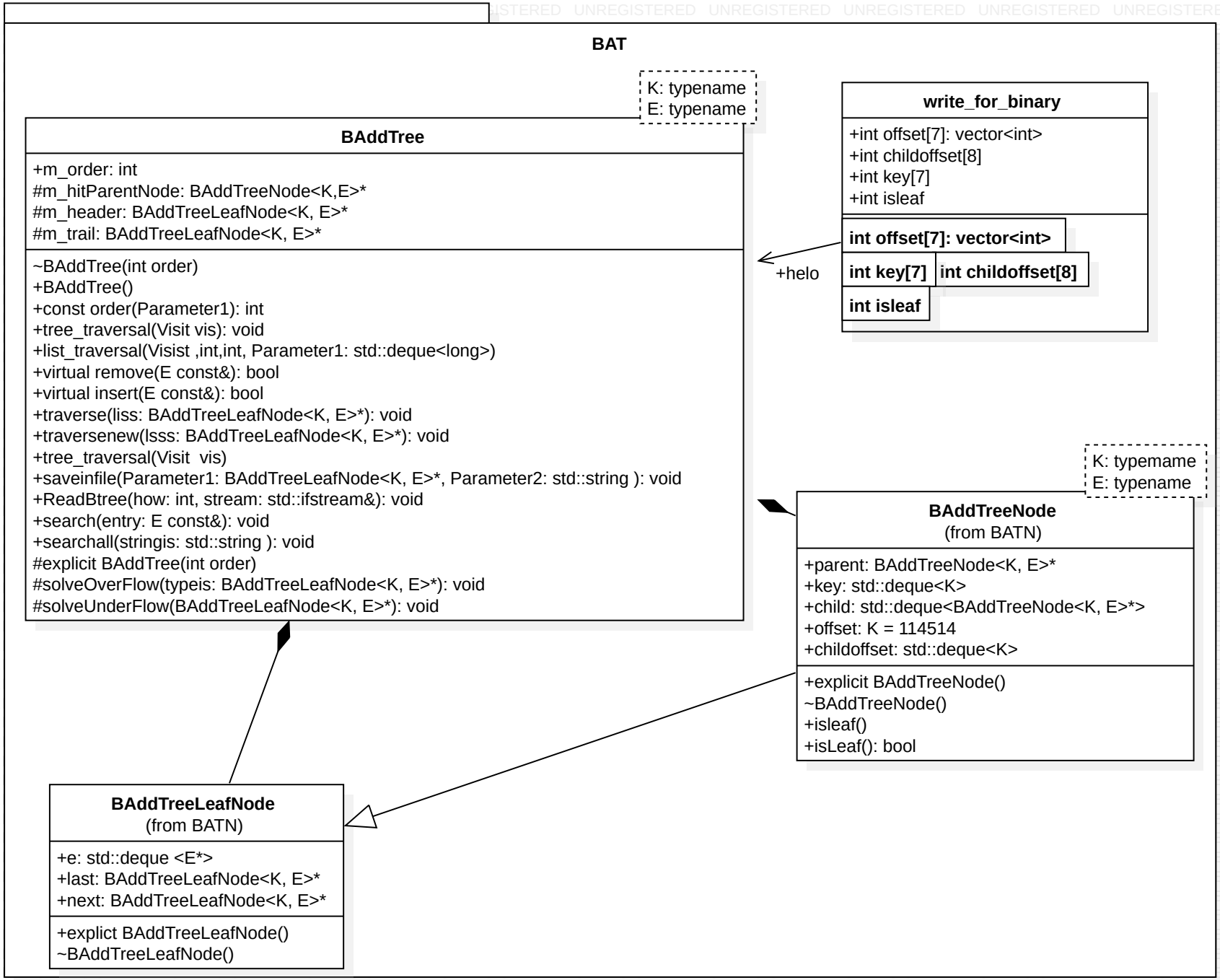
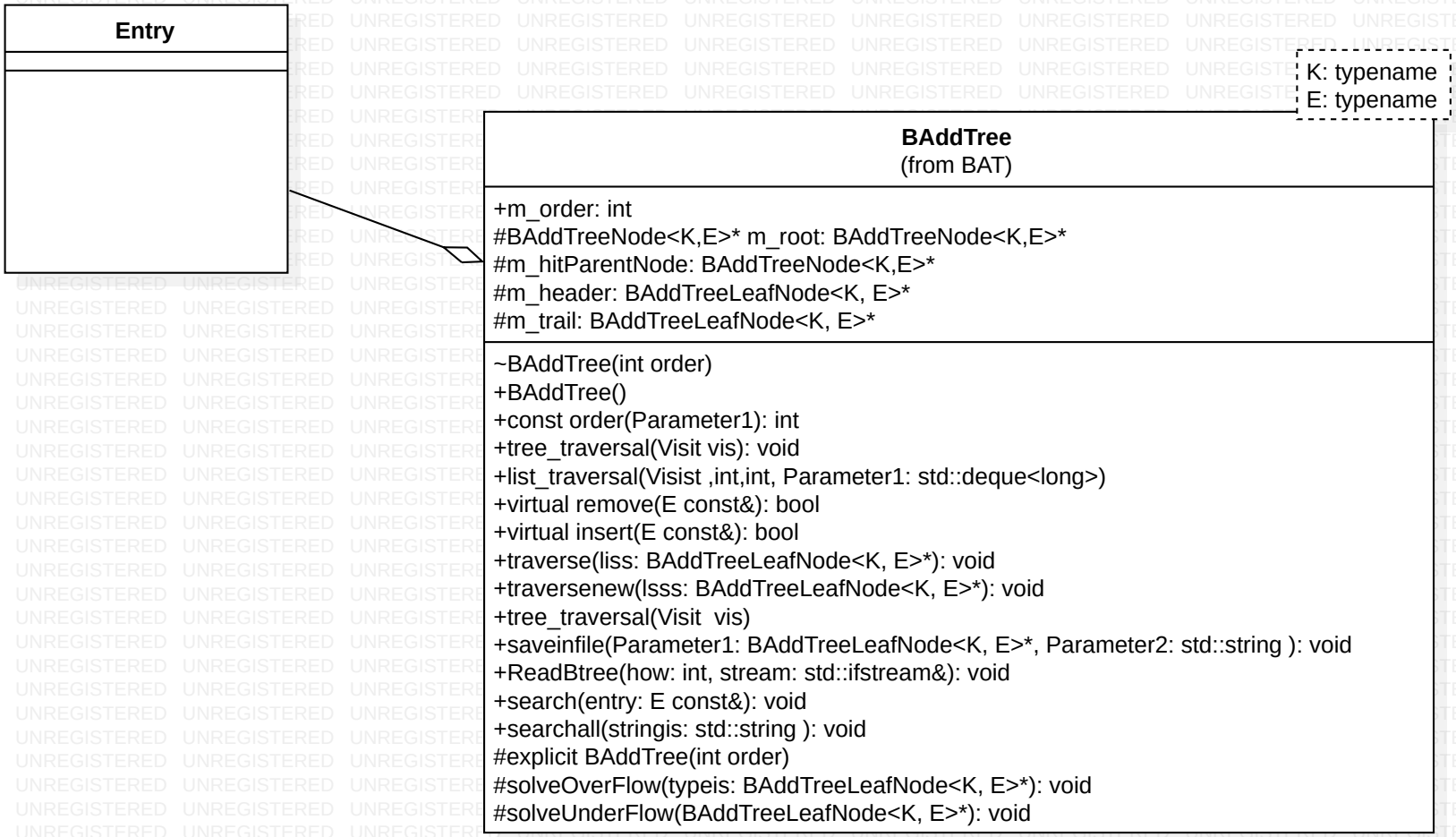
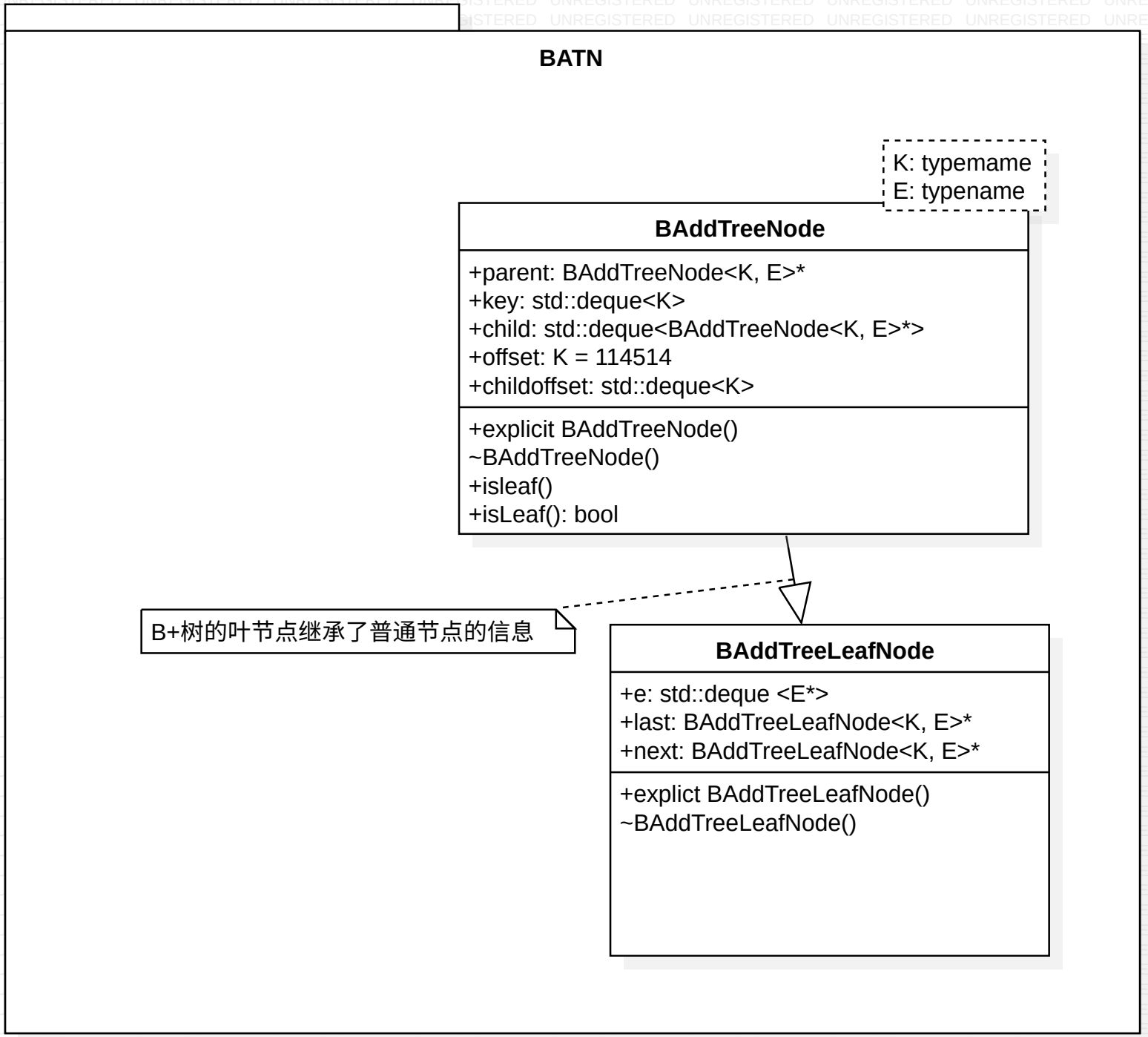
Model::BAddTree



Model::BAddTreeFactory



Model::BAddTreeNode



Model::Table

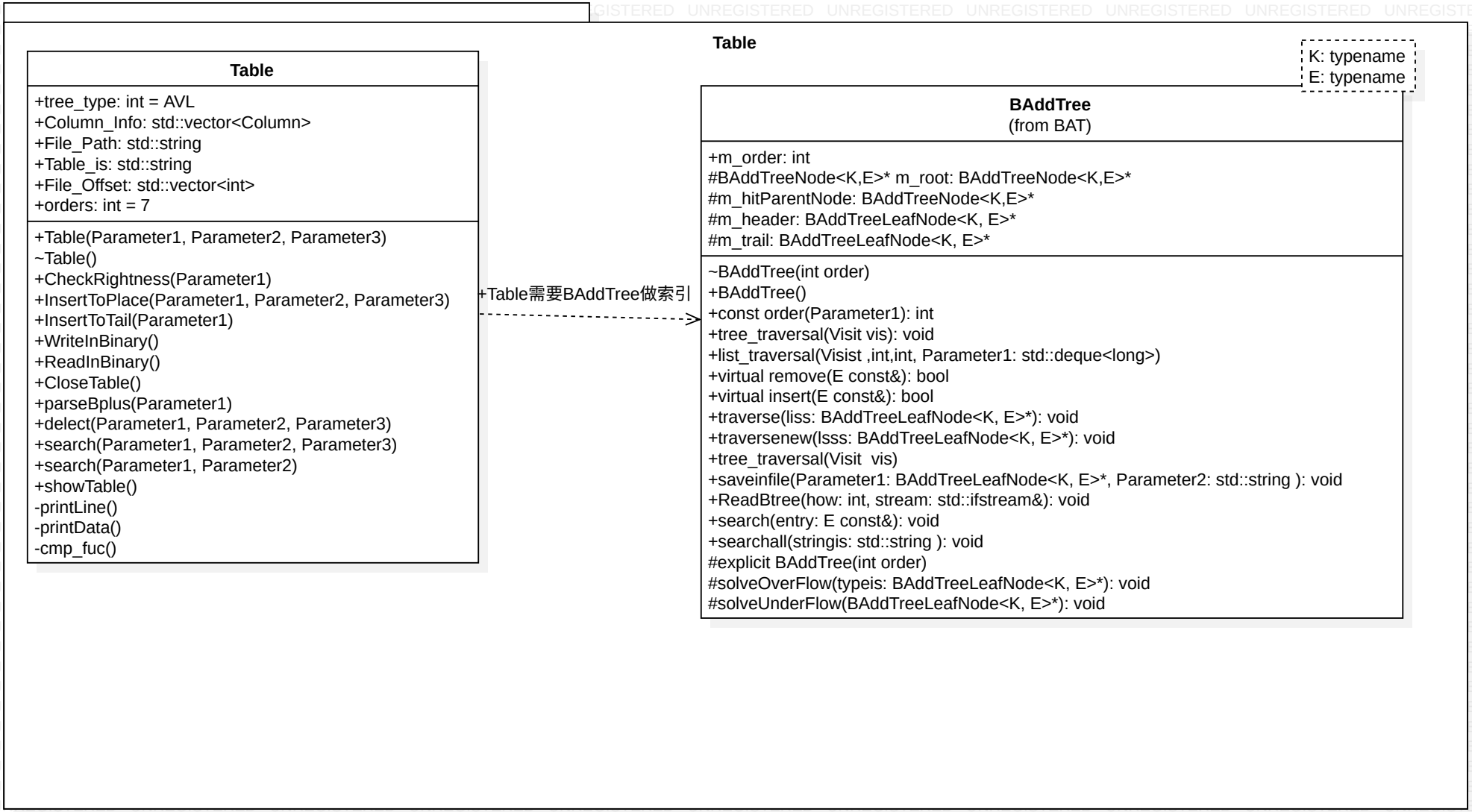
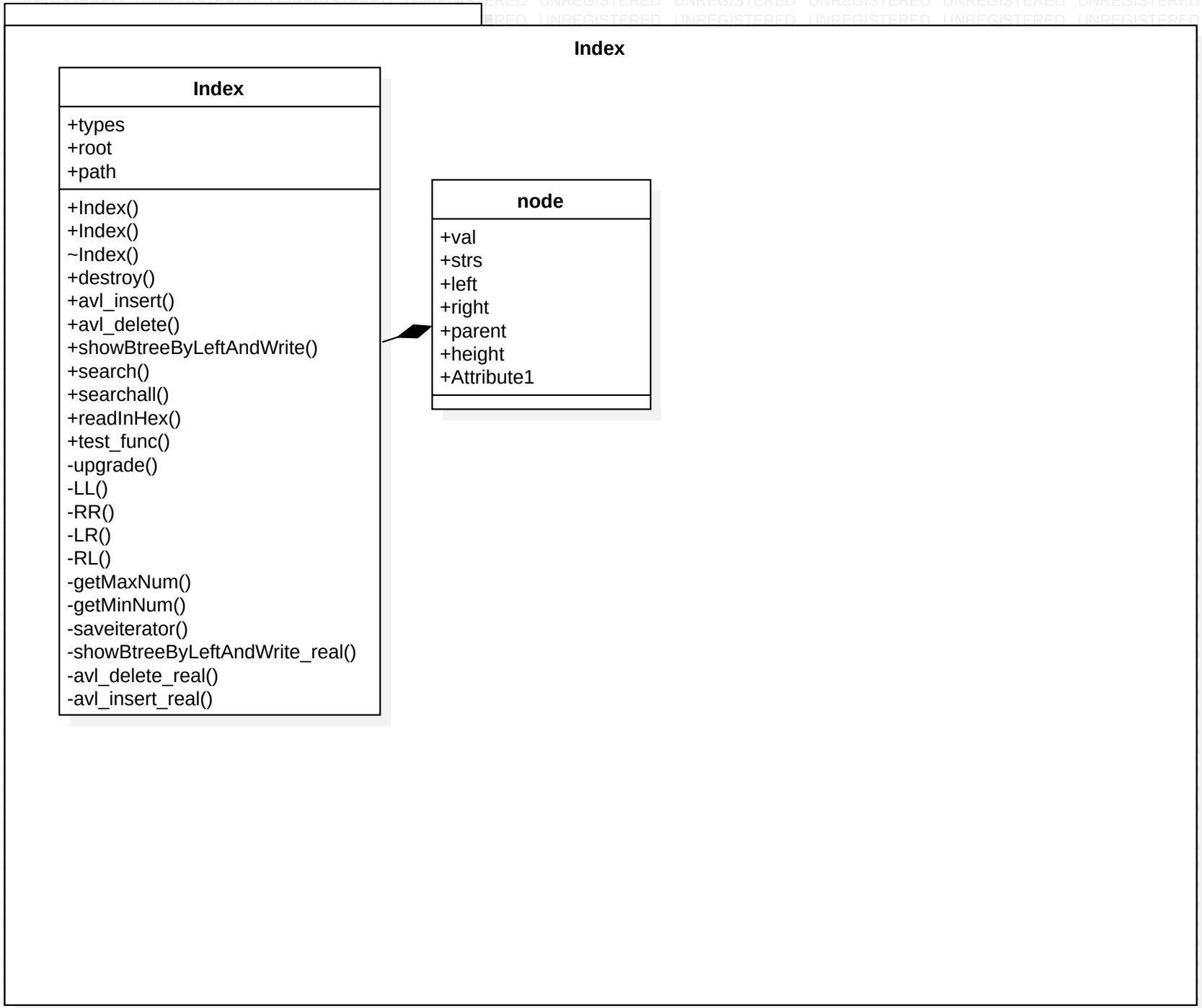


Table
Table
+tree_type: int = AVL
+Column_Info: std::vector<Column>
+File_Path: std::string
+Table_is: std::string
+File_Offset: std::vector<int>
