

Caneca++

programa ::= cabecalho classe (cabecalho classe | cabecalho interface)*

cabecalho ::= pacote importacao

pacote ::= "pacote" identificadorDePacote ";"

importacao ::= (("importe" identificadorDePacote | "importe" identificadorDeTipo) ";")*

classe ::= assinaturaDeClasse (interfaceDaClasse)? "inicio" corpoDaClasse "fim"

assinaturaDeClasse ::= "classe" modificadorDeAcessoFeminino
identificadorDeClasse

interfaceDaClasse ::= "implementa" listaDeIdentificadoresDeInterface

corpoDaClasse ::= (constante)* (atributo)* (construtor)+ (metodo)*

interface ::= "interface" modificadorDeAcessoFeminino
identificadorDeInterface "inicio" corpoDaInterface "fim"

corpoDaInterface ::= (constante)* (assinaturaDeMetodo ";")*

Caneca++

metodo ::= assinaturaDeMetodo "inicio" (instrucao)* "fim"

assinaturaDeMetodo ::= "metodo" modificadorDeAcessoMasculino identificadorDeTipo
identificador "(" listaDeArgumentos ")"

construtor ::= assinaturaDeConstrutor "inicio" (instrucao)* "fim"

assinaturaDeConstrutor ::= "construtor" modificadorDeAcessoMasculino
identificadorDeClasse "(" listaDeArgumentos ")"

atributo ::= "atributo" modificadorDeAcessoMasculino identificadorDeTipo identificador ";"

constante ::= "constante" modificadorDeAcessoFeminino identificadorDeTipo atribuicao ";"

modificadorDeAcessoMasculino ::= "publico" | "privado" | "protegido"

modificadorDeAcessoFeminino ::= "publica" | "privada" | "protegida"

listaDeParametros ::= (valor)? ("," valor)*

listaDeArgumentos ::= (identificadorDeTipo identificador)? ("," identificadorDeTipo
identificador)*

listaDeIdentificadoresDeInterface ::= identificadorDeInterface (","
identificadorDeInterface)*

Caneca++

declaracaoDeVariavel ::= identificadorDeTipo identificador

declaracaoDeArranjo ::= identificadorDeTipo ("[" constanteInteira "]")+
identificador

atribuicao ::= (esse ".")? identificador ("[" constanteInteira "]")* "="
valor

chamadaDeMetodo ::= (identificador | esse) "."

chamadaDeMetodoLocal

chamadaDeMetodoLocal ::= identificador "(" listaDeParametros ")"

instanciacao ::= novo identificadorDeClasse "(" listaDeParametros ")"

novo ::= "novo" | "nova"

esse ::= "esse" | "essa"

retorno ::= "retornar" valor

Caneca++

operacao ::= operacaoAritmetica | operacaoLogica

operacaoAritmetica ::= valor operadorAritmetico valor | valor
operadorAritmetico operacaoAritmetica | "-" operacaoAritmetica

operacaoLogica ::= valor operadorLogico valor | valor operadorLogico

operacaoLogica | valorBooleano | "~" operacaoLogica

operadorAritmetico ::= "+" | "-" | "*" | "/" | "%"

operadorLogico ::= ">" | "<" | "==" | "!=" | ">=" | "<="

condicional ::= se | se senao

se ::= "se" "(" valor ")" "inicio" (instrucao)* "fim"

senao ::= "senao" "inicio" (instrucao)* "fim"

para ::= "para" "(" condicaoPara ")" "inicio" (instrucao)* "fim"

condicaoPara ::= atribuicao ";" valor ";" operacaoAritmetica

enquanto ::= "enquanto" "(" valor ")" "inicio" (instrucao)* "fim"

Caneca++

valor ::=

valorBooleano

| constanteInteira

| constanteDecimal

| constanteReal

| literalCaractere

| literalTexto

| literalArranjo

| (esse ".")? identificador ("[" constanteInteira "]")*

| esse

| chamadaDeMetodo

| chamadaDeMetodoLocal

| instanciacao

| operacao

| "(" valor ")"

Caneca++

instrucao ::=

valor ";"

| declaracaoDeVariavel ";"

| declaracaoDeArranjo ";"

| atribuicao ";"

| retorno ";"

| condicional

| para

| enquanto

Caneca++

identificadorDeTipo ::= identificadorDeClasse |
identificadorDeInterface

identificadorDeClasse ::= (identificadorDePacote ".")? identificador

identificadorDeInterface ::= (identificadorDePacote ".")? identificador

identificadorDePacote ::= identificador ("." identificador)*

identificador ::= [a-Z] ([a-Z] | [0-9] | "_")*

valorBooleano ::= "verdadeiro" | "falso"

constanteInteira ::= "0" | ((" "-")? [1-9] ([0-9])*)

constanteReal ::= (" "-")? ("0" [1-9]* | [1-9] [0-9]*) "." ([0-9])+

literalCaractere ::= "" ([a-Z] | [0-9] | " " | "\n" | "\t" | "\s" | "\r") ""

literalTexto ::= "" (literalCaractere)* ""

literalArranjo ::= "[" (valor ("," valor)*)? "]"

Caneca++

```
pacote caminho.qualquer;
```

```
importe biblioteca.qualquer.uma.ClasseDaBiblioteca;
```

```
Importe biblioteca.qualquer.uma.pacoteInteiroDaBiblioteca;
```

```
classe publica MinhaClasse implementa InterfaceQualquer inicio
```

```
    constate publica REAL PI = 3.14;
```

```
    atributo privado Objeto meuAtributo;
```

```
    construtor publico MinhaClasse() inicio
```

```
        esse.meuAtributo = novo Objeto("valorQualquer");
```

```
        Inteiro[3] arranjoNadaFaz = [1, 2, 3];
```

```
        modulo(0);
```

```
    fim
```

```
    metodo publico Inteiro modulo(Inteiro valorQualquer) inicio
```

```
        Inteiro modulo;
```

```
        se (valorQualquer >= 0) inicio
```

```
            modulo = valorQualquer;
```

```
        fim senao inicio
```

```
            modulo = -valorQualquer;
```

```
        fim
```

```
        retorne modulo;
```

```
    fim
```

```
fim
```


Caneca++

Grafos de Sintaxe:

<http://railroad.my28msec.com/rr/ui>