEAD** -- primitive]

**MEMBER** -- primitive or sub-sketch

**MEMBER** -- primitive or sub-sketch

...

-End of sketch

Primitive:

**plain text row**

**plain text row**

...

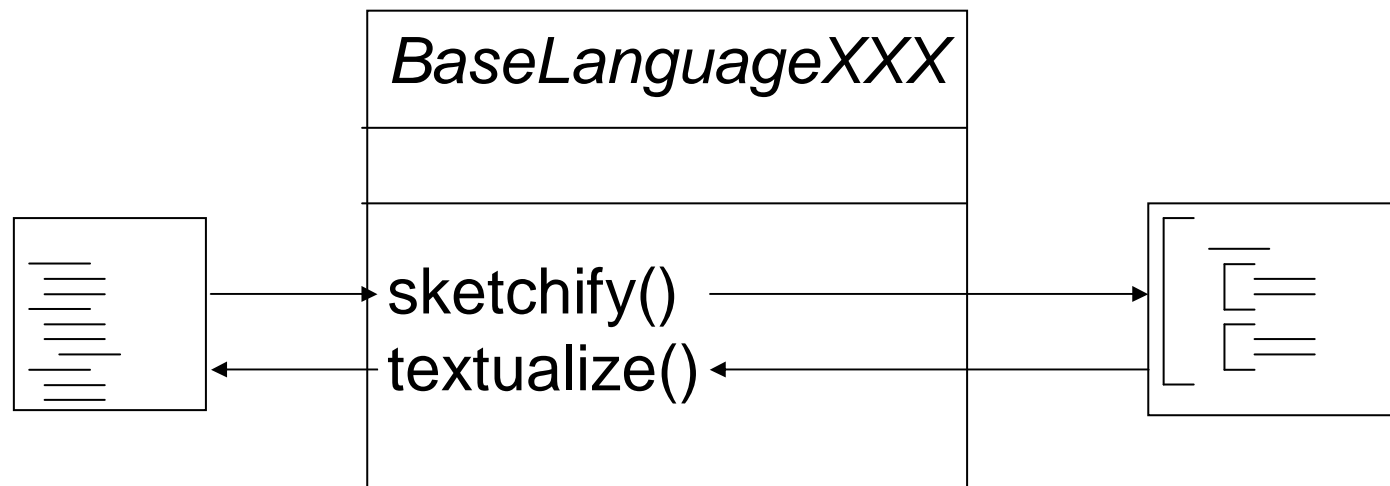
**plain text row comment**

# **Skm-text editing: demo**

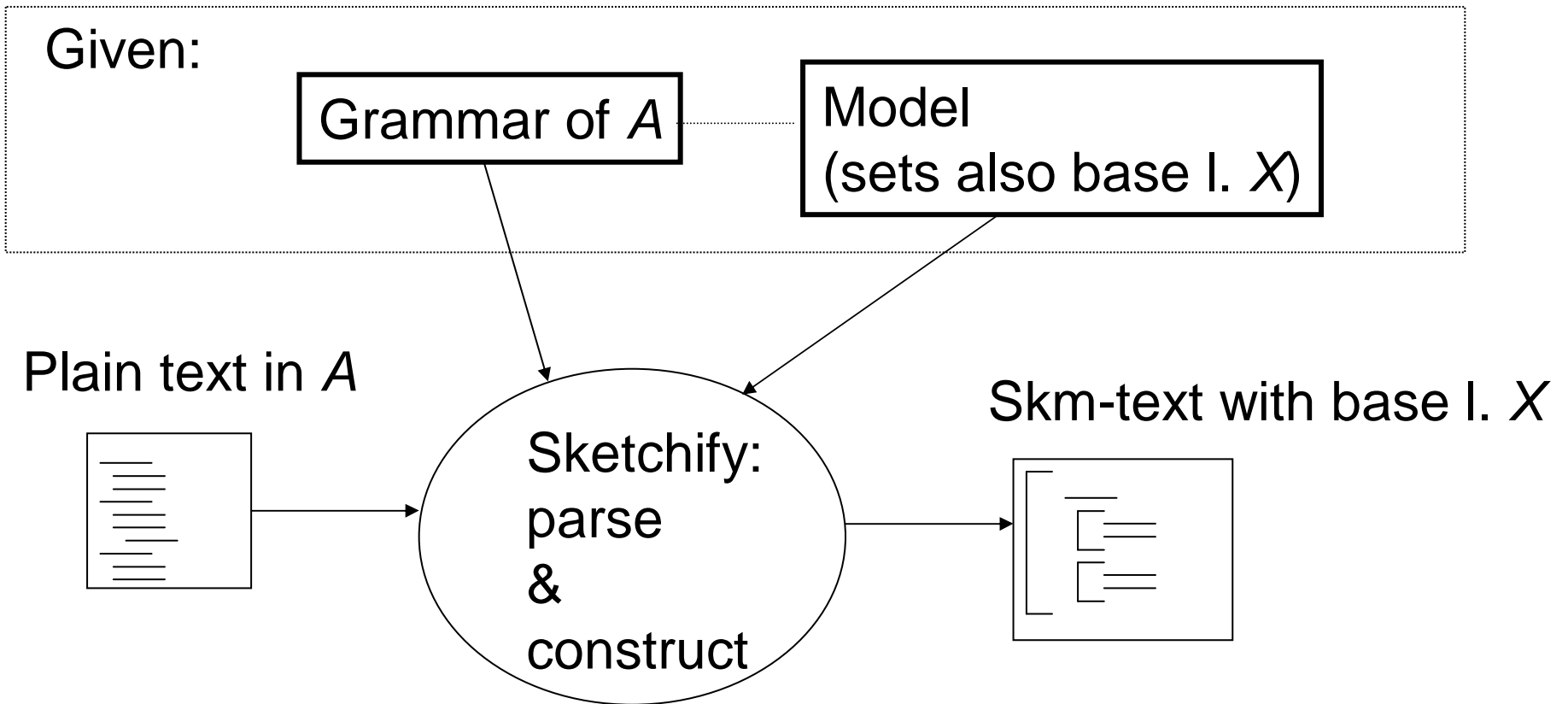
Second Lübeck-Tartu Workshop  
on Informatics September 25-27,  
2006



