Bricks Style Grading API

```
/******************************************************************/
/* 1. Style Grading for Declaration and Use of Variable           */
/*    a. numDecVars(ast)               --> integer >= 0           */
/*    b. numUndecVars(ast)             --> integer >= 0           */
/*    c. numVarsUsed(ast)              --> integer >= 0           */
/*    d. numVarsInFuncsUseGloVars(ast) --> integer >= 0           */
/*    e. isAnyFuncVar(ast)             --> boolean ? true:false   */
/******************************************************************/
```

```
Properties in a pnut object correspond to each function:
       nDV    : numDecVars(ast),
       nUDV   : numUndecVars(ast),
       nVU    : numVarsUsed(ast),
       nVFUGV : numVarsInFuncsUseGloVars(ast),
       isFV   : isAnyFuncVar(ast),
```

```
How to retrieve each corresponding value from a pnut object:
       var style       = pnut.collectStructureStyleFacts(ast);
       var style_nDV   = style.nDV;
       var style_nUDV  = style.nUDV;
       var style_nVU   = style.nVU;
       var style_nVFUGV= style.nVFUGV;
       var style_isFV  = style.isFV;
```

int numDecVars(ast):	calculate total number of declared variables in a program
int numUndecVars(ast):	calculate total number of undeclared variables that get used in a program
int numVarsUsed(ast):	calculate total number of variables used in a program
int numVarsInFuncsUseGloVars(ast):	calculate the number of variables in functions that uses global declared variables
bool isAnyFuncVar(ast):	exam if any function gets assigned to a variable in global level

```
/******************************************************************/
/* 2. Style Grading for Declaration and Use of Array              */
/*    a. numDecArrs(ast)      --> integer >= 0                    */
/*    b. numUndecArrs(ast)    --> integer >= 0                    */
/*    c. numArrsUsed(ast)     --> integer >= 0                    */
/******************************************************************/
```

```
Properties in a pnut object correspond to each function:
       nDA      : numDecArrs(ast),
       nUDA     : numUndecArrs(ast),
       nAU      : numArrsUsed(ast),
```

```
How to retrieve each corresponding value from a pnut object:
       var style       = pnut.collectStructureStyleFacts(ast);
       var style_nDA   = style.nDA;
       var style_nUDA  = style.nUDA;
       var style_nAU   = style.nAU;
```

int numDecArrs(ast):	calculate total number of declared arrays in a program
int numUndecArrs(ast):	calculate total number of undeclared arrays that get used in a program

int numArrsUsed(ast):	calculate total number of arrays that are used in a program

```
/******************************************************************/
/* 3. Style Grading for Declaration and Use of Object             */
/*    a. numDecObjs(ast)                 --> integer >= 0         */
/*    b. numUndecObjs(ast)               --> integer >= 0         */
/*    c. numObjsUsed(ast)                --> integer >= 0         */
/*    d. isAnyFuncBoundToAFuncRtnObj(ast)--> boolean ? true:false */
/******************************************************************/
```

```
Properties in a pnut object correspond to each function:
       nDO      : numDecObjs(ast),
       nUDO     : numUndecObjs(ast),
       nOU      : numObjsUsed(ast),
       isFBAFRO : isAnyFuncBoundToAFuncRtnObj(ast),
```

```
How to retrieve each corresponding value from a pnut object:
       var style          = pnut.collectStructureStyleFacts(ast);
       var style_nDO      = style.nDO;
       var style_nUDO     = style.nUDO;
       var style_nOU      = style.nOU;
       var style_isFBAFRO = style.isFBAFRO;
```

int numDecObjs(ast):	calculate total number of declared objects in a program
int numUndecObjs(ast):	calculate total number of undeclared objects that get used in a program
int numObjsUsed(ast):	calculate total number of objects that are used in a program
bool isAnyFuncBoundToAFuncRtnObj(ast):	identify if any function return object is bound to a function

```
/******************************************************************/
/* 4. Style Grading for Use of While Loop                         */
/*    a. numWhileLoopsInGloLev(ast)       --> integer >= 0        */
/*    b. numNestedWhileLoopsInGloLev(ast) --> integer >= 0        */
/*    c. numWhileLoopsInFuncs(ast)        --> integer >= 0        */
/*    d. numNestedWhileLoopsInFuncs(ast) --> integer >= 0         */
/*    e. numWhileLoopsInAProgram(ast)     --> integer >= 0        */
/******************************************************************/
```

```
Properties in a pnut object correspond to each function:
       nWLGL    : numWhileLoopsInGloLev(ast),
       nNWLGL   : numNestedWhileLoopsInGloLev(ast),
       nWLF     : numWhileLoopsInFuncs(ast),
       nNWLF    : numNestedWhileLoopsInFuncs(ast),
       nWLAP    : numWhileLoopsInAProgram(ast),
```

```
How to retrieve each corresponding value from a pnut object:
       var style       = pnut.collectStructureStyleFacts(ast);
       var style_nWLGL = style.nWLGL;
       var style_nNWLGL= style.nNWLGL;
       var style_nWLF  = style.nWLF;
       var style_nNWLF = style.nNWLF;
       var style_nWLAP = style.nWLAP;
```

