

# GELLO Class File BNF

```
GelloPackage ::= PackageStatement

PackageStatement ::=
    <PACKAGE> NameOrLiteral ImportsStatement? PackageElement*
<ENDPACKAGE>

ImportsStatement ::=
    <IMPORTS> ImportName ( "," ImportName)*

ImportName ::=
    NameOrLiteral

PackageElement ::= PackageStatement | ClassDefinition

ClassDefinition ::=
    <CLASS> ClassDefName (<EXTENDS> ClassRef)? ClassMember* |
    <CLASS> ClassDefName <EQUAL> ClassRef |
    <CLASS> ClassName EnumerationType

ClassDefName ::=
    NameOrLiteral |
    NameOrLiteral "(" ClassRefList ")" |
    NameOrLiteral "<" NameList ">"

ClassRefList ::= ClassRef ( "," ClassRef)*

NameList ::= Name ( Name " ," )*

ClassRef ::=
    ClassName |
    ClassName "(" ClassRef ")" |
    <TUPLE>

ClassMember ::=
    NameOrLiteral ( "(" ClassFormalParams? ")"
)? ":" ClassRef

ClassFormalParams ::=
    ClassFormalParam ( "," ClassFormalParam)*

ClassFormalParam ::=
    Name ":" ClassRef

EnumerationType ::=
    <ENUM> "(" (EnumLiteral ( "," EnumLiteral )*) ")"

EnumLiteral ::=
    NameOrLiteral

NameOrLiteral ::=
    Name | <STRING_LITERAL>

ClassName ::=
    NameWithPath | <STRING_LITERAL>

NameWithPath ::=
    Name ( "::" Name)*

Name ::=
    <ID> ( "." <ID>)*

<STRING_LITERAL: ('\"' ( ~[ '\"' , "\n", "\r" ])* '\"' | \"' ( ~[ '\"' ,
"\n", "\r" ])* \"' ) >

<ID: [ "a"- "z", "A"- "Z" ] ( [ "a"- "z", "A"- "Z", "0"- "9" ] | "_" ( [ "a"- "z", "A"- "Z", "
0"- "9" ] )+ ) * >

<ENUM: "Enum">

<TUPLE: "Tuple" >

<EQUAL: "=" >

<ENDPACKAGE: "endPackage" | "EndPackage" | "endpackage">

<CLASS: "class" | "Class">
```

```
<EXTENDS: "extends" >
```

```
<IMPORTS: "Imports" | "imports">
```

```
<PACKAGE: "Package" | "package">
```