## Base language of skm-text



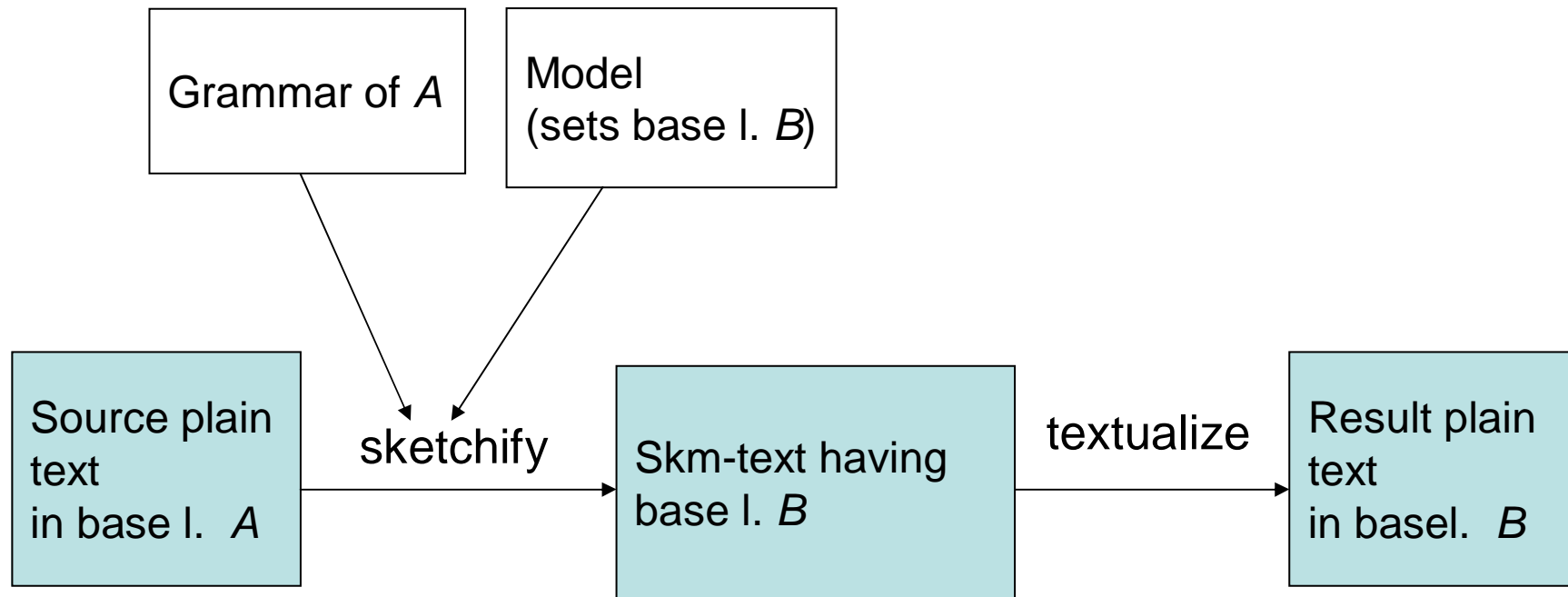
# Grammar-based sketchifying



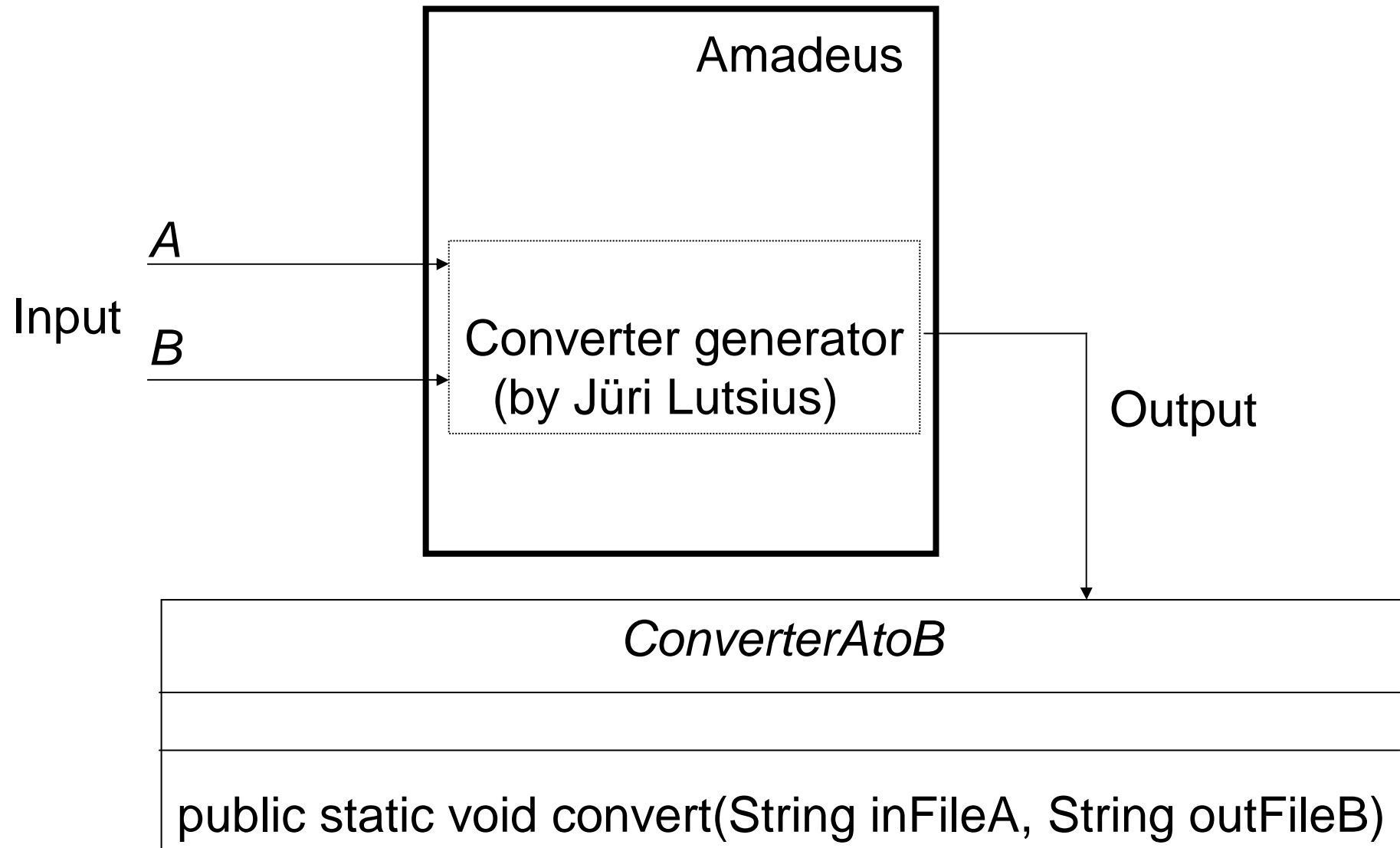
# Sketchyfyng: demo

Second Lübeck-Tartu Workshop  
on Informatics September 25-27,  
2006

# Conversion



# Converter generation



## References

J.Kiho. SKM. Sketchy Modeling of Computer Texts.

RR UT ICS, Tartu, 2000, 64 p.

J.Kiho. Sketchy Modeling for XML.

7th Symposium on Programming Languages and Software Tools. Szeged, Hungary, June 15-16, 2001, pp. 183-197

J.Kiho, S.Solopova. Heterogeneous File Projects

in the Sketchy Modeling Environment.

The Tenth Nordic Workshop on Programming and Software Development Tools and Techniques. IT-University, Copenhagen, August 18-20, 2002, pp. 71-79

N. Tomassova, J. Kiho. Grammar-Based Visual Software Converter Development.

The 9th Symposium on Programming Languages and Software Tools SPLST'2005. Tartu, Estonia, August 13-14, 2005, pp. 206-219