GRelation

A C++ library for generic one-to-many and many-to-many object relations.

T. Glebe

Contents

1

template <class T> struct **G_trait**

Class configuration

`G_trait` allows to configure the template classes. As a default, the Relations are based on a STL `list` container. This allows fast addition and removal of object relations with the cost of no sorting and no possiblity for direct access. Per default, pointers to objects are stored in the STL container. The behaviour of the GRelation container and iterator can be configured by specializing the G_trait structure. For an example see HERA-B Note 00-188.

Author: T. Glebe

2

template <class T> struct **GR_trait**

Type promotion.

`GR_trait` is used for type promotion. This is needed in case you want to create a relation to a base class. Then just specialize `GR_trait` to return the base type of the class. Example:

```
class A : public BaseIf {
...
};

template <>
struct GR_trait<A> {
  typedef BaseIf val;
};

class B {
public:
  GRelationList<BaseIf> rel;

  GRelationList<BaseIf>& container(const Ba-
seIf& rhs) { return rel; }
...
};

int main() {
  A a;
  B b;

  createSingleRelation(b,a);
}
```

Author: T. Glebe

3

template <class L, class R> const bool **createRelation** (L& l, R& r)

Many-to-Many

Create bidirectional Relation between two Objects.

Author: T. Glebe

4

template <class L, class R> const bool **createRelation** (L& l, GRelationList<R>& rcont, R& r)

Many-to-Many

Create bidirectional Relation between two Objects, where one relation containers is given explicitly as an argument.

Author: T. Glebe

5

template <class L, class R> const bool **createRelation** (L& l, GRelationList<R>& rcont, R& r, GRelationList<L>& lcont)

Many-to-Many

Create bidirectional Relation between two Objects, where the relation containers are given explicitly as an argument.

Author: T. Glebe

6

template <class L, class R> const bool **createSingleRelation** (L& l, R& r)

One-to-Many

Create unidirectional Relation between two Objects.

Author: T. Glebe

7

template <class L, class R> const bool **createSingleRelation** (L& l, GRelationList<R>& rcont, R& r)

One-to-Many

Create unidirectional Relation between two Objects, where the relation container is given by an argument

Author: T. Glebe

8

template <class L, class R> const bool **removeRelation** (L& l, R& r)

Many-to-Many

Remove bidirectional Relation between two Objects.

Author: T. Glebe

9

template <class L, class R> const bool **removeRelation** (L& l, GRelationList<R>& rcont, R& r)

Many-to-Many

Remove bidirectional Relation between two Objects, where one container is given by an argument.

Author: T. Glebe

10

template <class L, class R> const bool **removeRelation** (L& l, GRelationList<R>& rcont, R& r, GRelationList<L>& lcont)

Many-to-Many

Remove bidirectional Relation between two Objects, where the containers are given by an argument.

Author: T. Glebe

11

template <class L, class R> const bool **removeSingleRelation** (L& l, R& r)

One-to-Many

Remove unidirectional Relation between two Objects.

Author: T. Glebe

12

template <class L, class R> const bool **removeSingleRelation** (L& l, GRelationList<R>& rcont, R& r)

One-to-Many

Remove unidirectional Relation between two Objects, where the container is given by an argument.

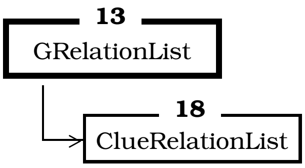
Author: T. Glebe

13

template <class T> class **GRelationList**

Container.

Inheritance



Public Members

— Typedef's —

typedef GRelationIterator <T>
iterator

typedef ConstGRelationIterator <T>
const_iterator

typedef ReverseGRelationIterator <T>
reverse_iterator

typedef ConstReverseGRelationIterator <T>
const_reverse_iterator

— Constructors —

GRelationList ()

GRelationList (const GRelationList<T>& rhs)

virtual ~**GRelationList** ()

— Access functions —

inline void
push_back (const Gvalue_type& rel)

inline Gpointer
back () const

inline int **size** () const

inline bool

empty () const

— Iterator methods —

inline iterator
begin ()

inline iterator
end ()

inline const_iterator
begin () const

inline const_iterator
end () const

inline reverse_iterator
rbegin ()

inline reverse_iterator
rend ()

inline const_reverse_iterator
rbegin () const

inline const_reverse_iterator
rend () const

Gpointer **operator[]** (unsigned int i) const
direct access

— Operators —

const GRelationList <T> &
operator= (const GRelationList<T>& rhs)

inline bool
operator== (const GRelationList<T>& rhs) const

— Expert functions —

void **print** (std::ostream& os) const

inline void
unbook (const Gvalue_type& rel)
get rid of relation

inline void
clear (void) *remove all relations without unbooking them*

template <class L> void
unbook_all (const L& lhs) const

Relation container class. As a default, the GRelationList container is mapped onto a STL `list` container. Therefore no direct access to relations is possible. The `operator[]()` emulates direct access. This is of course time intensive and should not be used if the size of the container is large. For other containers the `operator[]()` method can be overloaded.

