## paceval\_cRegisteredObject

- # paceval\_cCleanupHandler
   \* handle\_CleanupHandler
- # unsigned long registerPosition
- + paceval\_cRegisteredObject (paceval\_cCleanupHandler \*handle\_CleanupHandler\_in)
- + ~paceval cRegisteredObject()
- + void setRegisterPosition (unsigned long registerPosition\_in)
- + unsigned long getRegister Position()

## paceval\_cBaseAtomicGraphNode

- # bool \* handle\_stackHasCache OptionAvailable
- # paceval\_sNodeSpecificData
   \* handle sNodeSpecificData
- + paceval\_cBaseAtomicGraph Node(paceval\_cCleanupHandler \*handle\_CleanupHandler\_in)
- + void initiateData(const char \*operator\_in, long valueNode1\_in, long valueNode2 \_in, long resultNode\_in, long position\_in, const char \*valueOperator\_in)
- + ~paceval\_cBaseAtomicGraph Node()
- + paceval\_eListOfPointerTypes getPointerType()
- + long getPosition()
- + paceval\_eOperatorTypes
   getOperator()
- + long \* getValueField1()
- + long \* getValueField2()
- + long \* getResultField()
- + bool hasCacheOptionAvailable (unsigned long stackNumber\_in)
- + bool CreateLookaheadCache Data()
- + void setLevelMultithread Jump(long levelMultithreadJump\_in)
- + bool hasLevelMultithread
  Jump(long \*levelMultithreadJump\_out)
- + void setValueLevelMultithread
  (unsigned long idSingleCalculation
  \_in, const paceval\_eCalculationPrecision
  Types useCalculationPrecision\_in, long
  double valueLevelMultithreadAsLongDouble
  \_in, double valueLevelMultithreadAsDouble
  \_in, float valueLevelMultithreadAsFloat\_in,
  bool hasTrustedLevelMultithreadMinMaxResult
  \_in, long double valueLevelMultithreadMinValue
  \_in, long double valueLevelMultithreadMaxValue in)
- + bool hasValueLevelMultithread
  (unsigned long idSingleCalculation
  \_in, const paceval\_eCalculationPrecision
  Types useCalculationPrecision\_in, long
  double \*valueLevelMultithreadAsLongDouble
  \_out, double \*valueLevelMultithreadAsDouble
  \_out, float \*valueLevelMultithreadAsFloat\_out,
  bool \*hasTrustedLevelMultithreadMinMaxResult
  \_out, long double \*valueLevelMultithreadMinValue
  out, long double \*valueLevelMultithreadMaxValue\_out)
- + void setZeroCachingJump (long zeroCachingJump\_in)
- + bool hasZeroCachingJump (long \*zeroCachingJump\_out)

## paceval\_cAtomicGraphNode Operation

- + long lockedInnerCacheFor StackNumber
- # bool outerCachedLinkedResult Available
- # long outerCachedLinkedResult
- # bool \* handle\_InnerCached ResultAvailable
- # void \*\* handle\_InnerCachedData
- + paceval\_cAtomicGraphNodeOperation(paceval\_cCleanupHandler \*handle\_CleanupHandler\_in)
- void initializeDataOperation (const char \*operator\_in, long valueNode1\_in, long valueNode2\_in, long resultNode \_in, long position\_in)
- + ~paceval\_cAtomicGraphNode Operation()

+ void setOuterCachedLinked

- Result(long outerCachedLinkedResult\_in)
  + bool hasOuterCachedResult
- (long \*outerCachedLinkedResult\_out)
- bool updateInnerCachedResult
   (const unsigned long stackNumber
   \_in, const paceval\_eCalculationPrecision
   Types useCalculationPrecision\_in, const
   void \*handle\_value1\_in, const void \*handle
   \_value2\_in, const void \*handle\_result\_in,
   const bool hasTrustedMinMaxValues\_in, const
   long double \*trustedResult\_in, const long double
   \*trustedMinValue\_in, const long double \*trustedMaxValue\_in)
- + bool hasInnerCachedResult
  (const unsigned long stackNumber
  \_in, const paceval\_eCalculationPrecision
  Types useCalculationPrecision\_in, const
  void \*handle\_value1\_in, const void \*handle
  \_value2\_in, long double \*resultAsLongDouble
  \_out, double \*resultAsDouble\_out, float \*resultAsFloat
  - \_out, double \*resultAsDouble\_out, float \*resultAsFloat \_out, bool \*hasTrustedMinMaxValues\_out, long double \*trustedResult\_out, long double \*trustedMinValue\_out, long double \*trustedMaxValue out)
- # void initiateData(const char \*operator\_in, long valueNode1\_in, long valueNode2 \_in, long resultNode\_in, long

position\_in, const char \*valueOperator\_in)

## paceval\_cAtomicGraphNode AddValue

- + paceval\_cAtomicGraphNode
   AddValue(paceval\_cCleanupHandler
   \*handle\_CleanupHandler\_in)
   + void initializeDataAddVariable
- (const char \*operator\_in,
  long valueNode1\_in, long position
  \_in, long resultNode\_in)
- + void initializeDataAddValue (const char \*operator\_in, long resultNode\_in, long position\_in, const char \*valueOperator\_in)