Why GitHub?

Enterprise

Explore

Marketplace

Pricing

Search

/

Sign in   Sign up

sh-soltanpour / **compilerPhase2**

Watch 0   ★ Star 0   Fork 0

<> Code   ⊙ Issues **0**   Pull requests **0**   ▥ Projects **0**   Security   Insights

Add actor and variable addition to symbol table
Loading branch information

Browse files

sh-soltanpour committed Dec 4, 2017

1 parent 45735ba   commit 16233d177234a75fc65c636edb8981a7273f637f

Showing **8 changed files** with **176 additions** and **9 deletions**.

Unified   Split

⌄ 17 ■■■■ Actor.java

...

```
@@ -0,0 +1,17 @@
1  + public class Actor {
2  +   public Actor(String name, int mailboxSize) {
3  +     this.name = name;
4  +     this.mailboxSize = mailboxSize;
5  +   }
6  +
7  +   public String getName() {
8  +     return this.name;
9  +   }
10 +
11 +   public int getMailboxSize() {
12 +     return this.mailboxSize;
13 +   }
14 +
15 +   String name;
16 +   int mailboxSize;
17 + } 0↵
```

⌄ 66 ■■■□ Atalk.g4

...

```
@@ -1,17 +1,65 @@
1   1     grammar Atalk;
2   -
    2   + @members{
    3   +   void print(String str){
```

```
  3 +    void print(String str){
  4 +        System.out.println(str);
  5 +    }
  6 +  void beginScope() {
  7 +      int offset = 0;
  8 +      if(SymbolTable.top != null)
  9 +          offset = SymbolTable.top.getOffset(Register.SP);
 10 +      SymbolTable.push(new SymbolTable());
 11 +      SymbolTable.top.setOffset(Register.SP, offset);
 12 +    }
 13 +  void endScope(){
 14 +    print("Stack offset: " + SymbolTable.top.getOffset(Register.SP));
 15 +    SymbolTable.pop();
 16 +  }
 17 +  void putLocalVar(String name, Type type) throws ItemAlreadyExistsException{
 18 +    SymbolTable.top.put(
 19 +            new SymbolTableLocalVariableItem(
 20 +                new Variable(name, type),
 21 +                SymbolTable.top.getOffset(Register.SP)
 22 +            )
 23 +        );
 24 +  }
 25 +  void putGlobalVar(String name, Type type) throws ItemAlreadyExistsException{
 26 +    SymbolTable.top.put(
 27 +            new SymbolTableGlobalVariableItem(
 28 +                new Variable(name, type),
 29 +                SymbolTable.top.getOffset(Register.GP)
 30 +            )
 31 +        );
 32 +  }
 33 + }
```

```
  3  34    program:
  4     -    (actor | NL)*
     35 +    {beginScope();}(actor | NL)*{endScope();}
  5  36      ;
  6  37
  7  38      actor:
  8     -    'actor' ID '<' CONST_NUM '>' NL
     39 +    'actor' name=ID '<' mailboxSize=CONST_NUM '>' NL
     40 +      {
     41 +      try{
     42 +        Tools.putActor($name.text,Integer.parseInt($mailboxSize.text));
     43 +      }
     44 +      catch(ItemAlreadyExistsException e) {
     45 +            print("Exists");
     46 +        }
     47 +      finally{beginScope(); }
     48 +      }
  9  49      (state | receiver | NL)*
 10  50      'end' (NL | EOF)
     51 +    {endScope();}
 11  52      ;
 12     -
 13  53    state:
 14     -    type ID (',' ID)* NL
     54 +    type name = ID (',' ID)* NL
     55 +      {
     56 +      try{
     57 +        putGlobalVar($name.text,$type.return_type);
     58 +      }
     59 +      catch(ItemAlreadyExistsException e) {
     60 +            print("Exists");
     61 +        }
     62 +    }
 15  63      ;
 16  64
 17  65    receiver:
```

