

Simple_stat
<ul style="list-style-type: none"> <li>- mean: double</li> <li>- SD: double</li> <li>- min: double</li> <li>- max: double</li> <li>- sum: double</li> <li>- size: size</li> <li>- calculate(): void</li> <li>- check_unique(const T&amp; data): boo</li> <li>- calculate_mean(): double</li> <li>- calculate_SD(): double</li> <li>- calculate_min(): double</li> <li>- calculate_max(): double</li> <li>- unique_data: AList&lt;T&gt;*</li> <li>- num_repetitions: Alist&lt;int&gt;*</li> </ul>
<ul style="list-style-type: none"> <li>+ Simple_stat()</li> <li>+ ~Simple_stat()</li> <li>+ get_mean(): double</li> <li>+ get_SD(): double</li> <li>+ get_min(): double</li> <li>+ get_max(): double</li> <li>+ feed(C const&amp; container): void</li> <li>+ append(const T&amp; data): void</li> <li>+ search(T const&amp; data): std::pair&lt;int, int&gt;</li> <li>+ printAll(): void</li> <li>+ print_unique(): void</li> <li>+ print_num_repetitions(): void</li> <li>+ get_sum(): double</li> <li>+ getLength(): int</li> <li>+ getTotalLength(): size</li> <li>+ print_all_stats(): void</li> <li>+ removem(int&amp; data, int&amp; num): void</li> <li>+ empty(): void</li> <li>+ unique_set(): set&lt;T&gt;</li> <li>+ [](int index): T operator</li> </ul>



AList
<ul style="list-style-type: none"> <li>- maxSize: int</li> <li>- listSize: int</li> <li>- curr: int</li> <li>- listArray: E*</li> </ul>
<ul style="list-style-type: none"> <li>+ AList(int size = 20)</li> <li>+ ~AList()</li> <li>+ Insert(const E&amp; it): void</li> <li>+ append(const E&amp; it): void</li> <li>+ remove(): E</li> <li>+ moveToStart(): void</li> <li>+ moveToEnd(): void</li> <li>+ prev(): void</li> <li>+ next(): void</li> <li>+ length() const: int</li> <li>+ currPos() const: int</li> <li>+ moveToPos(int pos): void</li> <li>+ getValue() const: const E&amp;</li> <li>+ isAtEnd(): bool</li> <li>+ ssearch(const E&amp; it): bool</li> <li>+ ibsearch(const E&amp; search_this): bool</li> <li>+ rbsearch(const E&amp; search_this, int left, int right): bool</li> </ul>



List
<ul style="list-style-type: none"> <li>- =(const List&amp;): void operator</li> <li>- List(const List&amp;)</li> </ul>
<ul style="list-style-type: none"> <li>+ List()</li> <li>+ ~List(): virtual</li> <li>+ clear(): virtual void</li> <li>+ insert(const E&amp; item): virtual void</li> <li>+ append(const E&amp; item): virtual void</li> <li>+ remove(): virtual E</li> <li>+ moveToStart(): virtual void</li> <li>+ moveToEnd(): virtual void</li> <li>+ prev(): virtual void</li> <li>+ next(): virtual void</li> <li>+ length() const: virtual void</li> <li>+ currPos() const: virtual int</li> <li>+ moveToPos(int pos): virtual void</li> <li>+ getValue() const: virtual const E&amp;</li> <li>+ isAtEnd(): virtual bool</li> </ul>