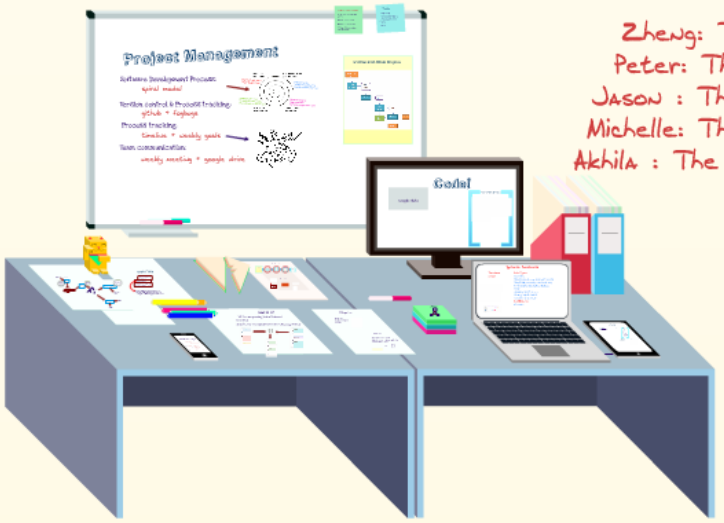
Your Life After TT!

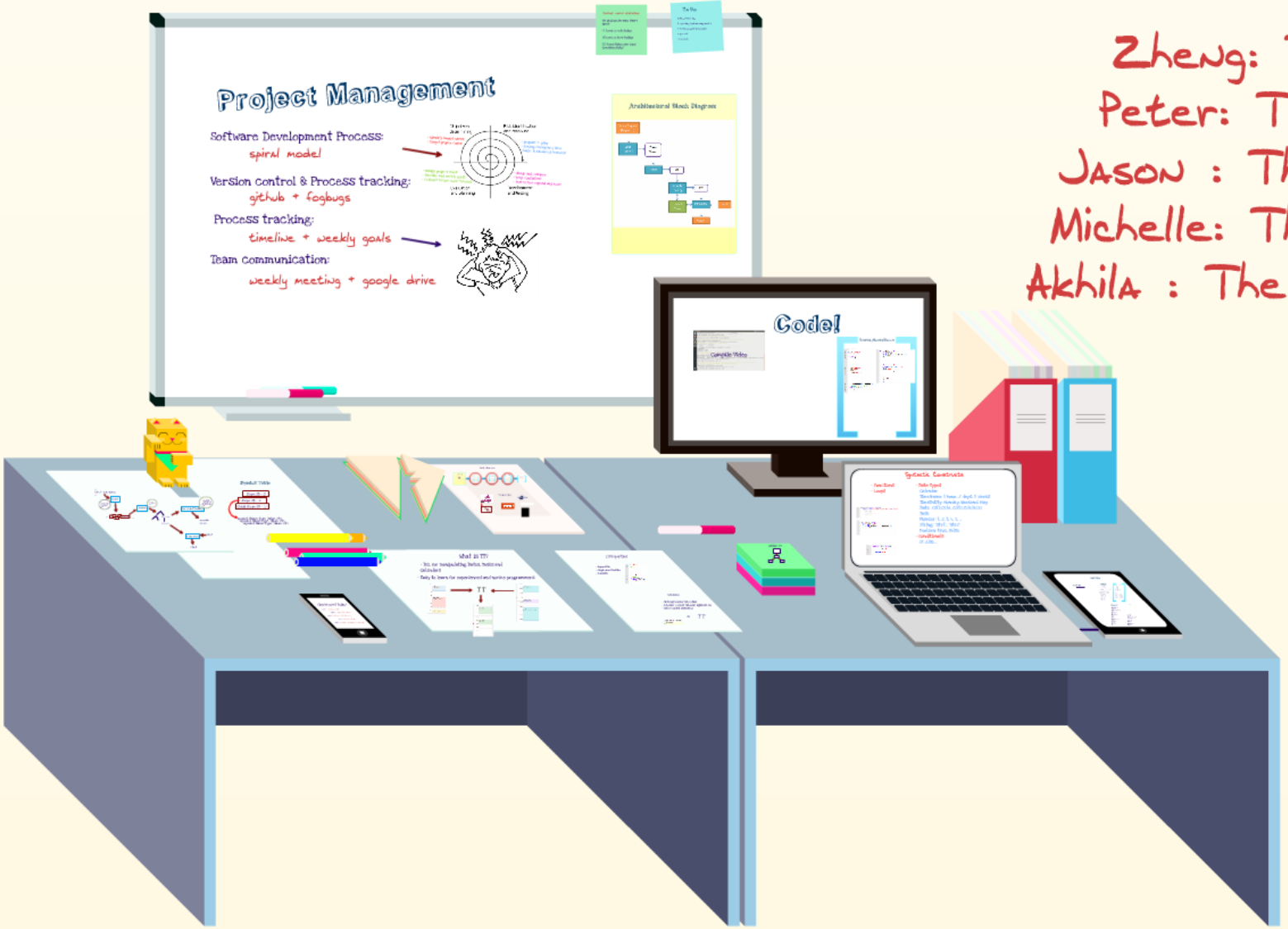
Time & Task

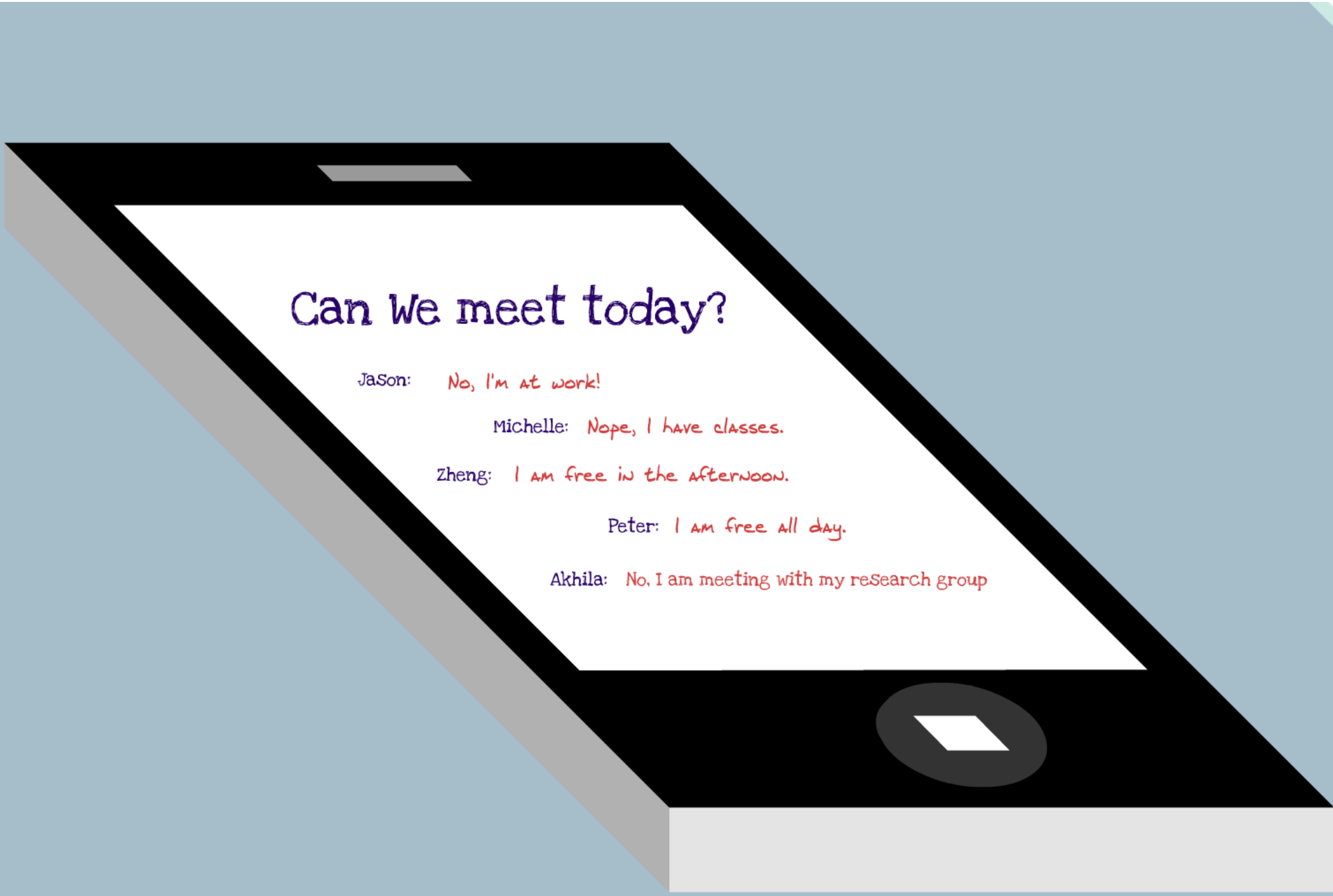
Zheng: The PM
Peter: The Guru
Jason : The Architect
Michelle: The Tester
Akhila : The Integrator



Time & Task

Zheng: The PM
Peter: The Guru
Jason : The Architect
Michelle: The Tester
Akhila : The Integrator

