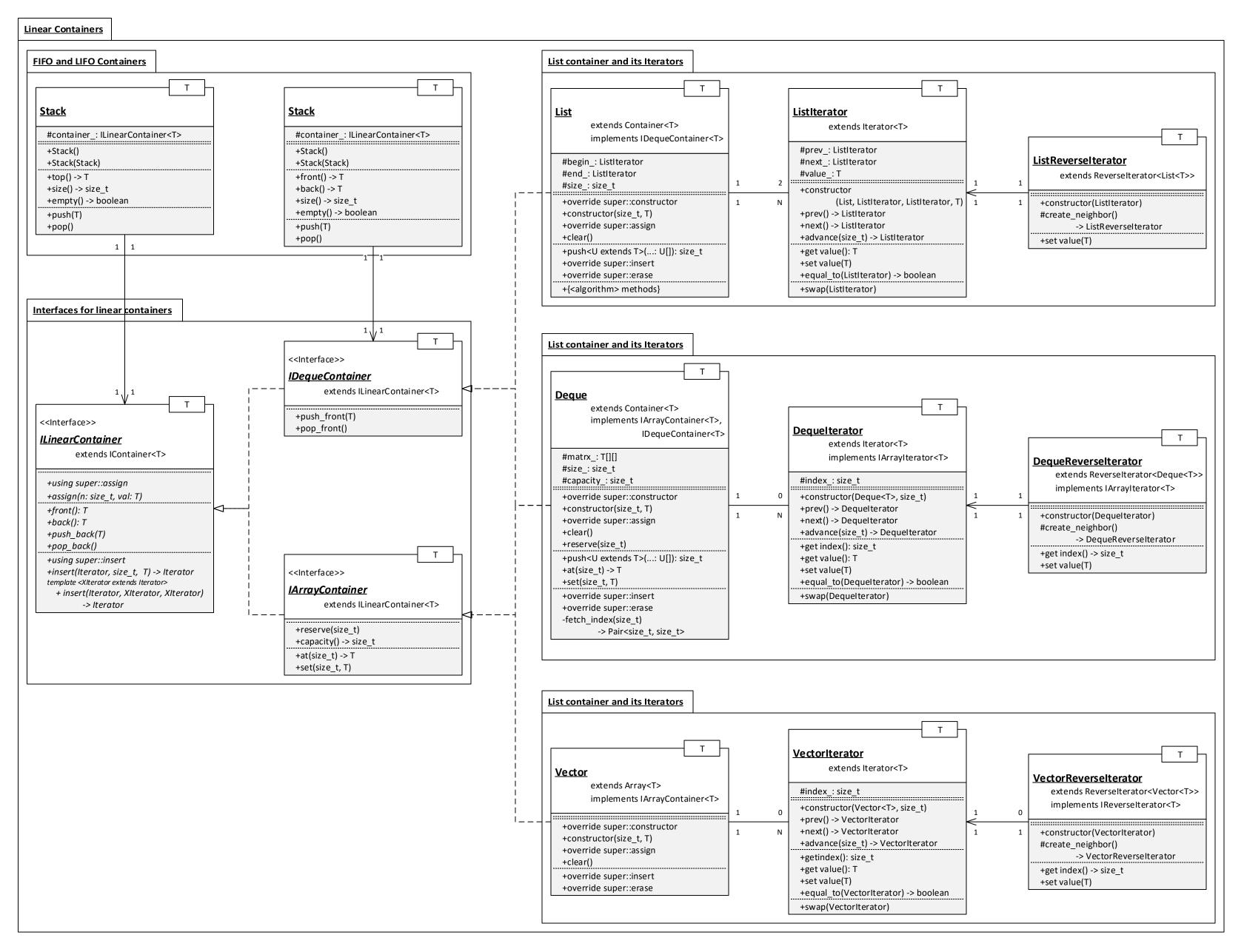
Containers outline Abstract Containers Т <<Interface>> *IContainer* **Linear Containers** Linear Containers template <XIterator extends Iterator> +assign(first: XIterator, last: XIterator) - Vector +clear() Container - Deque +begin() -> Iterator<T> implements IContainer<T> - List +end() -> Iterator<T> FIFO & LIFO Containers +rbegin() -> ReverseIterator<T> +constructor() - Queue +rend() -> Reverselterator<T> +constructor(Container) +size() -> number - Stack +constructor(Iterator, Iterator) +empty() -> boolean +clear() +push<U extends T>(...: U[]) -> number +insert(Iterator, T) -> Iterator<T> +erase(Iterator) -> Iterator<T> template <XIterator extends Iterator> +erase(Iterator, Iterator) -> Iterator +swp(IContainer) **Abstract Iterators Hashed & Tree-structured Containers** Container extends IContainer Hashed Containers 0 Ν **Reverselterator** HashSet - HasMap extends Container::Iterator **Iterator** - HashMultiSet #base : Container::iterator - HashMultiMap #source: IContainer<T> +constructor(Container::iterator) Tree-structured Containers +consturctor(IContainer) #create neighbor() -> Reverselterator - TreeSet +prev(): Iterator +base() -> Container::iterator +next(): Iterator - TreeMap +prev() -> ReverseIterator +advance(size t): Iterator - TreeMultiSet +next() -> Reverselterator +get value() -> T - TreeMultiMap +advance(size t) -> ReverseIterator +equal_to(Iterator) -> boolean PriorityQueue +get value() -> Container::value type +swap(Iterator) +equal to(Reverselterator) -> boolean +swap(Reverselterator) 1 1 1



+pop()

+hash() -> size_t

+swap(MapIterator)