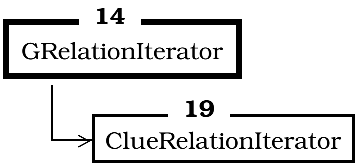
Author: T. Glebe

14

template <class T> class **GRelationIterator**

Iterator.

Inheritance



Public Members

— Constructors —

GRelationIterator ()

GRelationIterator (const GRelation_it& it)

virtual ~**GRelationIterator** ()

— Operators —

inline self&
operator++ ()

inline self&
operator– ()

inline Gpointer
operator* () const

inline Gpointer
operator-> () const

inline bool
operator== (const GRelationIterator<T>& rhs) const

inline bool
operator!= (const GRelationIterator<T>& rhs) const

Forward Relation Iterator. This is a STL compatible iterator for the

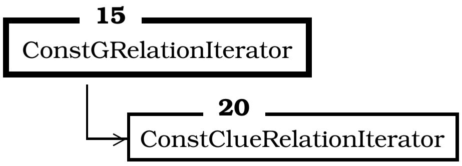
GRelationList container. The iterator works on the container which is specified in the `G_trait` structure.

Author: T. Glebe

15 template <class T> class **ConstGRelationIterator**

Iterator

Inheritance



Public Members

— Constructors —

ConstGRelationIterator ()

ConstGRelationIterator (const GRelation_cit& it)

virtual ~**ConstGRelationIterator** ()

— Operators —

inline self&
operator++ ()

inline self&
operator– ()

inline Gconst_pointer
operator* () const

inline const Gconst_pointer
operator-> () const

inline bool
operator== (const ConstGRelationIterator<T>& rhs) const

inline bool
operator!= (const ConstGRelationIterator<T>& rhs) const

Constant Forward Relation Iterator This is a STL compatible iterator for the GRelationList container. The values returned by this iterator are constant and cannot be modified (read-only). The iterator works on the container which is specified in the `G_trait` structure.

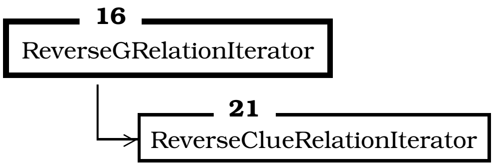
Author: T. Glebe

16

template <class T> class **ReverseGRelationIterator**

Iterator

Inheritance



Public Members

— Constructors —

ReverseGRelationIterator ()

ReverseGRelationIterator (const GRelation_rit& it)

virtual ˜**ReverseGRelationIterator** ()

— Operators —

inline self& **operator++** ()

inline self& **operator–** ()

inline Gpointer **operator*** () const

inline Gpointer **operator->** () const

inline bool **operator==** (const ReverseGRelationIterator<T>& rhs) const

inline bool

operator!= (const ReverseGRelationIterator<T>& rhs) const

Reverse Relation Iterator. This is a STL compatible iterator for the GRelationList container. The iterator traverse the GRelationList container in inverse order as the GRelationIterator. The iterator works on the container which is specified in the `G_trait` structure.

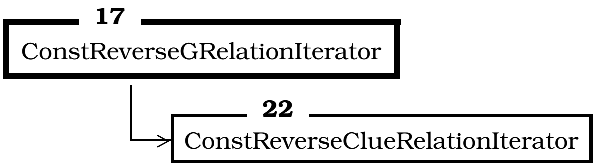
Author: T. Glebe

17

template <class T> class **ConstReverseGRelationIterator**

Iterator

Inheritance



Public Members

— Constructors —

ConstReverseGRelationIterator ()

ConstReverseGRelationIterator (const GRelation_rcit& it)

virtual ˜**ConstReverseGRelationIterator** ()

— Operators —

inline self&
operator++ ()

inline self&
operator– ()

inline Gconst_pointer
operator* () const

inline const Gconst_pointer
operator-> () const

inline bool
operator== (const ConstReverseGRelationIterator<T>& rhs) const

inline bool

operator!= (const ConstReverseGRelationIterator<T>& rhs) const

Constant Reverse Relation Iterator. This is a STL compatible iterator for the GRelationList container. The iterator traverse the GRelationList container in inverse order as the ConstGRelationIterator. The values returned by this iterator are constant and cannot be modified (read-only). The iterator works on the container which is specified in the `G_trait` structure.