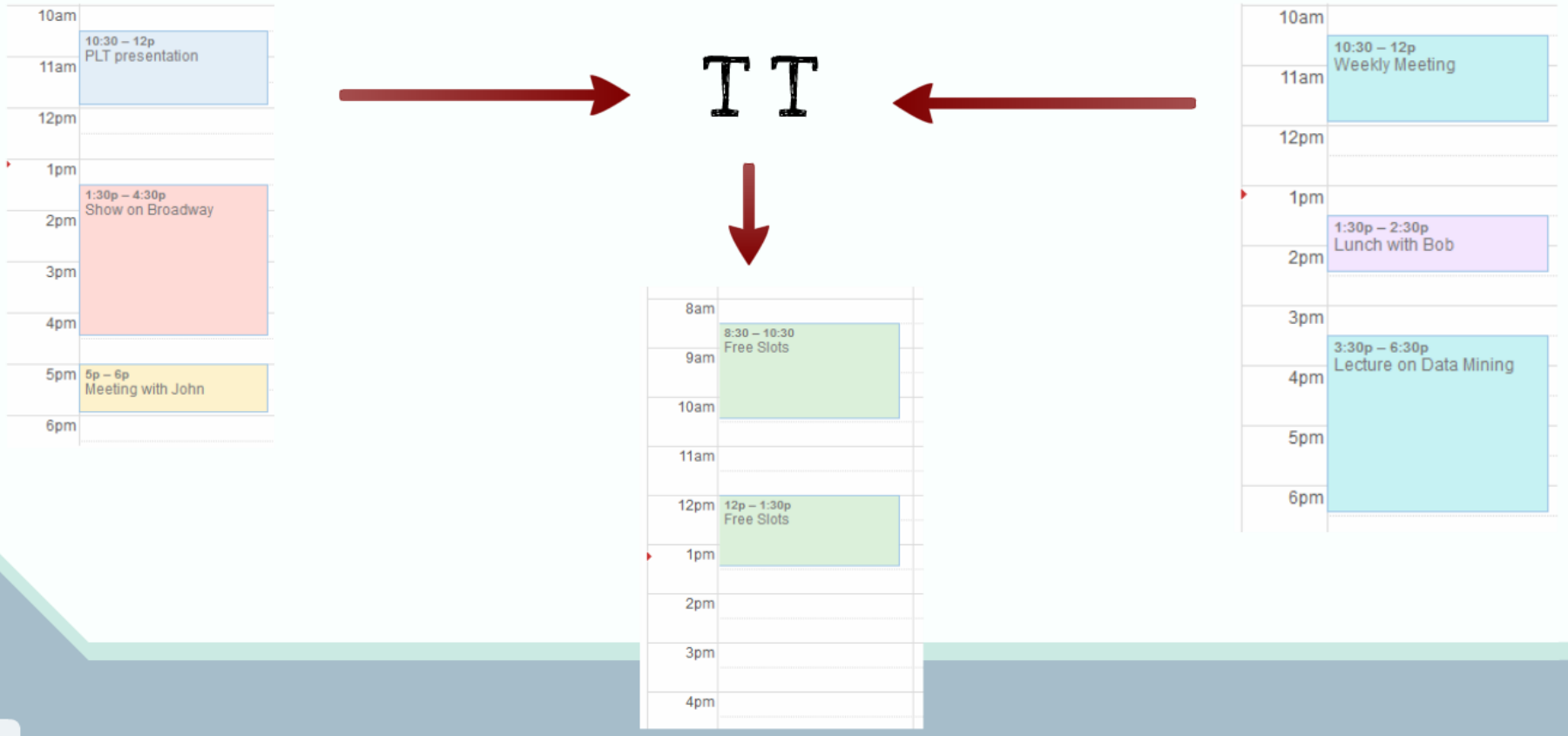



Can We meet today?
Jason: No, I'm at work!
Michelle: Nope, I have classes.
Zheng: I am free in the afternoon.
Peter: I am free all day.
Akhila: No, I am meeting with my research group

What is TT?

- DSL for manipulating Dates, Tasks and Calendars

- Easy to learn for experienced and novice programmers



TT Properties

- Imperative
- Simple and Familiar
- Readable

```
1   import "<std>";
2
3   //FAMILIAR
4   main() {
5       //SIMPLE
6       Date a = 2013.05.16;
7       Date b = a + 1 day;
8       //output 2013.05.17
9       print(b);
10
11      //READABLE
12      Calendar myCalendar;
13      every Task t in myCalendar {
14          print(t);
15      }
16  }
```

Motivation

- Java deprecated Date class
- Relevant context Calendar applications
- Gather useful statistics

vs.

```
java.util.Date d = new java.util.Date();
int year = d.getYear();
```

int java.util.Date.getYear()

@Deprecated

Collect useful statistics

The windows dev team @MSFT spent :

40 hours on code design

55 hours on dev & testing

100 hours fixing user bugs (Something fishy)

To Do

1. fix grammar bug
2. operating Systems assignment :(
3. Meet my project team mates
4. pay rent
5. call mom

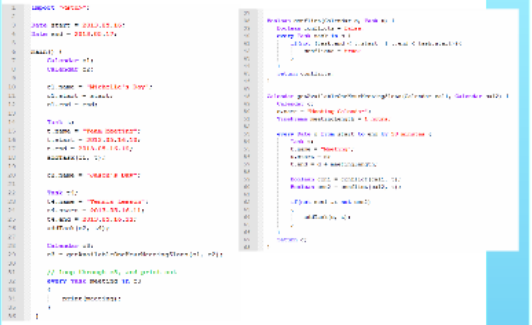
Tutorial_MeetingTime.tt

```
import "<std>";

Date start = 2013.05.16;
Date end = 2013.05.17;

main() {
    Calendar c1;
    Calendar c2;

    c1.name = "Michelle's Day";
    c1.start = start;
    c1.end = end;

    Task t;
    t.name = "Team Meeting";
    t.start = 2013.05.16.10;
    t.end = 2013.05.16.12;
    addTask(c1, t);

    c2.name = "Jason's Day";

    Task t4;
    t4.name = "Tennis Lesson";
    t4.start = 2013.05.16.11;
    t4.end = 2013.05.16.12;
    addTask(c2, t4);

    Calendar c3;
    c3 = getAvailableOneHourMeetingSlots(c1, c2);

    // loop through c3, and print out
    every Task meeting in c3
    {
        print(meeting);
    }
}
```

```

Boolean conflict(Calendar c, Task t) {
   Boolean conflicts = false;
   every Task task in c {
       if(not (task.end < t.start || t.end < task.start)){
           conflicts = true;
       }
   }
   return conflicts;
}

Calendar getAvailableOneHourMeetingSlots(Calendar cal1, Calendar cal2) {
   Calendar c;
   c.name = "Meeting Calendar";
   TimeFrame meetingLength = 1 hours;

   every Date d from start to end by 30 minutes {
       Task t;
       t.name = "Meeting";
       t.start = d;
       t.end = d + meetingLength;

       Boolean con1 = conflict(cal1, t);
       Boolean con2 = conflict(cal2, t);

       if(not con1 && not con2)
       {
           addTask(c, t);
       }
   }
   return c;
}
```

Code!

Tutorial_MeetingTime.tt



Syntactic Co

- Functions
- Loops

- Data-Types
Calendar
TimeFrame

Syntactic Constructs

- Functions
- Loops

```
populateTasks( Calendar c, String name) {
    every Date d from 2013.05.16.12.00 to 2013.05.16.17.00 by 30 minutes {
        Task t;
        t.name = name;
        t.start = d;
        t.end = d + 30 minutes;
        addTask(c, t);
    }
}
```

```
Boolean conflict( Calendar c, Task t) {
    Boolean conflicts = false;
    every Task task in c{
        if ( not (task.end < t.start || t.end < task.start)) {
            conflicts = true;
            break;
        }
    }
    return conflicts;
}
```

```
15  Boolean isMay(Date d){
16      if (is(d, May))
17          return true;
18      return false;
19  }
```

- Data-Types
  - Calendar
  - TimeFrame: 1 hour, 2 days, 3 weeks
  - TimeEntity: Monday, Weekend, May
  - Date: 2013.05.16, 2013.05.16.16.00
  - Task
  - Number: 1, 2, 3, 4, 5, ...
  - String: "str1", "str2"
  - Boolean: true, false
- Conditionals
  - if...else...

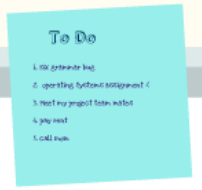
- Functions
- Loops

```
15  populateTasks( Calendar c, String name) {
16      every Date d from 2013.05.16.12.00 to 2013.05.16.17.00 by 30 minutes {
17          Task t;
18          t.name = name;
19          t.start = d;
20          t.end = d + 30 minutes;
21          addTask(c, t);
22      }
23  }
```

```
3
4   Boolean conflict( Calendar c, Task t) {
5       Boolean conflicts = false;
6       every Task task in c{
7           if ( not (task.end < t.start || t.end < task.start)) {
8               conflicts = true;
9               break;
10          }
11      }
12      return conflicts;
13  }
14
```

- Data-Types
  - Calendar
    - TimeFrame: 1 hour, 2 days, 3 weeks
    - TimeEntity: Monday, Weekend, May
    - Date: 2013.05.16, 2013.05.16.16.00
  - Task
    - Number: 1, 2, 3, 4, 5, ...
    - String: "str1", "str2"
    - Boolean: true, false
- Conditionals
  - if...else...

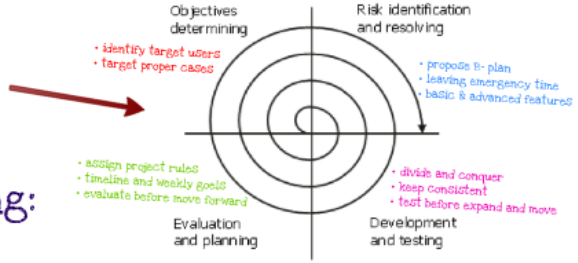
```
15    Boolean isMay(Date d){
16        if (is(d, May))
17            return true;
18        return false;
19    }
```



Project Management

Software Development Process:

spiral model



Version control & Process tracking:

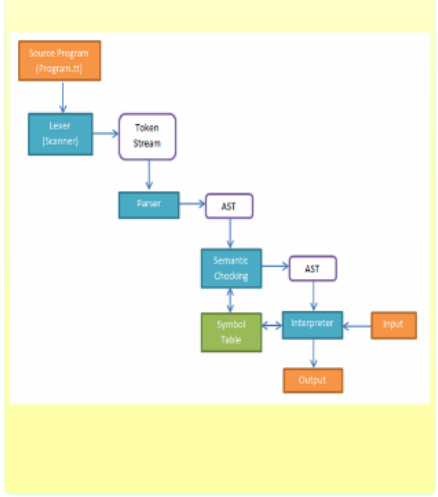
github + fogbugs

Process tracking:

timeline + weekly goals

Team communication:

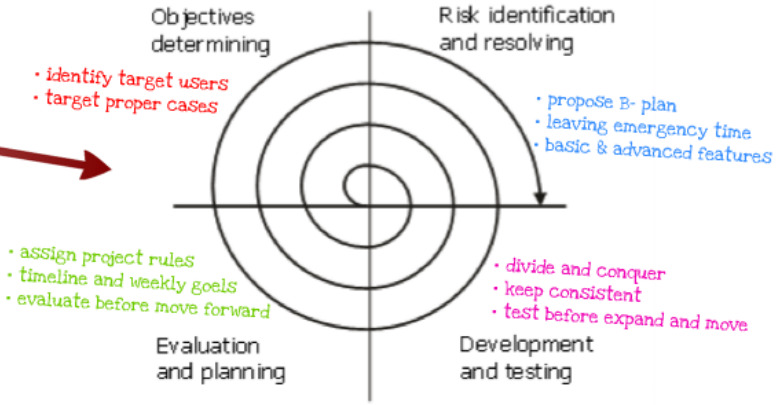
weekly meeting + google drive



Project Management

Software Development Process:

spiral model



Version control & Process tracking:

