

Currency
<pre># abbreviation: string # currencyNote : string # currencyNotePlural : string # currencyCoin : string # currencyCoinPlural : string # wholeParts : int # fractionalParts : int</pre>
<pre>+ Currency() : + Currency(abbr : string, note : string, notePlural string, coin : string, + coinPlural : string, whole : int, fractional : int) + ~Currency() : virtual + fractionsInWhole : static const int + empty() : void + isEmpty() : bool + isSameType(currency : const Currency) : bool + addWhole(a : const double) : void + addFractional(a : const int) : void + getCurrencyAbbreviation() const : string + getCurrencyNote() const : string + getCurrencyNotePlural() const : string + getCurrencyCoin() const : string + getCurrencyCoinPlural() const : string + setCurrencyAbbreviation(abbr : const string) : void + setCurrencyNote(note : const string) : void + setCurrencyNotePlural(notePlural : const string) : void + setCurrencyCoin(coin : const string) : void + setCurrencyCoinPlural(coinPlural : const tring) : void + getWholeParts() const : int + getFractionalParts() const : int + geTotalWholeParts() const : double + getTotalFractionaParts() const : int + setWholeParts(whole : const int) : void + setFractionalParts(fractional : const int) : void + setTotalFractionalParts(totalFractional : const int) : void + setTotalWholeParts(totalWhole : const double) : void</pre>

- + operator+ (a : const double) const : Currency
- + operator- (a : const double) const : Currency
- + operator+= (a : const double) : Currency&
- + operator-= (a : const double) : Currency&
- + operator+ (currency : const Currency&) const : Currency
- + operator- (currency : const Currency&) const : Currency
- + operator+= (currency : const Currency&) : Currency
- + operator-= (currency : const Currency&) : Currency
- + <<friend>>operator == (left : const Currency&, right : const Currency&) : bool
- + <<friend>>operator != (left : const Currency&, right : const Currency&) : bool
- + <<friend>>operator < (left : const Currency&, right : const Currency&) : bool
- + <<friend>>operator > (left : const Currency&, right : const Currency&) : bool
- + <<friend>>operator <= (left : const Currency&, right : const Currency&) : bool
- + <<friend>>operator >= (left : const Currency&, right : const Currency&) : bool
- + <<friend>>operator >> (is : istream&, currency : Currency&) : istream&
- + <<friend>>operator << (os: ostream&, currency : Currency&) : ostream&

Dollar
<ul style="list-style-type: none"> + Dollar (whole : int, fractional : int) : + ~Dollar() :

ExceptionCurrencyMismatch

+ what () const : const char*

ExceptionCurrencyNegative

+ what() const : const char*

List

- count : int - head* : Node<T>

+ List() : + ~List() : + isEmpty() : bool + getCount : int + insert(:T& , : int) : void + insertFirst(: T&) : void + insertLast(: T&) : void + remove(: int) : void + removeFirst() : void + removeLast() : void + removeAll() : void + getData(: int) : T& + getFirstData() : T& + getLastData() : T& + setData (: T&, : int) : void + setFirstData(: T&) : void + setLastData(: T&) : void

Node

+ data* : T

+ next* : Node<T>
+ Node() : + Node(: T&) : + Node(: T&, : Node<T>*) : + ~Node() :

Stack
+ push(: T&) : void + pop() : void + peek() : T& + empty() : void + count() : void