int numWhileLoopsInGloLev(ast):	calculate total number of while loops in global level
int numNestedWhileLoopsInGloLev(ast):	calculate total number of nested while loops in global level

int numWhileLoopsInFuncs(ast):	calculate total number of while loops in functions (local level)
int numNestedWhileLoopsInFuncs(ast):	calculate total number of nested while loops in functions (local)
int numWhileLoopsInAProgram(ast):	calculate total number of while loops in a program

```
/******************************************************************/
/* 5. Style Grading for Use of For Loop                           */
/*    a. numForLoopsInGloLev(ast)            --> integer >= 0     */
/*    b. numNestedForLoopsInGloLev(ast)      --> integer >= 0     */
/*    c. numForLoopsInFuncs(ast)             --> integer >= 0     */
/*    d. numNestedForLoopsInFuncs(ast)       --> integer >= 0     */
/*    e. numForLoopsInAProgram(ast)          --> integer >= 0     */
/******************************************************************/
```

```
Properties in a pnut object correspond to each function:
       nFLGL    : numForLoopsInGloLev(ast),
       nNFLGL   : numNestedForLoopsInGloLev(ast),
       nFLF     : numForLoopsInFuncs(ast),
       nNFLF    : numNestedForLoopsInFuncs(ast),
       nFLAP    : numForLoopsInAProgram(ast),
```

```
How to retrieve each corresponding value from a pnut object:
       var style       = pnut.collectStructureStyleFacts(ast);
       var style_nFLGL = style.nFLGL;
       var style_nNFLGL= style.nNFLGL;
       var style_nFLF  = style.nFLF;
       var style_nNFLF = style.nNFLF;
       var style_nFLAP = style.nFLAP;
```

int numForLoopsInGloLev(ast):	calculate total number of for loops in global level
int numNestedForLoopsInGloLev(ast):	calculate total number of nested for loops in global level
int numForLoopsInFuncs(ast):	calculate total number of for loops in functions (local level)
int numNestedForLoopsInFuncs(ast):	calculate total number of nested for loops in functions (local)
int numForLoopsInAProgram(ast):	calculate total number of for loops in a program

```
/******************************************************************/
/* 6. Style Grading for Declaration and Use of Function           */
/*    a. numDecFuncs(ast)               --> integer >= 0          */
/*    b. areCallExpsAllValid(ast)       --> boolean ? true:false  */
/*    c. areDecFuncsCalled(ast)         --> boolean ? true:false  */
/*    d. areDecFuncsCalledOnce(ast)     --> boolean ? true:false  */
/*    e. isAnyDecFuncPassedByRef(ast)   --> boolean ? true:false  */
/*    f. isAnyFuncReturnObj(ast)        --> boolean ? true:false  */
/******************************************************************/
```

```
Properties in a pnut object correspond to each function:
       nDF      : numDecFuncs(ast),
       areCEAV  : areCallExpsAllValid(ast),
       areDFC   : areDecFuncsCalled(ast),
       areDFCO  : areDecFuncsCalledOnce(ast),
       isADFPBR : isAnyDecFuncPassedByRef(ast),
       isAFRO   : isAnyFuncReturnObj(ast),
How to retrieve each corresponding value from a pnut object:
       var style         = pnut.collectStructureStyleFacts(ast);
       var style_nDF     = style.nDF;
       var style_areCEAV = style.areCEAV;
       var style_areDFC  = style.areDFC;
       var style_areDFCO = style.areDFCO;
       var style_isADFPBR= style.isADFPBR;
       var style_isAFRO  = style.isAFRO;
```

int numDecFuncs(ast):	calculate the number of declared functions in global level
bool areCallExpsAllValid(ast):	exam call expressions that all call declared functions in which functions are declared on the top of call expressions
bool areDecFuncsCalled(ast):	exam all declared functions get called in a program
bool areDecFuncsCalledOnce(ast):	exam all declared functions get called exactly once in a program
bool isAnyDecFuncPassedByRef(ast):	exam if any function is a pass-by-reference function or not
bool isAnyFuncReturnObj(ast):	identify if any function returns an object in a program

```
/******************************************************************/
/* 7. Style Grading for Recursive Function                        */
/*    a. isRecursiveFunction(ast)  --> boolean ? true:false       */
/******************************************************************/
```

```
Properties in a pnut object correspond to each function:
       isRF     : isRecursiveFunction(ast)
```

```
How to retrieve each corresponding value from a pnut object:
       var style      = pnut.collectStructureStyleFacts(ast);
       var style_isRF = style.isRF;
```

bool isRecursiveFunction(ast):	exam if a function is recursive or not by checking its return statement