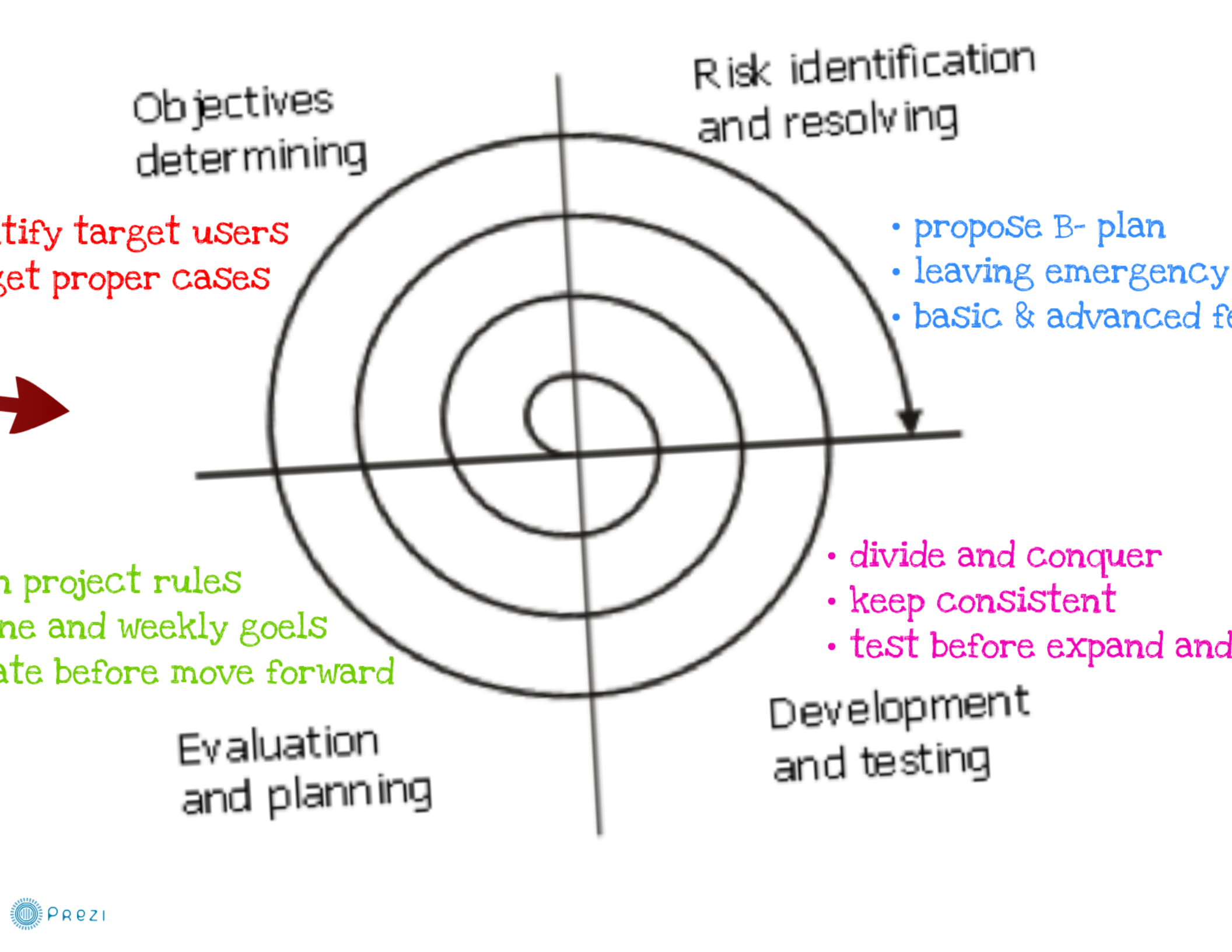
github + fogbugs

Process tracking:

timeline + weekly goals

Team communication:

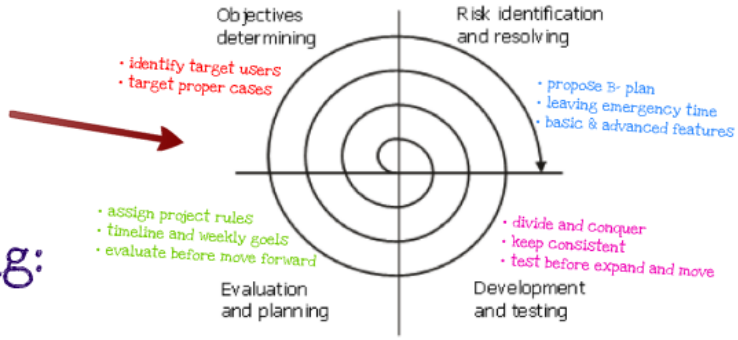
weekly meeting + google drive


Objectives determining
Risk identification and resolving
tify target users
get proper cases
• propose B- plan
• leaving emergency
• basic & advanced fe
n project rules
ne and weekly goels
ate before move forward
• divide and conquer
• keep consistent
• test before expand and
Evaluation and planning
Development and testing

Project Management

Software Development Process:

spiral model



Version control & Process tracking:

github + fogbugs

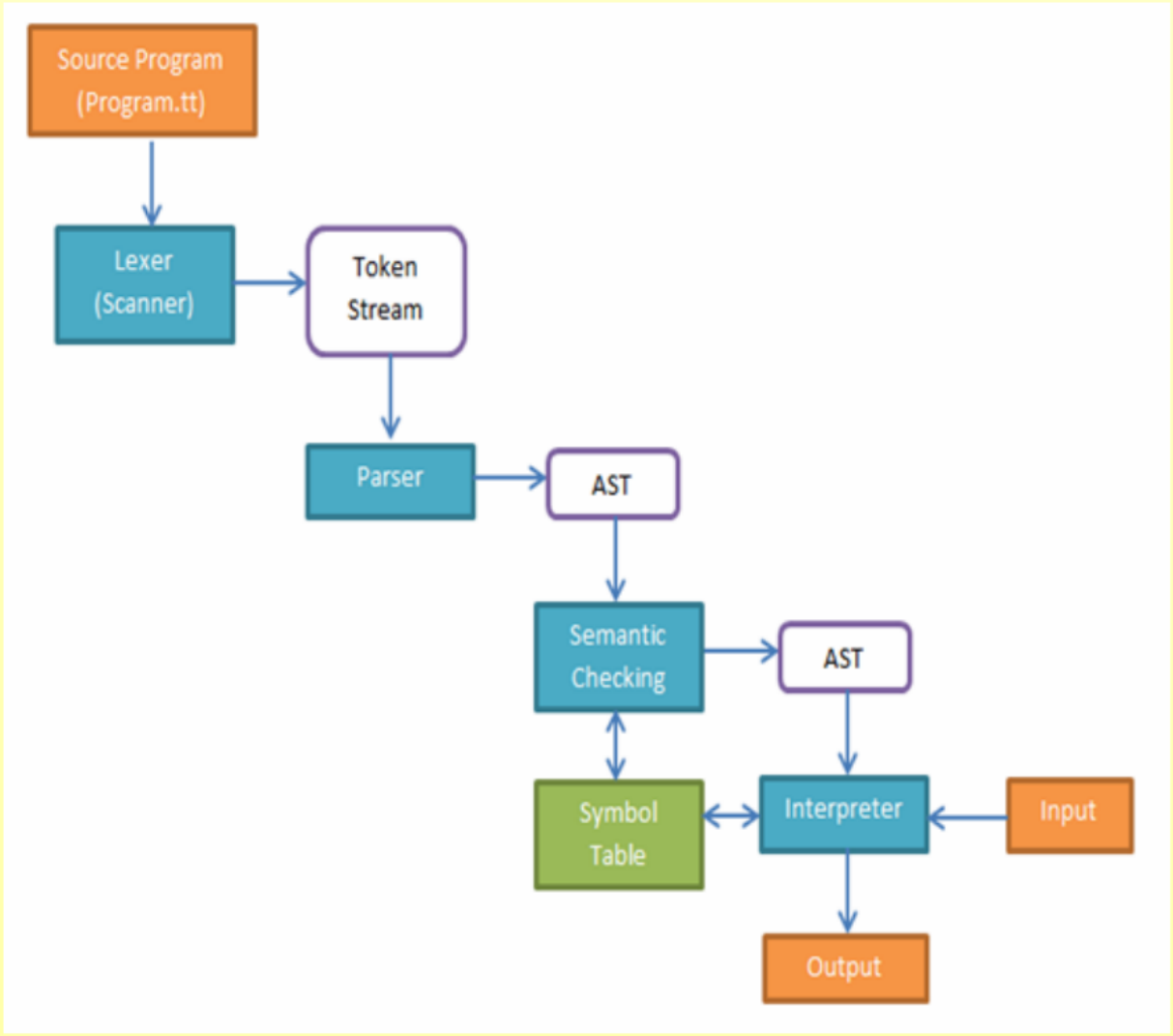
Process tracking:

timeline + weekly goals

Team communication:

weekly meeting + google drive

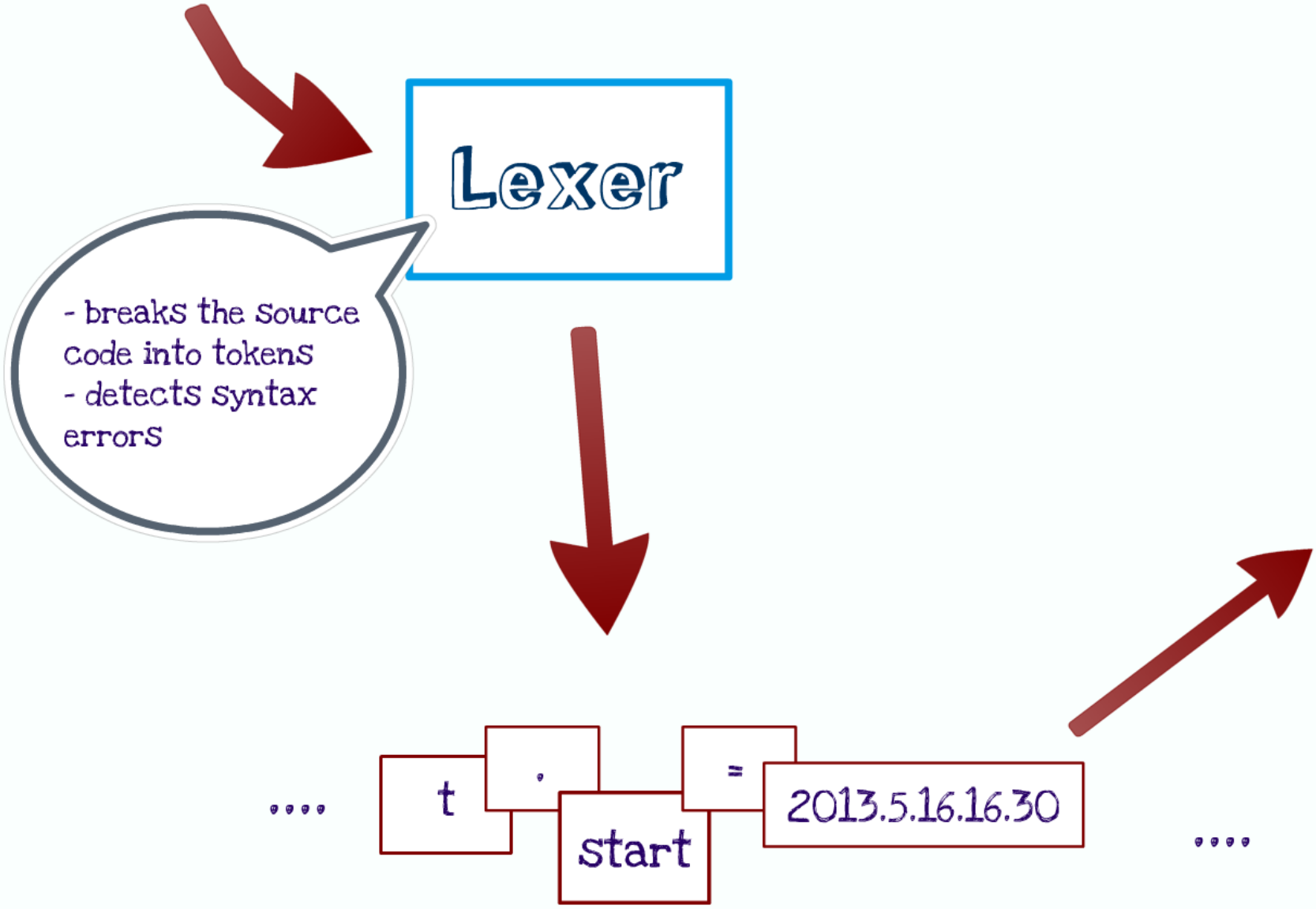
Architectural Block Diagram

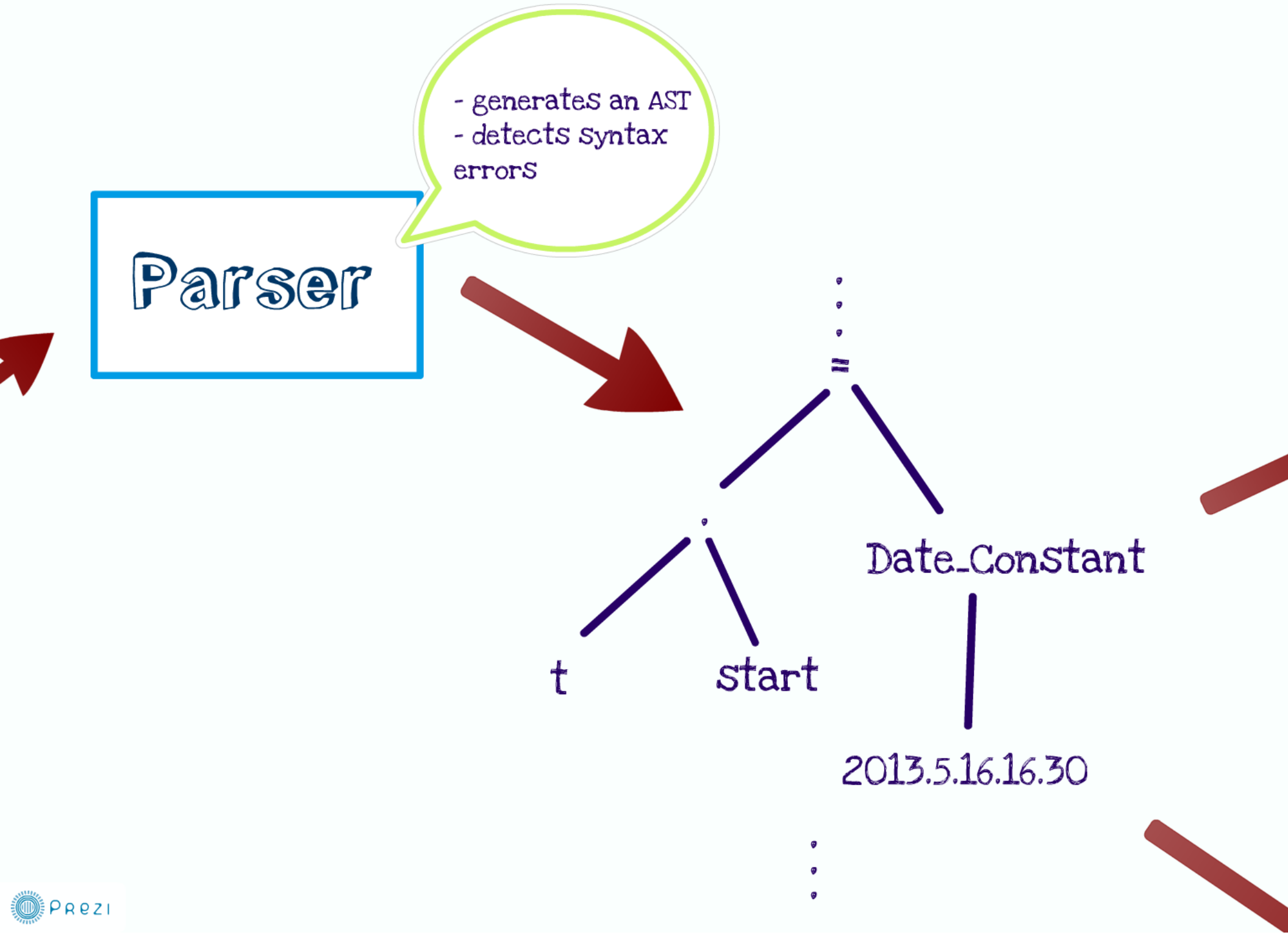


....

t.start = 2013.5.16.16.30;

....

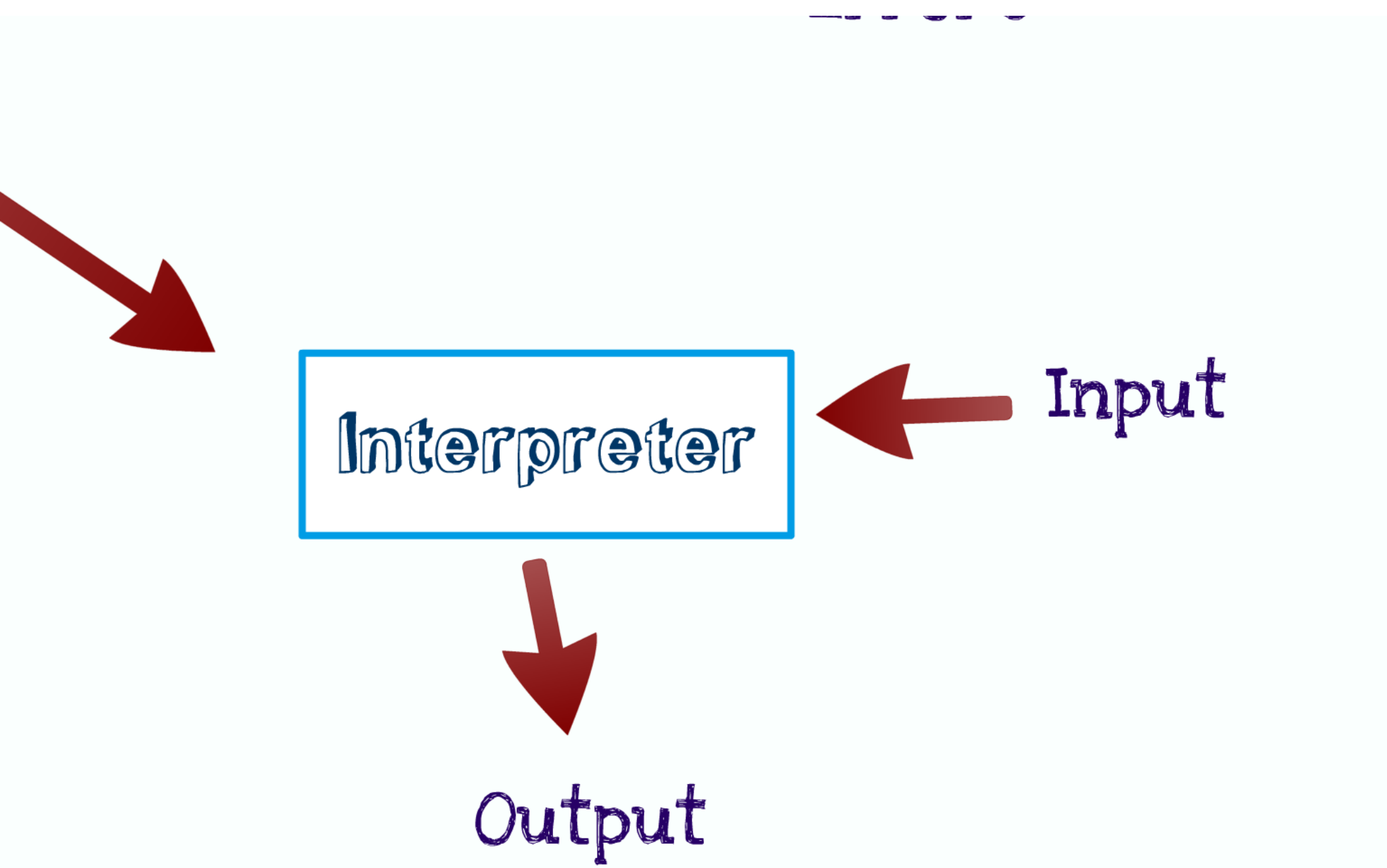



- generates an AST
- detects syntax errors
Parser
=
.
t
start
Date_Constant
2013.5.16.16.30

