Đã bắt đầu vào lúc	Tuesday, 22 February 2022, 4:32 PM
Tình trạng	Đã hoàn thành
Hoàn thành vào lúc	Monday, 28 February 2022, 8:42 PM
Thời gian thực hiện	6 ngày 4 giờ
Điểm	5,00/5,00
Điểm	**10,00** của 10,00 (**100%**)

Given the grammar of MP as follows:

program: vardecls EOF;

vardecls: vardecl vardecltail;

vardecltail: vardecl vardecltail | ;

vardecl: mptype ids ';' ;

mptype: INTTYPE | FLOATTYPE;

ids: ID ',' ids | ID;

INTTYPE: 'int';

FLOATTYPE: 'float';

ID: [a-z]+ ;

Please copy the following class into your answer and modify the bodies of its methods to return the height of the parse tree? Your code starts at line 10.

class TerminalCount(MPVisitor):

  def visitProgram(self,ctx:MPParser.ProgramContext):

    return None

  def visitVardecls(self,ctx:MPParser.VardeclsContext):

    return None

  def visitVardecltail(self,ctx:MPParser.VardecltailContext):

    return None

  def visitVardecl(self,ctx:MPParser.VardeclContext):

    return None

  def visitMptype(self,ctx:MPParser.MptypeContext):

    return None

  def visitIds(self,ctx:MPParser.IdsContext):

    return None

Answer: (penalty regime: 0 %)

```
1  class TerminalCount(MPVisitor):
2      def visitProgram(self,ctx:MPParser.ProgramContext):
3          return 1 + self.visit(ctx.vardecls())
4
5      def visitVardecls(self,ctx:MPParser.VardeclsContext):
6          return 1 + max(self.visit(ctx.vardecl()), self.visit(ctx.vardecl
7
8      def visitVardecltail(self,ctx:MPParser.VardecltailContext):
9          if ctx.vardecl():
10             return 1 + max(self.visit(ctx.vardecl()), self.visit(ctx.var
11         else:
12             return 0
13
14     def visitVardecl(self,ctx:MPParser.VardeclContext):
15         return 1 + max(self.visit(ctx.mptype()), self.visit(ctx.ids()))
16
17     def visitMptype(self,ctx:MPParser.MptypeContext):
18         return 1
19
20     def visitIds(self,ctx:MPParser.IdsContext):
```

```
21    if ctx.ids():
22        return 1 + self.visit(ctx.ids())
23
```

	Test	Expected	Got	
✔	`"int a;"`	4	4	✔
✔	`"""int a,b;"""`	5	5	✔
✔	`"int a;float b;"`	5	5	✔
✔	`"int a,b;float c;"`	5	5	✔
✔	`"int a,b;float c,d,e;"`	7	7	✔

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Given the grammar of MP as follows:

program: vardecls EOF;

vardecls: vardecl vardecltail;

vardecltail: vardecl vardecltail | ;

vardecl: mptype ids ';' ;

mptype: INTTYPE | FLOATTYPE;

ids: ID ',' ids | ID;

INTTYPE: 'int';

FLOATTYPE: 'float';

ID: [a-z]+ ;

Please copy the following class into your answer and modify the bodies of its methods to count the internal nodes in the parse tree?

class ASTGeneration(MPVisitor):

  def visitProgram(self,ctx:MPParser.ProgramContext):

    return None

  def visitVardecls(self,ctx:MPParser.VardeclsContext):

    return None

  def visitVardecltail(self,ctx:MPParser.VardecltailContext):

    return None

  def visitVardecl(self,ctx:MPParser.VardeclContext):

    return None

  def visitMptype(self,ctx:MPParser.MptypeContext):

    return None

  def visitIds(self,ctx:MPParser.IdsContext):

    return None

Answer: (penalty regime: 0 %)

```
1  class ASTGeneration(MPVisitor):
2      def visitProgram(self,ctx:MPParser.ProgramContext):
3          return 1 + self.visit(ctx.vardecls())
4
5      def visitVardecls(self,ctx:MPParser.VardeclsContext):
6          return 1 + self.visit(ctx.vardecl()) + self.visit(ctx.vardecltai
7
8      def visitVardecltail(self,ctx:MPParser.VardecltailContext):
9          if ctx.vardecl():
10             return 1 + self.visit(ctx.vardecl()) + self.visit(ctx.vardec
11         else:
12             return 1
13     def visitVardecl(self,ctx:MPParser.VardeclContext):
14         return 1 + self.visit(ctx.mptype()) + self.visit(ctx.ids())
15
16     def visitMptype(self,ctx:MPParser.MptypeContext):
17         return 1
18
19     def visitIds(self,ctx:MPParser.IdsContext):
20         if ctx.ids():
21             return 1 + self.visit(ctx.ids())
22         else:
```

```
22 ▾          else:
23
```