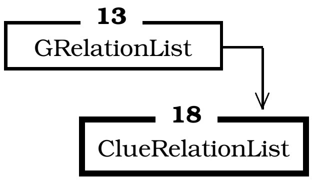
Author: T. Glebe

18

template <class T> class **ClueRelationList** : public GRelationList<ArtePointer<T> >

ARTE relation container

Inheritance



Public Members

— Constructors —

ClueRelationList ()

ClueRelationList (const ClueRelationList<T>& rhs)

˜**ClueRelationList** ()

— Expert functions —

void **Streamer** (TBuffer &R__b)
only available in ROOT

Relation container for the ARTE relation implementation of the BEE analysis framework.

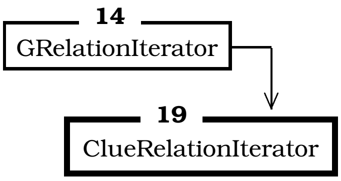
Author: T. Glebe

19

template <class T> class **ClueRelationIterator** : public GRelationIterator<ArtePointer<T> >

Iterator

Inheritance



Public Members

— Constructors —

ClueRelationIterator ()

ClueRelationIterator (const ClueRelationIterator<T>& rhs)

ClueRelationIterator (const GRelationIterator<ArtePointer<T> >& rhs)

˜**ClueRelationIterator** ()

ARTE relation like forward iterator.

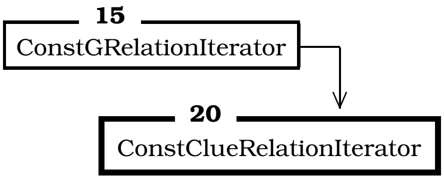
Author: T. Glebe

20

template <class T> class **ConstClueRelationIterator** : public ConstGRelationIterator<ArtePointer<T> >

Iterator

Inheritance



Public Members

— Constructors —

ConstClueRelationIterator ()

ConstClueRelationIterator (const ConstClueRelationIterator<T>& rhs)

ConstClueRelationIterator (const ConstGRelationIterator<ArtePointer<T> >& rhs)

~**ConstClueRelationIterator** ()

ARTE relation like read-only forward iterator.

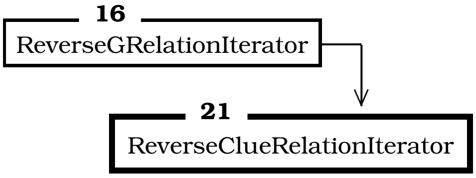
Author: T. Glebe

21

template <class T> class **ReverseClueRelationIterator** : public ReverseGRelationIterator<ArtePointer<T> >

Iterator

Inheritance



Public Members

— Constructors —

ReverseClueRelationIterator ()

ReverseClueRelationIterator (const ReverseClueRelationIterator<T>& rhs)

ReverseClueRelationIterator (const ReverseGRelationIterator<ArtePointer<T> >& rhs)

~**ReverseClueRelationIterator** ()

ARTE relation like backward iterator.

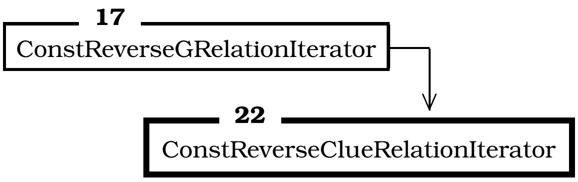
Author: T. Glebe

22

template <class T> class **ConstReverseClueRelationIterator** : public ConstReverseGRelationIterator<ArtePointer<T> >

Iterator

Inheritance



Public Members

— Constructors —

ConstReverseClueRelationIterator ()

ConstReverseClueRelationIterator (const ConstReverseClueRelationIterator<T>& rhs)

ConstReverseClueRelationIterator (const ConstReverseGRelationIterator<ArtePointer<T> >& rhs)

~**ConstReverseClueRelationIterator** ()

ARTE relation like read-only backward iterator.

Author: T. Glebe

Class Graph