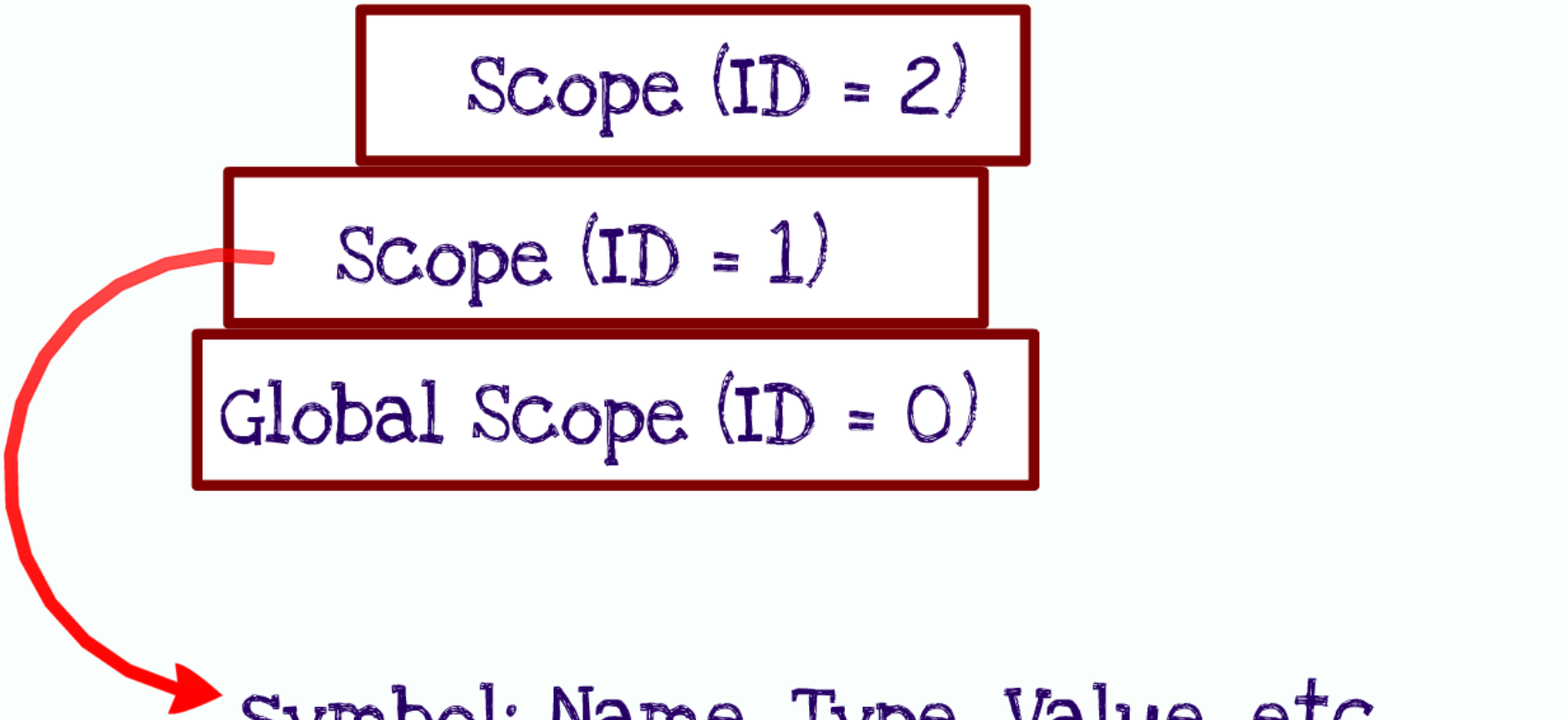
- Type Checking
- analyze functions and variables and put them into Symbol Table

