

Exame EPOCA Normal 2007-2008

1) Nome e Parâmetros de Construtor

IASTProgram, IASTCommand, IASTExpression
 Infix, Cur-MoveVel, IntVel, ClosureFunVel, PairVel

Implements IASTProgram

IASTProgram (List<String> vars, IASTCommand com);

IASTProgram (List<String> vars);

Implements IASTCommand

IASTAssign (String id, IASTExpression exp)

IASTSeq (IASTCommand Left, IASTCommand Right)

Implements IASTExpression

IASTid (String id)

IASTNum (int num)

IASTadd (IASTExpression left, IASTExpression right)

IASTFun (List<String> ids, List<String> vars, IASTCommand com, IASTExpression exp)

2 construtores de IASTFunction

IASTApp (String id, List<IASTExpression> exps)

IASTPair (IASTExpression first, IASTExpression second)

IASTSelectFirst (IASTExpression Pair)

IASTSelectSecond (IASTExpression Pair)

2) Public interface IASTProgram {
 Ival^{or val?} evaluate (IEnv env);
 }
 Public interface IASTCommand {
 Ival^{or val?} evaluate (IEnv env);
 }
 Public interface IASTExpression {
 Ival evaluate (IEnv env);
 }

//AST Program

```
IVal evaluate (IEnv env) {
    IEnv newEnv = env.BeginScope();
    for (Iteration <String> var = vars.iterator(); var.hasNext(); )
    {
        newEnv.Assoc (var.next(), null);
    }
    if (com != null) com.evaluate (newEnv);
    newEnv.EndScope();
    return noneVal();
}
```

//AST Assign

```
IVal evaluate (IEnv env) {
    env.Assoc (id, exp.evaluate (env));
    return noneVal();
}
```

//AST Seq

```
IVal evaluate (IEnv env) {
    left.evaluate (env);
    right.evaluate (env);
    return noneVal();
}
```

//AST id

```
IVal evaluate (IEnv env) {
    return env.find (id);
}
```

//AST num

```
IVal evaluate (IEnv env) {
    return new IntVal (num);
}
```

✓ AST add

```

Ival evaluate (Ienv env) {
  Ival l = left.evaluate (env);
  Ival R = right.evaluate (env);
  return new Ival (l.get() + R.get());
}
  
```

✓ // AST Fun

```

Ival evaluate (Ienv env) {
  return new ClosureFunVal (ids, vars, cum, exp, env);
}
  
```

✓ // AST APP

```

Ival evaluate (Ienv env) {
  ClosureFunVal cfv = (ClosureFunVal) env.find (id);
  Ienv envLoc = cfv.getEnv().beginScope();
  for (int i = 0; i < vars.size(); i++)
    envLoc.assoc (cfv.getIds().get(i), envPs.get(i).evaluate (env));
  IASTCommand com = cfv.getCum();
  if (com != null) com.evaluate (envLoc);
  Ival result = cfv.getExp().evaluate (envLoc);
  envLoc.endScope();
  return result;
}
  
```

✓ // AST Pair

```

Ival evaluate (Ienv env) {
  return new PairVal (first, second);
}
  
```

✓ // AST select first

```

Ival evaluate (Ienv env) {
  PairVal P = (PairVal) Pair.evaluate (env);
  return P.get first();
}
  
```

✓ // AST select second

```

PairVal P = (PairVal) Pair.evaluate (env);
return P.get second();
  
```