+orders: int = 7
+Table(Parameter1, Parameter2, Parameter3)
~Table()
+CheckRightness(Parameter1)
+InsertToPlace(Parameter1, Parameter2, Parameter3)
+InsertToTail(Parameter1)
+WriteInBinary()
+ReadInBinary()
+CloseTable()
+parseBplus(Parameter1)
+delect(Parameter1, Parameter2, Parameter3)
+search(Parameter1, Parameter2, Parameter3)
+search(Parameter1, Parameter2)
+showTable()
-printLine()
-printData()
-cmp_fuc()
+Table需要BAddTree做索引
K: typename
E: typename
BAddTree
(from BAT)
+m_order: int
#BAddTreeNode<K,E>* m_root: BAddTreeNode<K,E>*
#m_hitParentNode: BAddTreeNode<K,E>*
#m_header: BAddTreeLeafNode<K, E>*
#m_trail: BAddTreeLeafNode<K, E>*
~BAddTree(int order)
+BAddTree()
+const order(Parameter1): int
+tree_traversal(Visit vis): void
+list_traversal(Visist ,int,int, Parameter1: std::deque<long>)
+virtual remove(E const&): bool
+virtual insert(E const&): bool
+traverse(liss: BAddTreeLeafNode<K, E>*): void
+traversenew(lsss: BAddTreeLeafNode<K, E>*): void
+tree_traversal(Visit vis)
+saveinfile(Parameter1: BAddTreeLeafNode<K, E>*, Parameter2: std::string): void
+ReadBtree(how: int, stream: std::ifstream&): void
+search(entry: E const&): void
+searchall(stringis: std::string): void
#explicit BAddTree(int order)
#solveOverFlow(typeis: BAddTreeLeafNode<K, E>*): void
#solveUnderFlow(BAddTreeLeafNode<K, E>*): void

Model::Index


Index
Index
+types
+root
+path
+Index()
+Index()
~Index()
+destroy()
+avl_insert()
+avl_delete()
+showBtreeByLeftAndWrite()
+search()
+searchall()
+readInHex()
+test_func()
-upgrade()
-LL()
-RR()
-LR()
-RL()
-getMaxNum()
-getMinNum()
-saveiterator()
-showBtreeByLeftAndWrite_real()
-avl_delete_real()
-avl_insert_real()
node
+val
+strs
+left
+right
+parent
+height
+Attribute1