	Test	Expected	Got	
✔	`"int a;"`	6	6	✔
✔	`"""int a,b;"""`	7	7	✔
✔	`"int a;float b;"`	10	10	✔
✔	`"int a,b;float c;"`	11	11	✔
✔	`"int a,b;float c,d,e;"`	13	13	✔

Passed all tests! ✔

Given the grammar of MP as follows:

program: vardecl+ EOF;

vardecl: mptype ids ';' ;

mptype: INTTYPE | FLOATTYPE;

ids: ID (',' ID)*;

INTTYPE: 'int';

FLOATTYPE: 'float';

ID: [a-z]+ ;

and AST classes as follows:

class Program:#decl:list(VarDecl)

class Type(ABC): pass

class IntType(Type): pass

class FloatType(Type): pass

class VarDecl: #variable:Id; varType: Type

class Id: #name:str

Please copy the following class into your answer and modify the bodies of its methods to generate the AST of a MP input?

class ASTGeneration(MPVisitor):

  def visitProgram(self,ctx:MPParser.ProgramContext):

    return None

  def visitVardecl(self,ctx:MPParser.VardeclContext):

    return None

  def visitMptype(self,ctx:MPParser.MptypeContext):

    return None

  def visitIds(self,ctx:MPParser.IdsContext):

    return None

Answer: (penalty regime: 0 %)

```
 1  def flatten(lst):
 2      if not isinstance(lst, list):
 3          return [lst]
 4      if len(lst) == 0:
 5          return []
 6      if len(lst) == 1:
 7          return flatten(lst[0])
 8
 9      return flatten(lst[0]) + flatten(lst[1:])
10
11
12  class ASTGeneration(MPVisitor):
13
14      def visitProgram(self,ctx:MPParser.ProgramContext):
15          return Program(flatten([self.visit(x) for x in ctx.vardecl()]))
16
17      def visitVardecl(self,ctx:MPParser.VardeclContext):
18          typee = self.visit(ctx.mptype())
19          ids = self.visit(ctx.ids())
20          return [VarDecl(x, typee) for x in ids]
```

```
21
22    def visitMptype(self,ctx:MPParser.MptypeContext):
23        return IntType() if ctx.INTTYPE() else FloatType()
```

	Test	Expected
✔	`"int a;"`	`Program([VarDecl(Id(a),IntType)])`
✔	`"""int a,b;"""`	`Program([VarDecl(Id(a),IntType),VarDecl(Id(b),IntType)])`
✔	`"int a;float b;"`	`Program([VarDecl(Id(a),IntType),VarDecl(Id(b),FloatType)])`
✔	`"int a,b;float c;"`	`Program([VarDecl(Id(a),IntType),VarDecl(Id(b),IntType),VarDecl(Id(c),FloatType)])`
✔	`"int a,b;float c,d,e;"`	`Program([VarDecl(Id(a),IntType),VarDecl(Id(b),IntType),VarDecl(Id(c),FloatType),VarDecl(Id(d),FloatType),VarDecl(`

Passed all tests! ✔
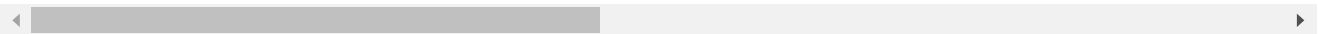
Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Given the grammar of MP as follows:

program: mptype EOF;

arraytype: primtype dimens ;

mptype: primtype | arraytype;

primtype: INTTYPE | FLOATTYPE;

dimens: dimen+;

dimen: '[' num '..' num ']';

num: '-'? INTLIT;

INTLIT: [0-9]+ ;

INTTYPE: 'integer';

FLOATTYPE: 'real';

and AST classes as follows:

class Type():abstract

class CompoundType(Type):abstract

class UnionType(CompoundType):#firstType:Type,secondType:primType

class ArrayType(CompoundType):#indexType:Type,eleType:primType

class PrimType(Type):abstract

class IntType(PrimType): pass

class FloatType(PrimType): pass

class RangeType(PrimType): #lowbound:int; highbound:int

class Id: #name:str

Please copy the following class into your answer and modify the bodies of its methods to generate the AST of a MP input?

class ASTGeneration(MPVisitor):

   def visitProgram(self,ctx:MPParser.ProgramContext):

     return None

   def visitMptype(self,ctx:MPParser.MptypeContext):

     return None

   def visitArraytype(self,ctx:MPParser.ArraytypeContext):

     return None

   def visitPrimtype(self,ctx:MPParser.PrimtypeContext):

     return None

   def visitDimens(self,ctx:MPParser.DimensContext):

     return None

   def visitDimen(self,ctx:MPParser.DimenContext):

     return None

   def visitNum(self,ctx:MPParser.DimenContext):

     return None

For example:

Test	Result
"integer[1..3]"	ArrayType(RangeType(1,3),IntType)

Answer: (penalty regime: 0 %)

```
1  class ASTGeneration(MPVisitor):
2
3      def visitProgram(self,ctx:MPParser.ProgramContext):
4          return self.visit(ctx.mptype())
5
6      def visitMptype(self,ctx:MPParser.MptypeContext):
7          if ctx.primtype():
8              return self.visit(ctx.primtype())
9          else:
10             return self.visit(ctx.arraytype())
11
12     def visitArraytype(self,ctx:MPParser.ArraytypeContext):
13         eleType = self.visit(ctx.primtype())
14         idxType = self.visit(ctx.dimens())
15         return ArrayType(idxType, eleType)
16
17     def visitPrimtype(self,ctx:MPParser.PrimtypeContext):
18         if ctx.INTTYPE():
19             return IntType()
20         return FloatType()
21
22     def visitDimens(self,ctx:MPParser.DimensContext):
23
```

	Test	Expected	Got
✔	"real [-3..0] [-10..-1]"	ArrayType(UnionType(RangeType(-3,0),RangeType(-10,-1)),FloatType)	ArrayType(U
✔	"integer[1..3]"	ArrayType(RangeType(1,3),IntType)	ArrayType(R
✔	"""integer [1..100] [-5..20] [100..3000]"""	ArrayType(UnionType(UnionType(RangeType(1,100),RangeType(-5,20)),RangeType(100,3000)),IntType)	ArrayType(U

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Given the grammar of MP as follows:

program: mptype EOF;

arraytype: primtype dimen | arraytype dimen ;

mptype: primtype | arraytype;

primtype: INTTYPE | FLOATTYPE;

dimen: '[' num '..' num ']';

num: '-'? INTLIT;

INTLIT: [0-9]+ ;

INTTYPE: 'integer';

FLOATTYPE: 'real';

and AST classes as follows:

class Type():abstract

class FloatType(PrimType): pass

class RangeType(PrimType): #lowbound:int; highbound:int

Please copy the following class into your answer and modify the bodies of its methods to generate the AST of a MP input?

```
class ASTGeneration(MPVisitor):

    def visitProgram(self,ctx:MPParser.ProgramContext):

        return None

    def visitMptype(self,ctx:MPParser.MptypeContext):

        return None

    def visitArraytype(self,ctx:MPParser.ArraytypeContext):

        return None

    def visitPrimtype(self,ctx:MPParser.PrimtypeContext):

        return None

    def visitDimen(self,ctx:MPParser.DimenContext):

        return None

    def visitNum(self,ctx:MPParser.DimenContext):

        return None
```

For example:

Test	Result
"integer[1..3]"	ArrayType(RangeType(1,3),IntType)

Answer: (penalty regime: 0 %)

```python
1  class ASTGeneration(MPVisitor):
2      def visitProgram(self,ctx:MPParser.ProgramContext):
3          return self.visit(ctx.mptype())
4
5      def visitMptype(self,ctx:MPParser.MptypeContext):
6          if ctx.primtype():
7              return self.visit(ctx.primtype())
8          prim, dimens = self.visit(ctx.arraytype())
9          dimens = reduce(lambda x, y: UnionType(x, self.visit(y)), dimens
10         return ArrayType(dimens, prim)
11
12     def visitArraytype(self,ctx:MPParser.ArraytypeContext):
13         if ctx.primtype():
14             return self.visit(ctx.primtype()), [ctx.dimen()]
15         prim, dimens = self.visit(ctx.arraytype())
16         dimens += [ctx.dimen()]
17         return prim, dimens
18
19     def visitPrimtype(self,ctx:MPParser.PrimtypeContext):
20         if ctx.INTTYPE():
21             return IntType()
22         return FloatType()
23
```

	Test	Expected	Got
✔	"real [-3..0] [-10..-1]"	ArrayType(UnionType(RangeType(-3,0),RangeType(-10,-1)),FloatType)	ArrayType(U
✔	"integer[1..3]"	ArrayType(RangeType(1,3),IntType)	ArrayType(F
✔	"""integer [1..100] [-5..20] [100..3000]"""	ArrayType(UnionType(UnionType(RangeType(1,100),RangeType(-5,20)),RangeType(100,3000)),IntType)	ArrayType(U

Passed all tests! ✔

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

◄ AST Quiz CC03

Chuyển tới...

Slides ►