

Size : size_t
T : typename

CharArray

```
+ Append(T*, size_t) :void
+ CharArray()
+ CharArray(CharArray<T, OtherSize>&)
+ CharArray(T*)
+ CharArray(T*, size_t)
+ ~CharArray()
+ Clear() :void
+ Empty() :CharArray<T, Size>&
+ GetLength() :size_t {query}
+ GetPtr() :T* {query}
+ GetSize() :size_t {query}
+ operator!=(CharArray<T, OtherSize>&) :bool {query}
+ operator!=(T*) :bool {query}
+ operator[](size_t) :T {query}
+ operator[](size_t) :T&
+ operator+(CharArray<T, OtherSize>&) :CharArray<T, Size>
+ operator+(T*) :CharArray<T, Size>
+ operator+=(CharArray<T, OtherSize>&) :CharArray<T, Size>&
+ operator+=(T*) :CharArray<T, Size>&
+ operator+=(T) :CharArray<T, Size>&
+ operator=(CharArray<T, OtherSize>&) :CharArray<T, Size>&
+ operator==(CharArray<T, OtherSize>&) :bool {query}
+ operator==(T*) :bool {query}
+ Print(T*, ...) :void
+ SetLength(size_t) :void

«friend»
+ operator!=(T*, CharArray<T, Size>&) :bool
+ operator+(T*, CharArray<T, Size>&) :CharArray<T, Size>
+ operator==(T*, CharArray<T, Size>&) :bool
```

T : typename

CharList

```
+ Append(T*, size_t) :void
+ CharList(size_t)
+ CharList(CharList&)
+ CharList(T*)
+ CharList(T*, size_t)
+ ~CharList()
+ Clear() :void
+ Compact() :void
+ EnsureSize(size_t) :void
+ GetLength() :size_t {query}
+ GetPtr() :T* {query}
+ GetSize() :size_t {query}
+ operator!=(CharList&) :bool {query}
+ operator!=(CharArray<T, Size>&) :bool {query}
+ operator!=(T*) :bool {query}
+ operator[](size_t) :T {query}
+ operator[](size_t) :T&
+ operator<<(CharList&) :CharList&
+ operator<<(CharArray<T, Size>&) :CharList&
+ operator<<(T*) :CharList&
+ operator<<(bool) :CharList&
+ operator<<(T) :CharList&
+ operator<<(!32) :CharList&
+ operator<<(U32) :CharList&
+ operator<<(!64) :CharList&
+ operator<<(U64) :CharList&
+ operator<<(F32) :CharList&
+ operator<<(D64) :CharList&
+ operator=(CharList&) :CharList&
+ operator==(CharList&) :bool {query}
+ operator==(CharArray<T, Size>&) :bool {query}
+ operator==(T*) :bool {query}
+ Print(T*, ...) :void
+ Reserve(size_t) :void
+ SetLength(size_t) :void

«friend»
+ operator!=(CharArray<T, Size>&, CharList&) :bool
+ operator!=(T*, CharList<T>&) :bool
+ operator==(CharArray<T, Size>&, CharList&) :bool
+ operator==(T*, CharList<T>&) :bool
```