GammaLib - Feature #2128

Implement slicing in the Python interface of container classes

06/21/2017 10:26 PM - Knödlseder Jürgen

Status:	Closed	**Start date:**	06/21/2017
Priority:	Normal	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	1.4.0		

Description

The following should work:

```
>>> import gammalib
>>> obs=gammalib.GObservations('obs.xml')
>>> print(obs[0:1])
```

This apparently needs a special version of the __getitem__ method that takes a PyObject as argument (see https://stackoverflow.com/questions/23206796/how-to-get-python-slicing-to-work-with-my-c-array-class-using-swig):

```
%extend Array1D
{
   Array1D* __getitem__(PyObject *param)
   {
      if (PySlice_Check(param))
      {
         /* Py_ssize_t might be needed here instead of ints */
         int len = 0, start = 0, stop = 0, step = 0, slicelength = 0;

         len = this->size(); /* Or however you get the size of a vector */

         PySlice_GetIndicesEx((PySliceObject*)param, len, &start, &stop, &step, &slicelength);

         /* Here do stuff in order to return an Array1D that is the proper slice
            given the start/stop/step defined above */
      }

      /* Unexpected parameter, probably should throw an exception here */
   }
}
```

Related issues:

Related to GammaLib - Action # 1582: Add Python unit tests for all iterators	**Closed**	**11/24/2015**

History

#1 - 06/21/2017 11:53 PM - Knödlseder Jürgen

Here the code that allows slicing of the GObservations container and that also allows to return elements from the back of the container:

```
   GObservation* __getitem__(const int& index) {
      if (index >= 0 && index < self->size()) {       // counting from start
         return (*self)[index];
      }
      else if (index < 0 && self->size()+index >= 0) { // counting from end
         return (*self)[self->size()+index];
      }
      else {
```

```
        throw GException::out_of_range("__getitem__(int)", index, self->size());
    }
}
GObservations* __getitem__(PyObject *param) {
    if (PySlice_Check(param)) {
        Py_ssize_t start = 0;
        Py_ssize_t stop  = 0;
        Py_ssize_t step  = 0;
        Py_ssize_t len   = self->size();
        if (PySlice_GetIndices((PySliceObject*)param, len, &start, &stop, &step) == 0) {
            GObservations* obs = new GObservations;
            if (step > 0) {
                for (int i = (int)start; i < (int)stop; i += (int)step) {
                    obs->append(*(*self)[i]);
                }
            }
            else {
                for (int i = (int)start; i > (int)stop; i += (int)step) {
                    obs->append(*(*self)[i]);
                }
            }
            return obs;
        }
        else {
            throw GException::invalid_argument("__getitem__(PyObject)","Invalid slice indices");
        }
    }
    else {
        throw GException::invalid_argument("__getitem__(PyObject)","");
    }
}
```

#2 - 06/22/2017 12:03 AM - Knödlseder Jürgen

- Status changed from New to In Progress

- Assigned To set to Knödlseder Jürgen

- % Done changed from 0 to 10

Implemented slicing for GObservations container. Missing classes are:

- GApplicationPars
- GEnergies
- GModels
- GOptimizerPars
- GPhotons
- GSkyRegions
- GTimes

In addition we should think whether slicing would make sense for the following classes:

- GFits
- GFitsHeader
- GFitsTable
- GGti
- GNodeArray
- GPhases
- GXml
- GXmlNode
- GCTAEventList
- GLATEventList

#3 - 07/21/2017 02:42 PM - Knödlseder Jürgen

- % Done changed from 10 to 60

Here the progress I made:

```
GApplicationPars.hpp     done
GEbounds.hpp             done (no [] operator)
GEnergies.hpp            done
GGti.hpp                 done (no [] operator)
GObservations.hpp        done
GPhases.hpp              done (no [] operator)
GPhotons.hpp             done
GTimes.hpp               done
GModels.hpp              done
GNodeArray.hpp           done
GSkyRegions.hpp          done
GOptimizerPars.hpp
GFits.hpp
GFitsHeader.hpp
GXml.hpp
GXmlNode.hpp
GTestSuites.hpp
```

Unit tests are added for all classes that are done.

#4 - 07/23/2017 04:17 AM - Knödlseder Jürgen

- Status changed from In Progress to Closed

- % Done changed from 60 to 100

Now have done all:

GApplicationPars.hpp done
GEbounds.hpp done (no [] operator)
GEnergies.hpp done
GGti.hpp done (no [] operator)
GObservations.hpp done
GPhases.hpp done (no [] operator)
GPhotons.hpp done
GTimes.hpp done
GModels.hpp done
GNodeArray.hpp done
GSkyRegions.hpp done
GOptimizerPars.hpp done (but may lead to memory leak)
GXml.hpp done
GXmlNode.hpp done
GTestSuites.hpp done
GFits.hpp done
GFitsHeader.hpp done
GCTAEventList done
GLATEventList done

Merged into devel.

#5 - 07/23/2017 04:22 AM - Knödlseder Jürgen

- Related to Action #1582: Add Python unit tests for all iterators added