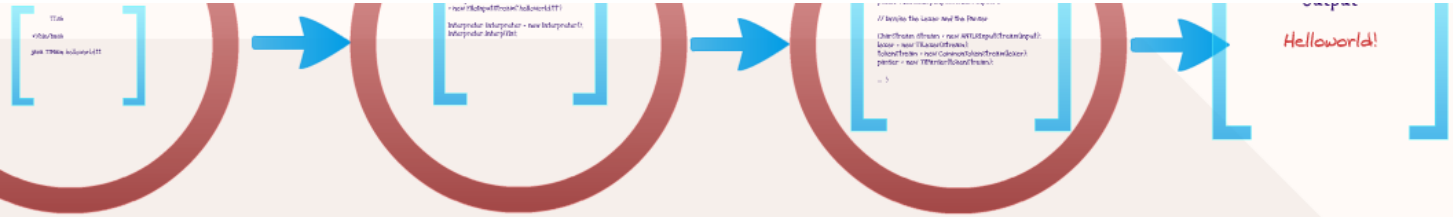
Semantic Checking

Semantic Errors



Symbol Table





Development Tools

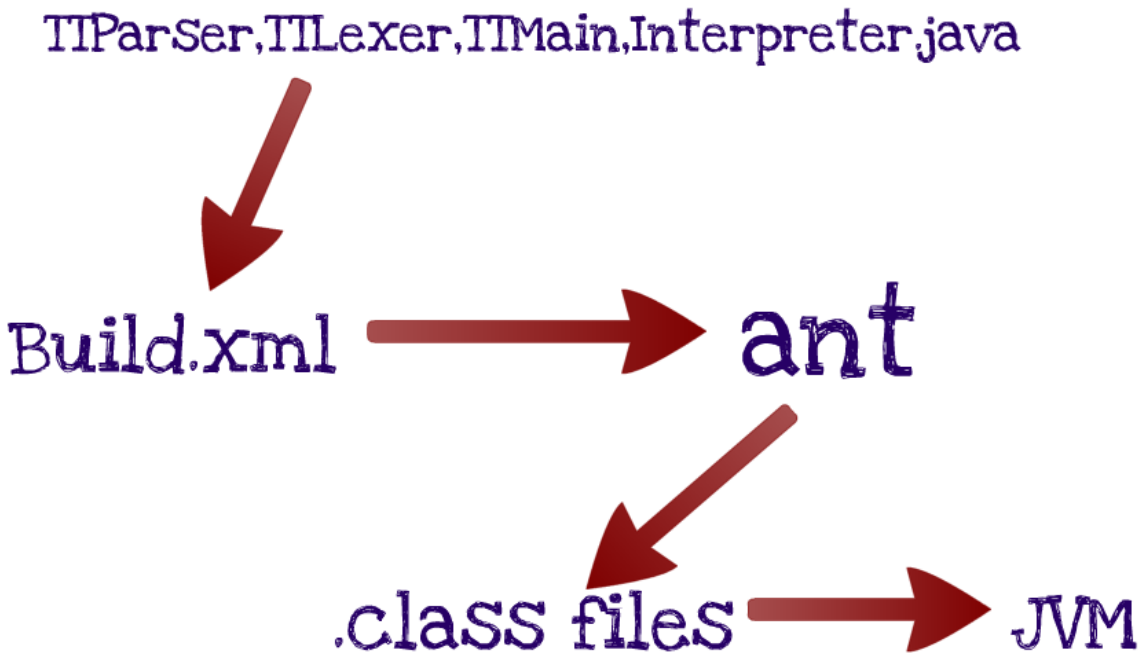
<APACHE ANT>

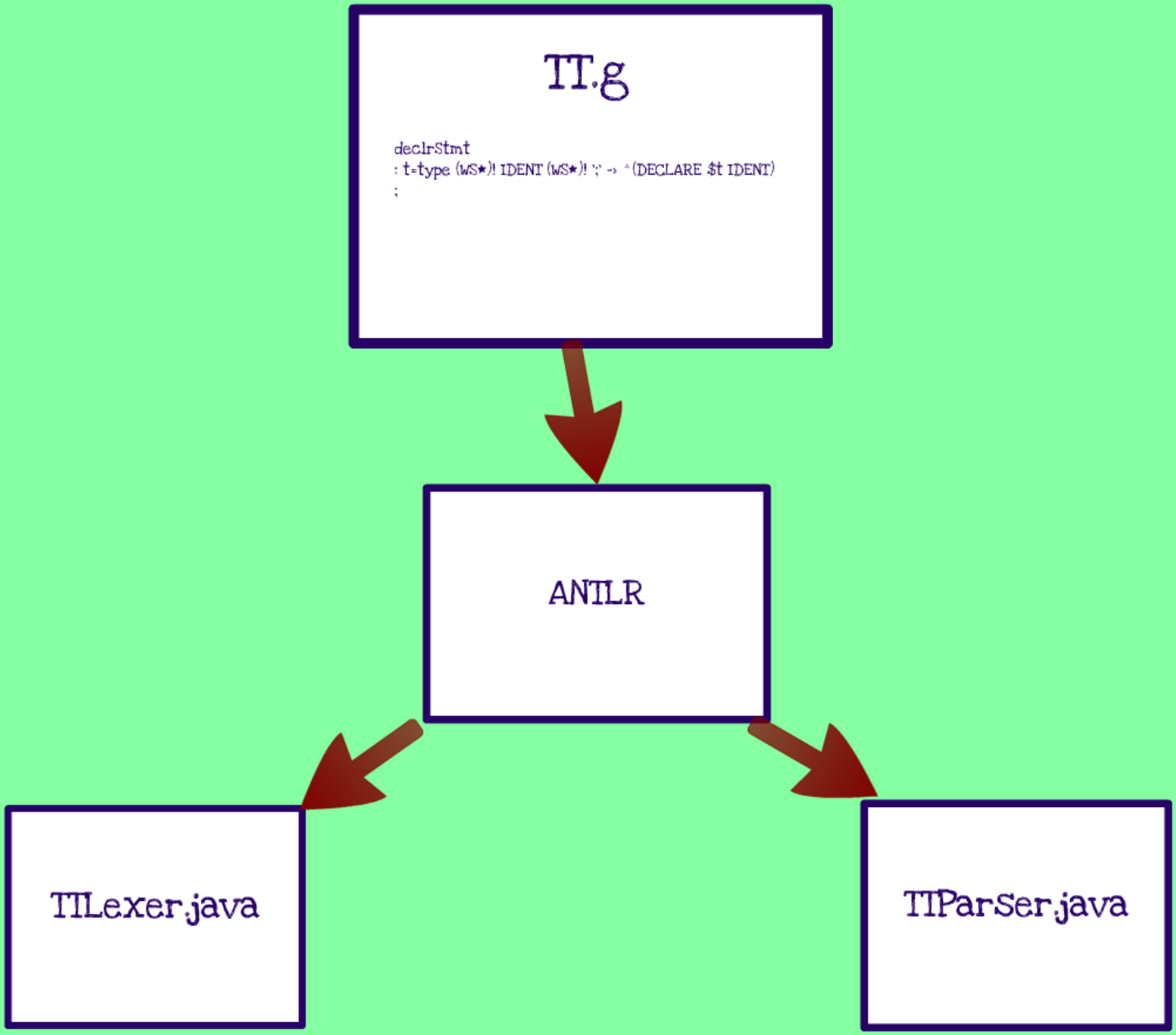
eclipse

ubuntu

GitHub

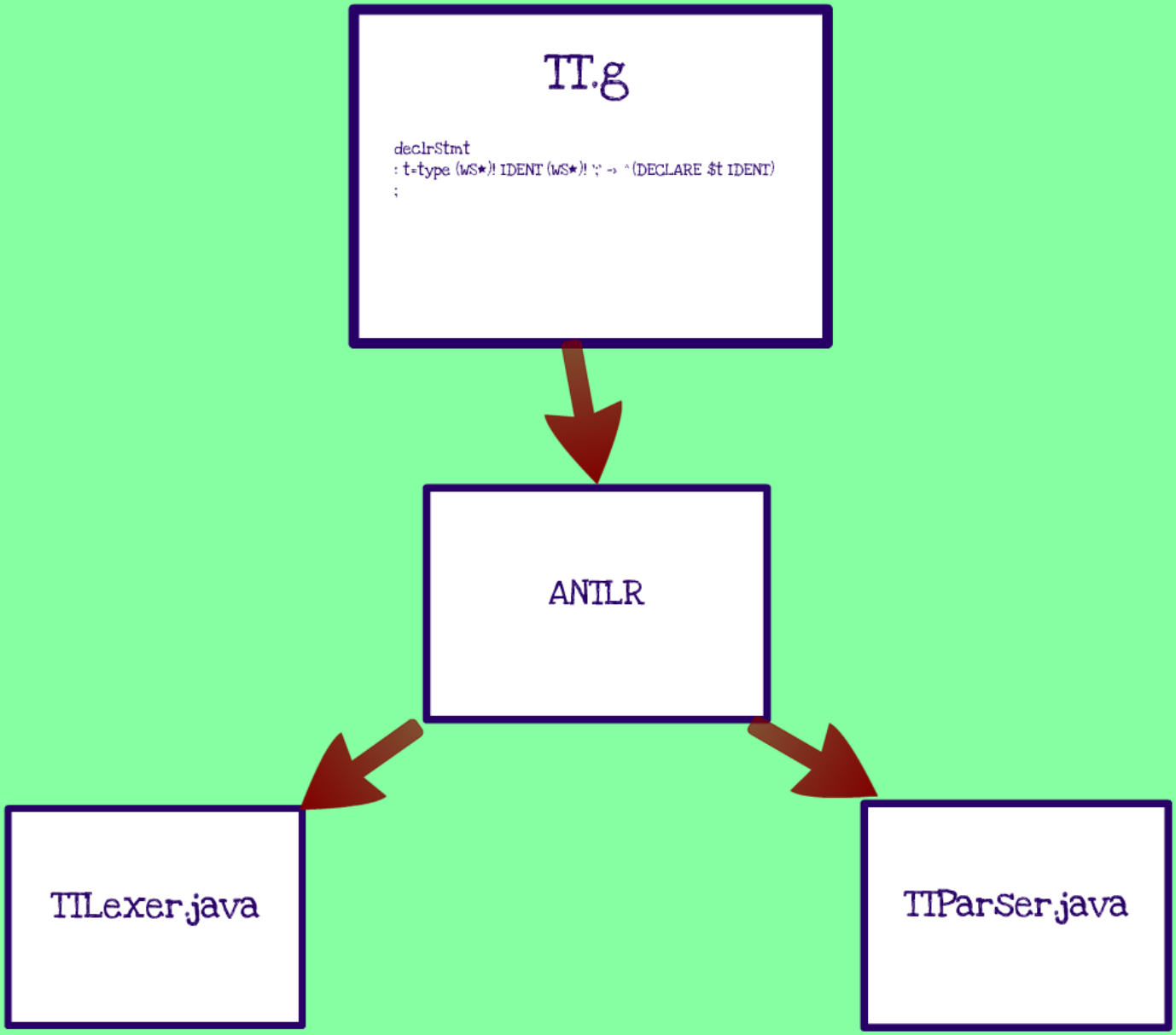
ANTLR


ITParSer,ITLexer,ITMain,Interpreter.java
Build.xml
ant
.class files
JVM

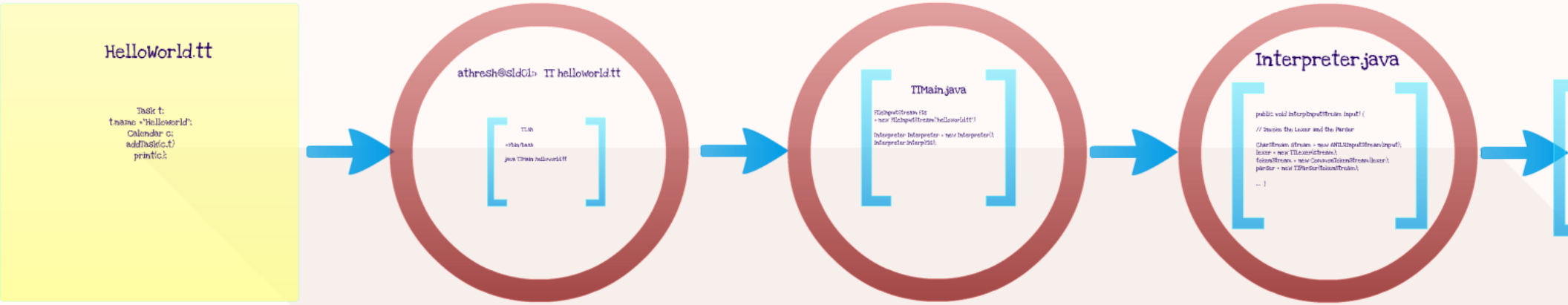

TT.g
declrStmt
: t=type (WS*)! IDENT (WS*)! ';' -> ^(DECLARE $t IDENT)
;
ANTLR
TILexer.java
TIParSer.java

TT.g

```
declrStmt
: t=type (WS*)! IDENT (WS*)! ';' -> ^(DECLARE $t IDENT)
;
```


TT.g
declrStmt
: t=type (WS*)! IDENT (WS*)! ';' -> ^(DECLARE $t IDENT)
;
ANTLR
TILexer.java
TIParSer.java

Run-Time Environment



Development Tools

Helloworld.tt

```
Task t;
t.name ="Helloworld";
Calendar c;
addTask(c,t)
print(c);
```

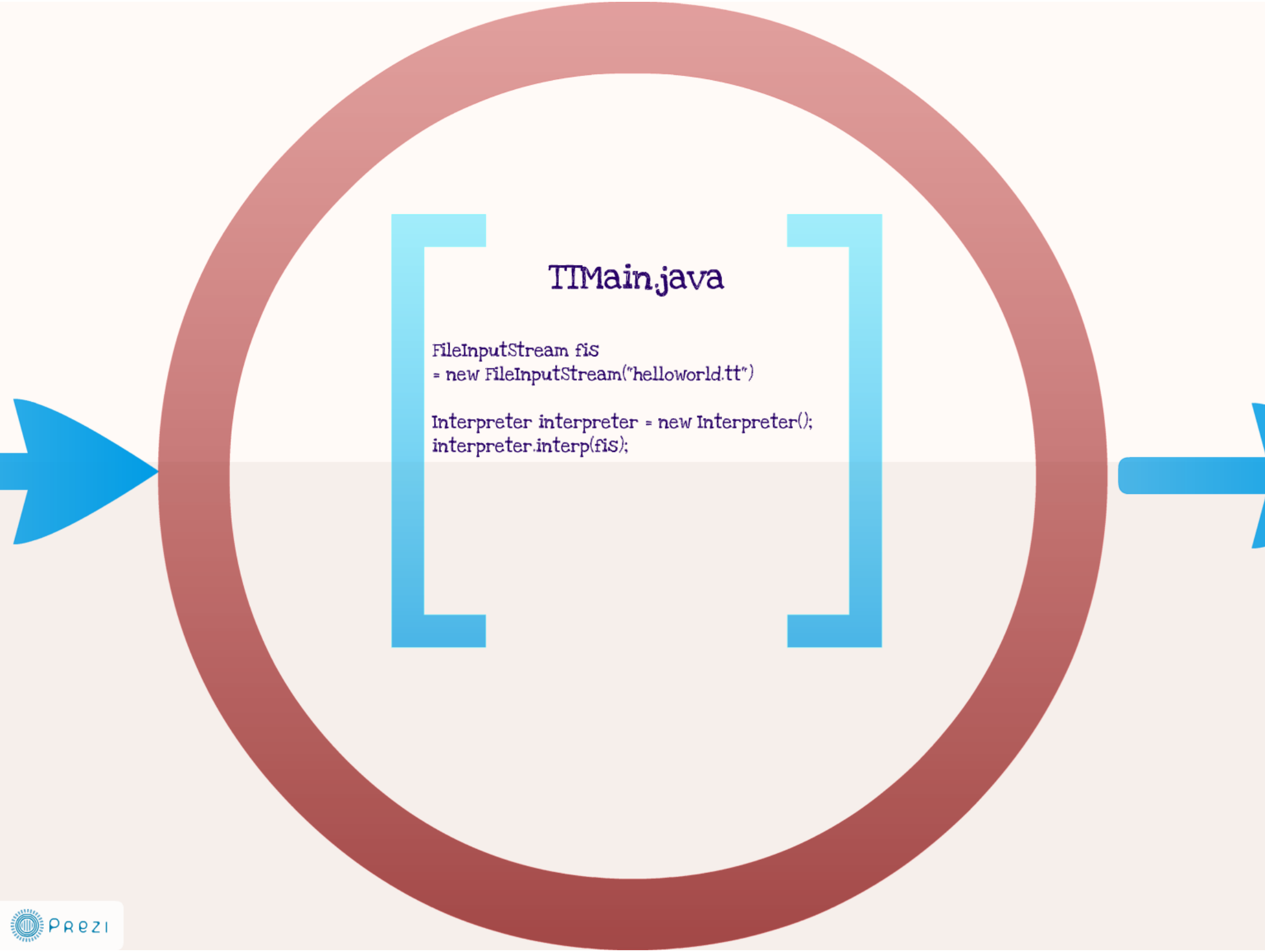
athresh@sld01:› TT helloworld.tt

TT.sh

```
#!/bin/bash

java TTMain helloworld.tt
```

TT.sh

```
#!/bin/bash

java TTMain helloworld.tt
```


TTMain.java
FileInputStream fis
= new FileInputStream("helloworld.tt")
Interpreter interpreter = new Interpreter();
interpreter.interp(fis);

TTMain.java

```
FileInputStream fis
= new FileInputStream("helloworld.tt")

Interpreter interpreter = new Interpreter();
interpreter.interp(fis);
```

Interpreter.java

```
public void interp(InputStream input) {

// Invoke the Lexer and the Parser

CharStream stream = new ANTLRInputStream(input);
lexer = new TILexer(stream);
tokenStream = new CommonTokenStream(lexer);
parser = new TIParser(tokenStream);

... }
```

```
public void interp(InputStream input) {

// Invoke the Lexer and the Parser

CharStream stream = new ANTLRInputStream(input);
lexer = new TILexer(stream);
tokenStream = new CommonTokenStream(lexer);
parser = new TIParser(tokenStream);

... }
```

Output

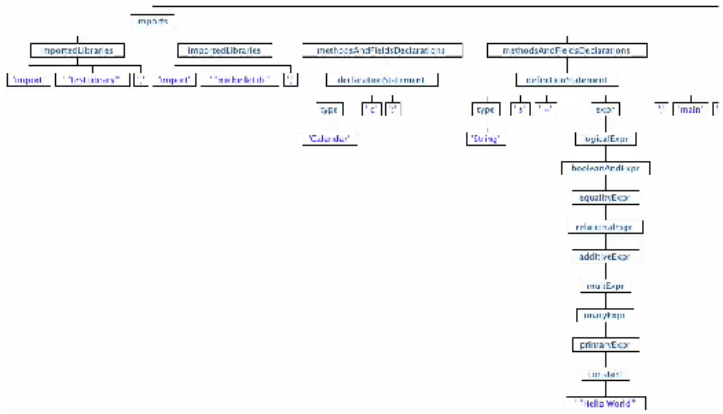
Helloworld!


