Concepts of programming languages - Bonus Project Language Documentation

TECHNISCHE UNIVERSITÄT DARMSTADT

Asad ur Rehman, Philipp Schreiber, Ritesh Shrestha, Jakob Stock
August 25, 2022

1 Repository

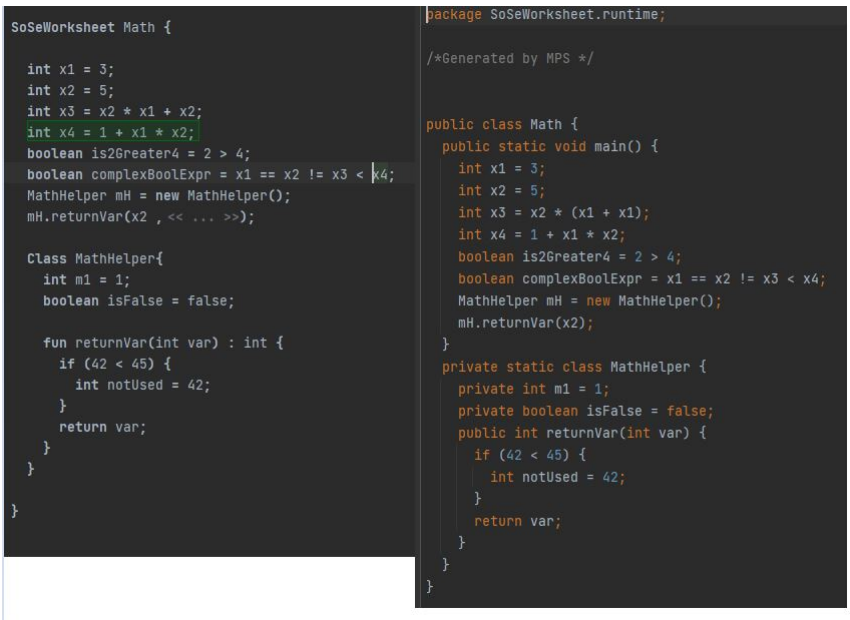
https://github.com/riteshCodes/cop-2022

2 Features

- Int and Bool Var Definition
- Variable References
- Expressions
  - +, +, *, / operator
  - brackets get generated in the java code if a lower precedence operator is chosen inside a higher precedence operator, e.g. + inside a / expression
  - ==, !=, <=, <, >, >= operator
  - int, bool literals
- Reassignments
- flow control
  - if, for, while
  - just one statement in each body of the flow control constructs
- classes
  - can have 0..n vars

  - can have 0..n functions
- functions
  - can have 0..n params
  - just one statement in the body
  - must have an return statement at the end
  - return type bool or int
- objects
  - Java syntax (just with an empty constructor)
  - can call methods and vars from object
- Scopes
  - just one var name per scope
- Constraints
  - var, class and method names can't have white spaces in the name
- Generator
  - Can generate Java code out of the Worksheets

3 Examples



```
SoSeWorksheet Math {

  int x1 = 3;
  int x2 = 5;
  int x3 = x2 * x1 + x2;
  int x4 = 1 + x1 * x2;
  boolean is2Greater4 = 2 > 4;
  boolean complexBoolExpr = x1 == x2 != x3 < x4;
  MathHelper mH = new MathHelper();
  mH.returnVar(x2 , << ... >>);

  Class MathHelper{
    int m1 = 1;
    boolean isFalse = false;

    fun returnVar(int var) : int {
      if (42 < 45) {
        int notUsed = 42;
      }
      return var;
    }
  }

}
```

```
package SoSeWorksheet.runtime;

/*Generated by MPS */


public class Math {
  public static void main() {
    int x1 = 3;
    int x2 = 5;
    int x3 = x2 * (x1 + x1);
    int x4 = 1 + x1 * x2;
    boolean is2Greater4 = 2 > 4;
    boolean complexBoolExpr = x1 == x2 != x3 < x4;
    MathHelper mH = new MathHelper();
    mH.returnVar(x2);
  }
  private static class MathHelper {
    private int m1 = 1;
    private boolean isFalse = false;
    public int returnVar(int var) {
      if (42 < 45) {
        int notUsed = 42;
      }
      return var;
    }
  }
}
```

Figure 1: Example Expressions

```
SoSeWorksheet Class {

  int age = 42 - 5 + 2;
  Dog dog = new Dog();
  dog.setAge(age , << ... >>);
  age = dog.age + age;

  Class Dog{
    int age = 0;
    boolean isMale = false;

    fun setAge(int a) : int {
      age = a;
      return a;
    }

    fun setGenderToMale() : boolean {
      isMale = true;
      return false;
    }

    fun add(int x, int y) : int {
      int varInAdd = x;
      return x + y;
    }
  }

}
```

```
package SoSeWorksheet.runtime;

/*Generated by MPS */


public class Class {
  public static void main() {
    int age = 42 - 5 + 2;
    Dog dog = new Dog();
    dog.setAge(age);
    age = dog.age + age;
  }
  private static class Dog {
    private int age = 0;
    private boolean isMale = false;
    public int setAge(int a) {
      age = a;
      return a;
    }
    public boolean setGenderToMale() {
      isMale = true;
      return false;
    }
    public int add(int x, int y) {
      int varInAdd = x;
      return x + y;
    }
  }
}
```

Figure 2: Example Classes

```
SoSeWorksheet FlowControl {

  int i = 0;
  int inc = 3;
  boolean isFalse = false;

  if (isFalse) {
    i = 1 + i;
  }

  for (int j = 0; 42; 1) {
    i = i + inc;
  }

  i = 0;

  while (i < 1000) {
    i = i * inc;
  }

  << ... >>

}
```

```
package SoSeWorksheet.runtime;

/*Generated by MPS */


public class FlowControl {
  public static void main() {
    int i = 0;
    int inc = 3;
    boolean isFalse = false;
    if (isFalse) {
      i = 1 + i;
    }
    for (int j = 0; j < 42; j = j + 1) {
      i = i + inc;
    }
    i = 0;
    while (i < 1000) {
      i = i * inc;
    }
  }
}
```

Figure 3: Example Flow Control