## • public string Root public ASType Typepublic List Children public void AddChild(ASTree t)public double Evaluate(Dictionary m) public string Serialize(IExporter exporter) public override string ToString() Calculateur • public static void memorizeVar(string expr) public enum ASType

**ASTree** 

	public interface lExporter
public string Export(ASTree tree)	
	public class RPN : IExporter
public string Export(ASTree tree)	
	public class TextExporter : IExporter
public string Export(ASTree tree)	

Index  public static void Main(string[] args) public static void Main(string[] args) public static void Display Tests(ASTree t, Dictionary m, ReversePolishNotation rpn, TextExporter te, LispExporter li)  public string Export(ASTreo tree)  Parser  protected string [] tokens protected string [] tokens protected Stack int memory protected Stack int memory protected Stack int forestMemory protected Stack int forestMemory public static ASTree LastExpression  public static ASTree LastExpression  public static bool Parse public bool ParseCmds public bool ParseCmds protected viol next Note protected viol recall	public class LispExporter
public static void Main(string[] args) public static void DisplayTests(ASTree t, Dictionary m, ReversePolishNotation rpn, TextExporter te, LispExporter li)  Parser  protected string [] tokens protected string [] tokens protected string one memory protected boil debug protected Stack int memory protected Stack int forestMemory protected Stack ASTree forest protected Stack int forestMemory public static bool Parse public static bool Parse public void d protected void nextToken protected void memorize protected void rexall protected bool leass	public string Export(ASTree tree)
Parser  protected string [] tokens protected string memory protected Stack int memory protected Stack int forestMemory protected Stack int forestMemory protected Stack and Free LastExpression  protected Stack and Free LastExpression  public static ASTree LastExpression  public static bool Parse public void d protected string token protected string and tokens protected stack int forestMemory public static bool Parse public static bool Parse public void d protected string token protected void memorize protected void memorize protected void mexiToken protected void mexiToken protected void mexit protected void me	Index
Parser  • protected string [] tokens • protected int pos • protected Stack int memory • protected Stack STree forest • protected Stack ASTree forest • protected Stack int forestMemory • public static ASTree LastExpression  • public static bool Parse • public bool ParseCmds • public void d • protected string token • protected void nextToken • protected void memorize • protected void recall • protected void recall	<ul> <li>public static void Main(string[] args)</li> <li>public static void DisplayTests(ASTree t, Dictionary m, ReversePolishNotation rpn, TextExporter te, LispExporter li)</li> </ul>
<ul> <li>protected string [] tokens</li> <li>protected int pos</li> <li>protected Stack int memory</li> <li>protected bool debug</li> <li>protected Stack ASTree forest</li> <li>protected Stack int forestMemory</li> <li>public static ASTree LastExpression</li> </ul> public static bool Parse <ul> <li>public bool ParseCmds</li> <li>public void d</li> <li>protected string token</li> <li>protected void nextToken</li> <li>protected void memorize</li> <li>protected void recall</li> <li>protected bool eos</li> </ul>	public string Export(ASTree tree)
<ul> <li>public bool ParseCmds</li> <li>public void d</li> <li>protected string token</li> <li>protected void nextToken</li> <li>protected void memorize</li> <li>protected void recall</li> <li>protected bool eos</li> </ul>	<ul> <li>protected string [] tokens</li> <li>protected int pos</li> <li>protected Stack int memory</li> <li>protected bool debug</li> <li>protected Stack ASTree forest</li> <li>protected Stack int forestMemory</li> </ul>
	<ul> <li>public bool ParseCmds</li> <li>public void d</li> <li>protected string token</li> <li>protected void nextToken</li> <li>protected void memorize</li> <li>protected void recall</li> <li>protected bool eos</li> </ul>