```
DeepCopy
                                                        + {DESTRUCTIVE COPY = false}: enum
                                                         clone(ptr: T* const&): T*
     NoCopy
                                                         release(ptr: T* const&): bool
                                                         swap(policy: DeepCopy&): void
    + {DESTRUCTIVE COPY = false}: enum
      clone(ptr: T* const&): T*
                                                    DeepPrimitiveCopy
      release(ptr: T* const&): bool
      swap (policy: NoCopy&): void
                                                    + {DESTRUCTIVE COPY = false}: enum
                                                   # clone(ptr: T* const&): T*
                                                   # release(ptr: T* const&): bool
                                                    # swap(policy: DeepPrimitiveCopy&): void
                                                                                 T: typename
OwnershipPolicy
                                        NonIntrusiveReferenceCounting
DestructiveCopy
                                         reference count: size t*
                                        + {DESTRUCTIVE COPY = false}: enum
+ {DESTRUCTIVE COPY = true}: enum
                                        + NonIntrusiveReferenceCounting()
 clone(ptr: T* const& ): T*
 release(ptr: T* const&): bool
                                         clone(ptr: T* const& ): T*
  swap(policy: DestructiveCopy&): void
                                         release(ptr: T* const&): bool
                                          swap (policy: NonIntrusiveReferenceCounting&): void
```