TESTING
FogBugz:
https://pit-tt.fogbugz.com/
JUnit:
Test Programs
• decalaredefine.tt
• ifelse.tt
• Loops.tt
• methods.tt
• operator.tt
• Program1.tt
• standard.library.tt
• strcmp.tt
Tutorial Programs
• tutorial_daysLeft.tt
• tutorial_HelloWorld.tt
• tutorial_MeetingTime.tt
• tutorial_RecurringTask.tt
• tutorial_ScheduleTasks.tt
• tutorial_studyTime.tt
Syntax:
Prezi

Syntax:

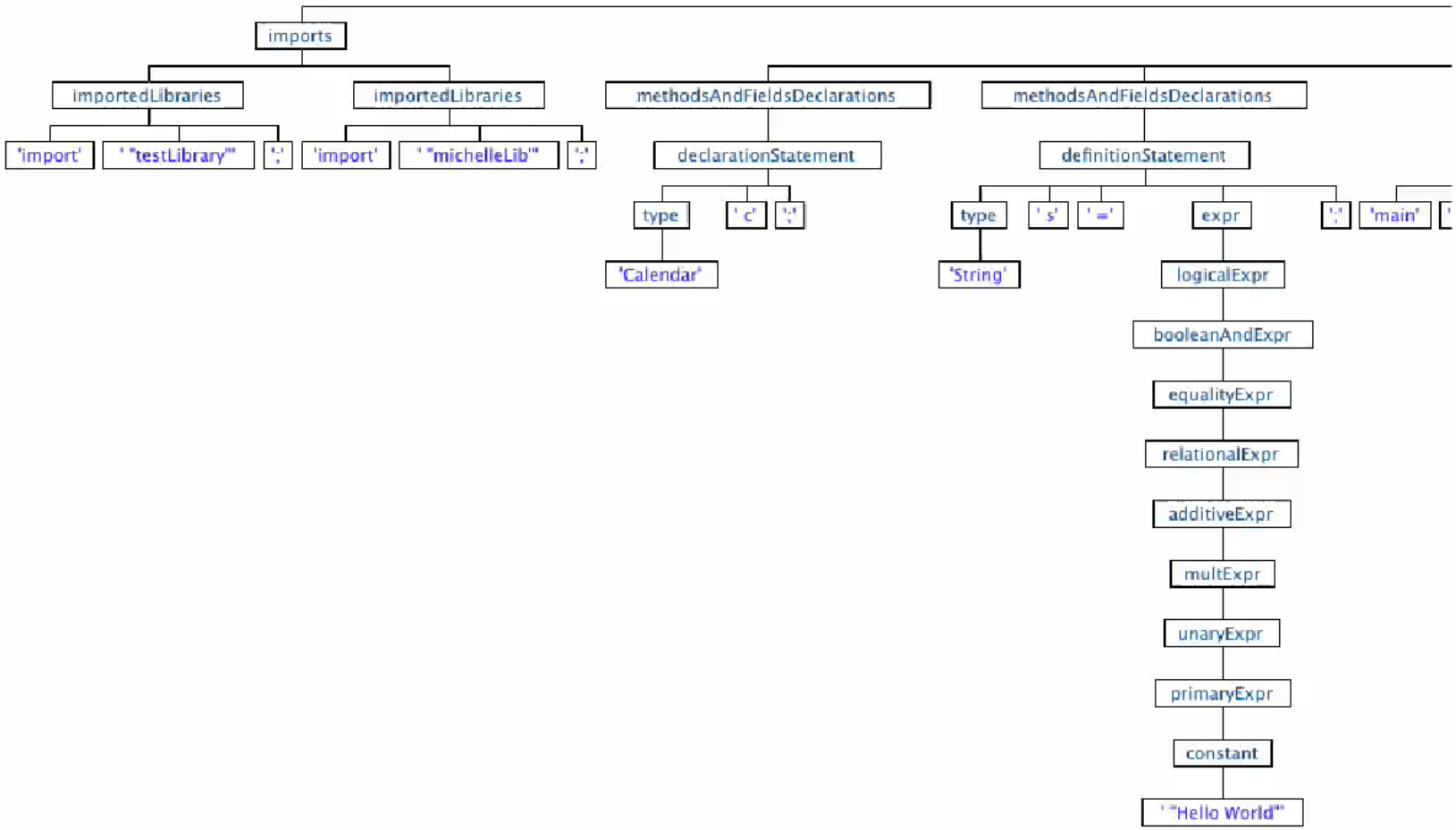
```
// Start with translationUnit
// Test imports and importedLibraries
import "testLibrary";
import "michelleLib";

// Test programBody and methodsAndFieldsDeclarations

// test GLOBAL declarationStatement
Calendar c;

// test GLOBAL definitionStatement
String s = "Hello World";

// Test main and block
main() {
// Test statement_type
// Test type and declarationStatement
String s;
Number n;
Date d;
Task t;
TimeFrame tf;
Calendar c;
TimeEntity time;

// Test basic definitionStatement for all type
String s = "hello there!";
Number n = 13;
Date d = 2013.02;
Task t = "Do Work";
TimeFrame tf = 1 day + 4 days;
Calendar c = "My Calendar";
TimeEntity mon = Monday;

// Test assignmentStmt
var = 3;

// Test ifThenStatement
if ( 3 < a) {
print ("3 is < a!");
}
...
```

Parse Tree:



```
if ( a == b ) {
// Do Something
}
else {
// Do Something else
}

// Test everyFromToByStatement also dateOrIdent
and timeframeOrIdent
every Date d from 2013.01.10 to endDate by 3 days
{

}

// Test everyInStatement
every Task t in c {
// test breakStatement
break;
}

// test constraintOptions
every Task t in c from 2013.05.02.10.30 to
2013.05.02.12.30 {
// test continueStatement
continue;
}
...
```

Parse Tree:



FogBugz:

https://plt-tt.fogbugz.com/

CASES IN SAMPLE PROJECT

Case	Title	Status
2	String does not accept spaces commenting out the two rul...	Closed (Fixed)
3	IDENT and STRING conflict use fragment if two Lexer rule...	Closed (Fixed)
4	YEAR and NUMBER conflict use fragment keyword	Closed (Fixed)
5	Force bracket use in if and if else statements? Yes, force ...	Closed (Fixed)
6	timeFrame and timeFrameConstant break allowing spaces ...	Closed (Fixed)
7	logical statements should take IDENT ex (e < 3) someo...	Resolved (Fixed)
8	what are all the every loop options Yes, include the last 2 ...	Closed (Fixed)
9	punctuation should be allowed inside of strings try to use ...	Resolved (Fixed)
10	should Read be upper case and print lower case Read sho...	Closed (Fixed)
11	timeFrameDefnStmt should allow assignment with 'until' W...	Closed (Fixed)
12	definition of a variable from another variable? var1 = var2;...	Closed (Fixed)
13	we shouldn't need WS* in between definitions in .g file, it is...	Closed (Fixed)
14	dateConstant does not work I think we can just accept 01-...	Resolved (Fixed)
15	Discontinuity Timeframe or TimeFrame	Resolved (Fixed)
16	Java.util.Calendar vs our Calendar.java We are probably ...	Closed (Fixed)

JUnit:

```
public void testIfElse()
{
FileInputStream fileStream = null;
try {

fileStream = new
FileInputStream("src/columbia/plt/tt/programs/ifelse.tt");
interpreter.interp(fileStream);

String[] programOut = outContent.toString().split("\n");
assertEquals(programOut[0], "YES n is in range");
assertEquals(programOut[1], "c is less than d");

} catch (FileNotFoundException e) {
// TODO Auto-generated catch block
e.printStackTrace();
} catch (antlr.RecognitionException e) {
// TODO Auto-generated catch block
e.printStackTrace();
} catch (IOException e) {
// TODO Auto-generated catch block
e.printStackTrace();
} catch (RecognitionException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
}
```

