

**UNIVERSITY OF OSLO**  
**Department of Informatics**

**A LISP Interpreter  
in Java**

Mandatory Assignment 2

INF3110

November 2, 2009



## 1 Introduction

In assignment 1, it was required to use a possibly new language and programming paradigm to solve a difficult problem. A large part of the program was already completed, in order to make the problem more manageable.

Since this assignment uses a known environment (the Java platform) to solve a known problem (exactly the same as assignment 1), the amount of pre-code given is far less.

## 2 Structure and Bounds

The pre-code is for your convenience. You're allowed to solve this problem in any manner you see fit, this implies that it's allowed to make changes to the pre-code. Focus on writing a structured solution in an object oriented style.

## 3 Pre-Code

### 3.1 The Parser and Lexer

As last time, the Lexer and the Parser are already done. You shouldn't have to change any of the files in the parser package (oblig2.parser.\*).

### 3.2 Interpreter

The interpreter reads the library file and starts the repl, the *main* method is here.

### 3.3 Symboltable

The symboltable is the environment. It's a mutable data structure, a list of hash tables. This structure will change as the interpreter evaluates expressions.

Each LISP function, in order to maintain static scoping, needs the environment that existed at creation time. The Symboltable has a *copy* method for this purpose.

### 3.4 SExp

This interface contains the method *interp*. *Interp* is supposed to have the same functionality as the ML function *eval*, from Assignment 1.

```
public SExp interp(Symboltable env);
```

### 3.5 SexpDummy

This class implements the SExp interface. It's of none real use, other than to make the pre-code compilable and runnable.

## 4 Editing and Running the program

The program may be compiled and executed from a terminal using ant:

```
\$ ant # or ant help
\$ ant compile
\$ ant run
```

### 4.1 IDE

It's recommended using an IDE like Eclipse or Netbeans. To import the pre-code as an Eclipse project (tested using version 3.5), do the following:

1. Press File (from the toolbar), then Import
2. From the General tab, select *Existing Project into Workspace*
3. Use the *select archive file* option.
4. Select the pre-code zip file.
5. Click finish.

## 5 Deadline

The assignment must be delivered within Sunday the 15th of November. This is a **hard deadline**.