@@ -20,9 +68,9 @@ receiver:

```
20  68        'end' NL
21  69      ;
22  70
23      - type:
24      -    'char' ('[' CONST_NUM ']')*
25      -  | 'int' ('[' CONST_NUM ']')*
    71  + type returns [Type return_type]:
    72  +    'char' ('[' CONST_NUM ']')* {$return_type = CharType.getInstance();}
    73  +  | 'int' ('[' CONST_NUM ']')*  {$return_type = IntType.getInstance();}
26  74      ;
27  75
28  76    block:
```

@@ -48,7 +96,7 @@ statement:

```
48  96      ;
49  97
50  98    stm_vardef:
51      -    type ID ('=' expr)? (',' ID ('=' expr)?)* NL
    99  +    type ID ('=' expr)? (',' ID ('=' expr)?)* NL
52  100      ;
53  101
54  102    stm_tell:
```

26 ▮▮▮▮▮ CharType.java

@@ -0,0 +1,26 @@

```java
1  + public class CharType extends Type {
2  +
3  +   public int size() {
4  +     return Type.WORD_BYTES / 4;
5  +   }
6  +
7  +   @Override
8  +   public boolean equals(Object other) {
9  +     if (other instanceof CharType)
10 +       return true;
11 +     return false;
12 +   }
13 +
14 +   @Override
15 +   public String toString() {
16 +     return "char";
17 +   }
18 +
19 +   private static CharType instance;
20 +
21 +   public static CharType getInstance() {
22 +     if (instance == null)
23 +       return instance = new CharType();
24 +     return instance;
25 +   }
26 + }
```

17 ▮▮▮▮▮ Receiver.java

```
@@ -0,0 +1,17 @@
public class Actor {
    public Receiver(String name, ArrayList<Type> argumentTypes) {
        this.name = name;
        this.argumentTypes = argumentTypes;
    }

    public String getName() {
        return name;
    }

    public ArrayList<Type> getArgumentTypes() {
        return argumentTypes;
    }

    String name;
    ArrayList<Type> argumentTypes;
}
```

18 ▮▮▮▮▮ SymbolTableActorItem.java

```
@@ -0,0 +1,18 @@
public class SymbolTableActorItem extends SymbolTableItem {
    public SymbolTableActorItem(Actor actor, int offset) {
        this.actor = actor;
        this.offset = offset;
    }

    @Override
    public String getKey() {
        return actor.getName();
    }

    public int getOffset() {
        return offset;
    }

    Actor actor;
    int offset;
}
```

16 ▮▮▮▮▮ SymbolTableGlobalVariableItem.java

```
@@ -0,0 +1,16 @@
public class SymbolTableGlobalVariableItem extends SymbolTableVariableItemBase {

    public SymbolTableGlobalVariableItem(Variable variable, int offset) {
        super(variable, offset);
    }

    @Override
    public Register getBaseRegister() {
        return Register.GP;
    }

    @Override
    public boolean useMustBeComesAfterDef() {
        return true;
    }
}
```

18 ▮▮▮▮▮ SymbolTableReceiverItem.java

```
...    ...    @@ -0,0 +1,18 @@
         1    + public class SymbolTableReceiverItem extends SymbolTableItem {
         2    +   public SymbolTableReceiverItem(Receiver receiver, int offset) {
         3    +     this.receiver = receiver;
         4    +     this.offset = offset;
         5    +   }
         6    +
         7    +   @Override
         8    +   public String getKey() {
         9    +     return receiver.getName();
        10    +   }
        11    +
        12    +   public int getOffset() {
        13    +     return offset;
        14    +   }
        15    +
        16    +   Receiver receiver;
        17    +   int offset;
        18    + } ⊘↵
```

∨ 7 ▮▮▮▮▮ Tools.java

```
...    ...    @@ -0,0 +1,7 @@
         1    + public class Tools {
         2    +
         3    +   static public void putActor(String name, int mailboxSize) throws ItemAlreadyExistsException {
         4    +     SymbolTable.top.put(new SymbolTableActorItem(new Actor(name, mailboxSize), SymbolTable.top.getOffset(Register.GP)))
         5    +   }
         6    +
         7    + } ⊘↵
```

0 comments on commit `16233d1`

Please sign in to comment.