Test Programs

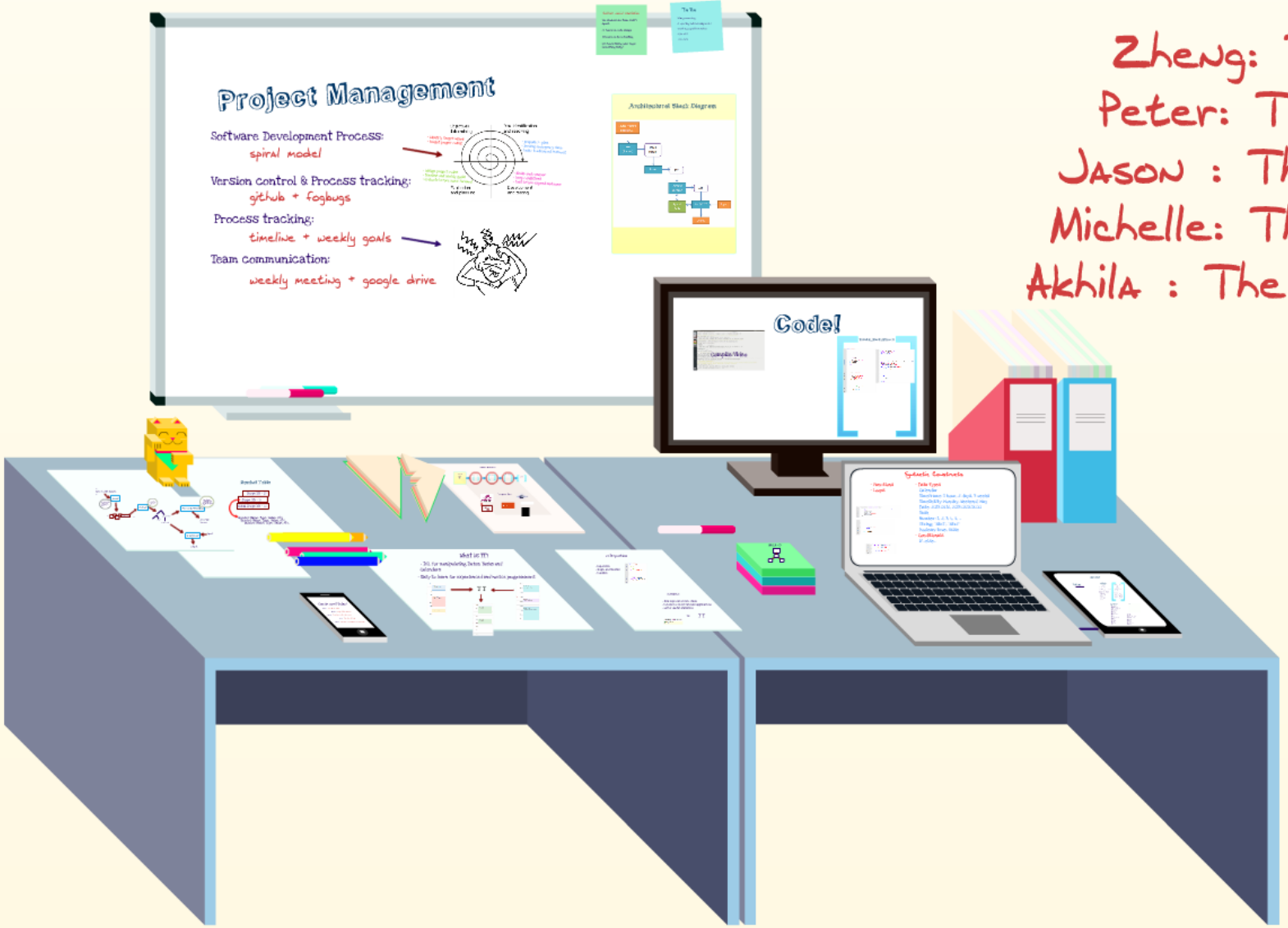
- decalaredefine.tt
- ifelse.tt
- Loops.tt
- methods.tt
- operator.tt
- Program1.tt
- standard_library.tt
- strcmp.tt

Tutorial Programs

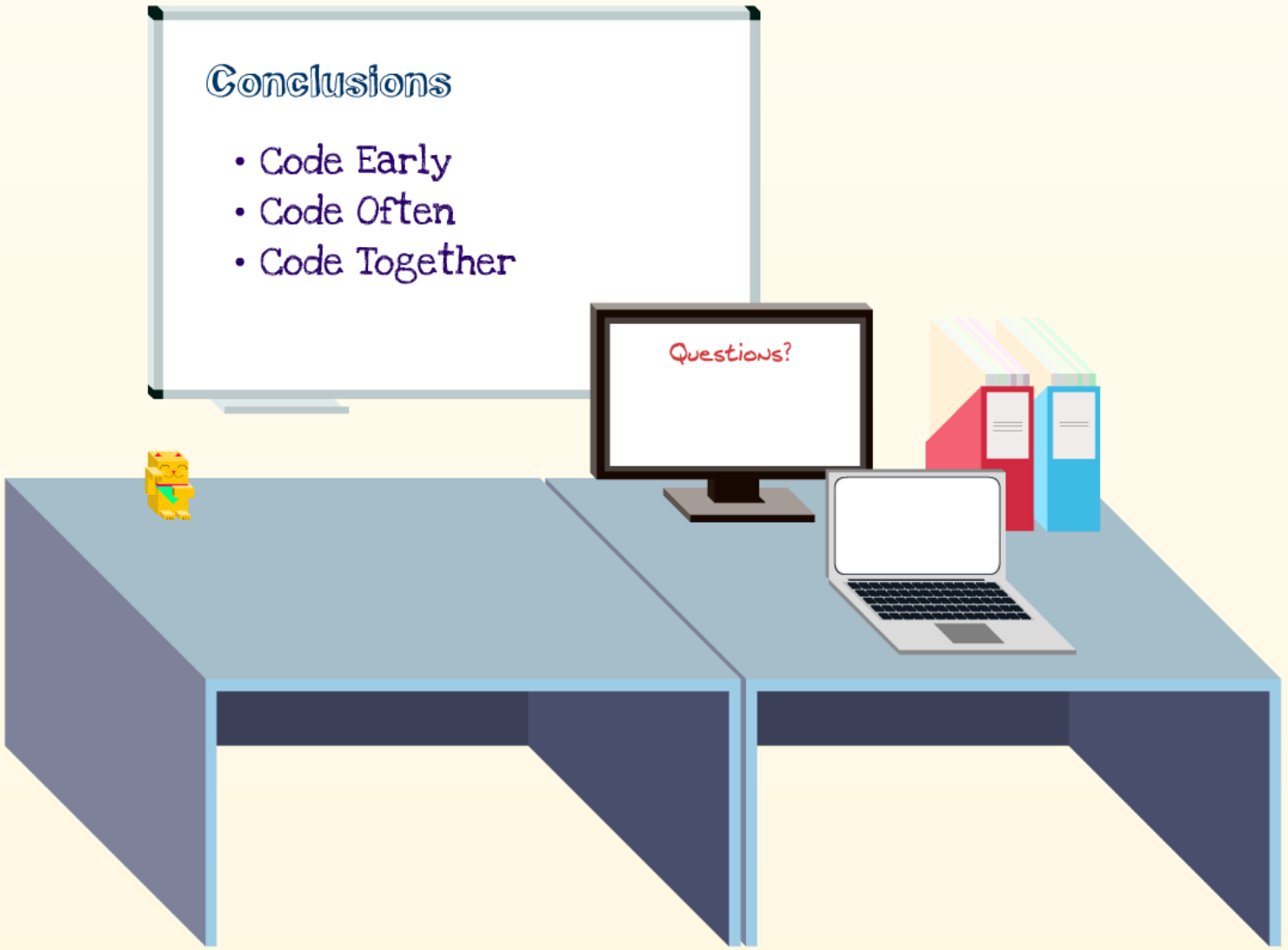
- tutorial_daysLeft.tt
- tutorial_HelloWorld.tt
- tutorial_MeetingTime.tt
- tutorial_RecurringTask.tt
- tutorial_ScheduleTasks.tt
- tutorial_StudyTime.tt

Time & Task

Zheng: The PM
Peter: The Guru
Jason : The Architect
Michelle: The Tester
Akhila : The Integrator



Your Life After TT!

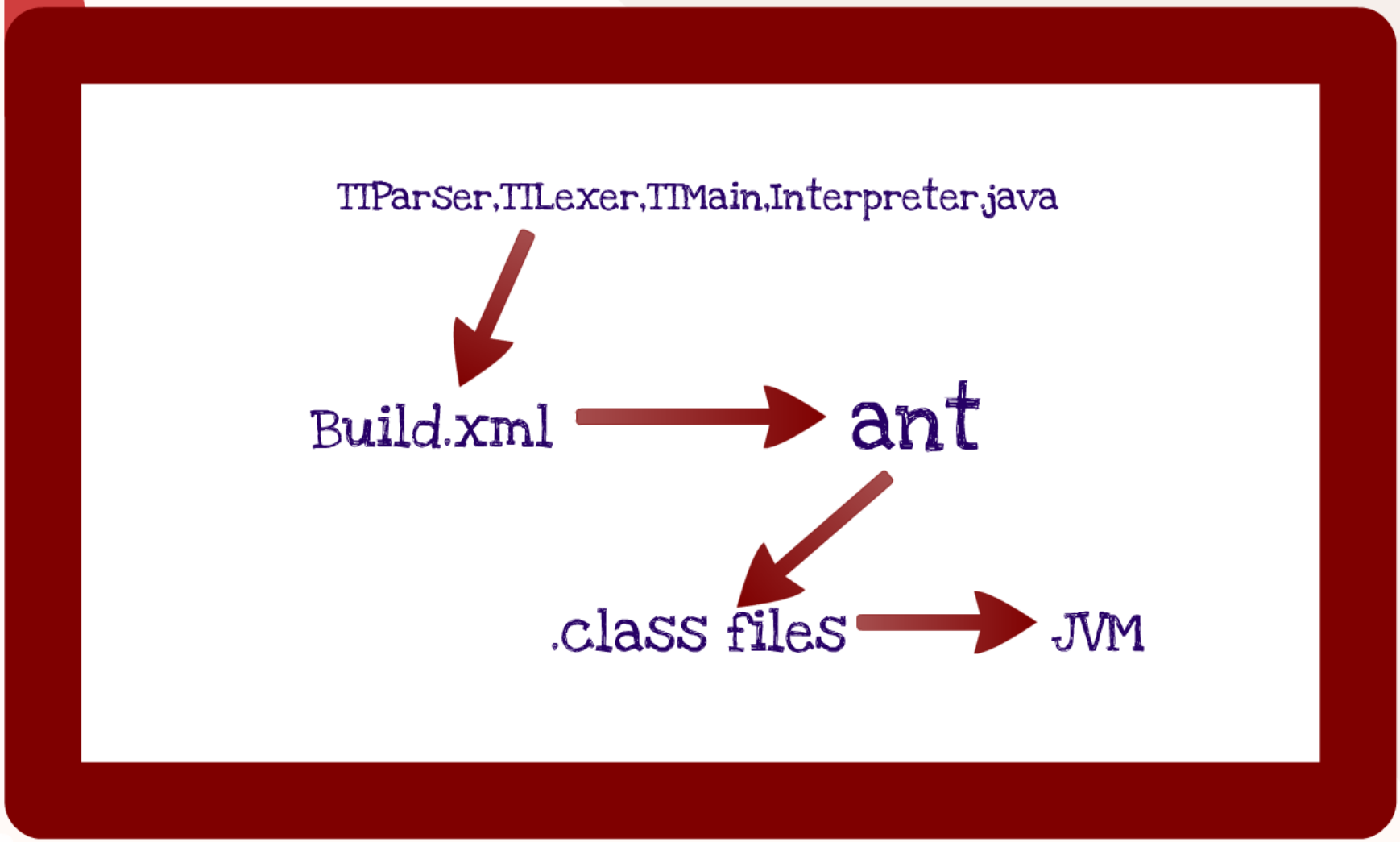


Conclusions

- Code Early
- Code Often
- Code Together

Questions?


Syntactic Constructs
- Functions
- Loops
- Data-Types
Calendar
TimeFrame: 1 hour, 2 days, 3 weeks
TimeEntity: Monday, Weekend, May
Date: 2013.05.16, 2013.05.16.16.00
Task
Number: 1, 2, 3, 4, 5, ...
String: "Str1", "Str2"
Boolean: true, false
- Conditionals
if..else...
ANTLR v3
IT Properties
TT


TIParser,TILexer,TIMain,Interpreter.java
Build.xml
ant
.class files
JVM