My Project

Generated by Doxygen 1.8.11

Contents

Chapter 1

CompressedStacks.cpp

The CompressedStacks.cpp module/library implements a time-space trade-off structure for stack's algorithms.

Category of algorithms

This compressed stack structure works correctly as a normal stack for any problems that read input from a file. However, the running time is optimal when the input would be read sequentially with a classical stack structure. For this reason, the only function implemented in the Problem template to solve it (to do a run) is the one presented below in a simplified version.

```
template <class T, class D> void Problem<T, D>::run() {
  initStack();
  while (notEndOfFile()) {
    D data = readInput(line);
    while (notEmptystack() && popCondition(data)) {
      elt = pop();
      popAction(elt);
    }
    if (pushCondition(data)) {
      pushAction(data);
      push(data);
    }
  }
}
```

Characterization of a problem

In the followwing examples, implementations of the Problem interface are given.

General example : `Instance<T,D>`

```
#include <string>
#include <vector>
#include <memory>

// T is the type of the context and D is the type of the input data.
class Instance: public Problem<T,D>{
public:
  Instance(std::string filePath) : Problem<T, D>(filePath) {}
private:
  // Functions to implement according to the problem and input
  D readInput(std::vector<std::string> line){
    std::cout << "Implement readInput for your instance" << std::endl;
    return 0;
  }
  std::shared_ptr<T> initStack(){
    std::cout << "Implement initStack for your instance" << std::endl;
    std::shared_ptr<T> context(nullptr);
    return context;
  }
  bool popCondition(D data){
    std::cout << "Implement mPopCondition for your instance" << std::endl;
    return false;
  }
  void popAction(Data<T, D> elt){
    std::cout << "Implement mPopAction for your instance" << std::endl;
  }
  bool pushCondition(D data){
    std::cout << "Implement mPushCondition for your instance" << std::endl;
    return true;
  }
  void pushAction(Data<T, D> elt){
    std::cout << "Implement mPushAction for your instance" << std::endl;
  }
};
```

Example with `T = int` and `D = int`: `Instance<int,int>`

The context is initialized at 0. The data (in cvs format) is read as a pair of string such that the first string is the data and the second is used to update the context. While the context is more than 0, the stack is poped and the context decreased by 1. If the data is more than 0 then it is pushed.

```
class Instance : public Problem<int, int> {
public:
  Instance(std::string filePath) : Problem<int, int>(filePath) {}

private:
  // Functions to run the stack
  int readInput(std::vector<std::string> line) {
    int value = std::stoi(line[0]);
    setContext(std::stoi(line[1]));
    return value;
  }
  std::shared_ptr<int> initStack() {
    std::shared_ptr<int> context(new int(0));
    return context;
  }
  bool popCondition(int data) {
    if ((getContext() > 0)) {
      return true;
    }
    return false;
  }
  void popAction(Data<int, int> elt) {
    std::cout << elt.toString() << " <<<< Pop!" << std::endl;
    setContext(getContext() - 1);
  }
  bool pushCondition(int data) {
    if (data > 0) {
      return true;
    }
    return false;
  }
  void pushAction(Data<int, int> elt) {
    std::cout << "Push >>>> " << elt.toString() << std::endl;
  }
};
```

How to run your problem

Suppose the class Instance implement the interface Problem<T,D> (as in some examples above). You can run an instance of your problem described in the input located at *filepath*. The last command just print an output in th console of your compressed stack after the run.

```
Instance stack(filePath);
stack.run();
stack.println();
```

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

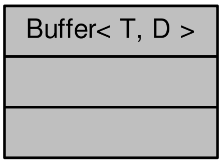
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

Class Documentation

4.1 Buffer< T, D > Class Template Reference

Collaboration diagram for Buffer< T, D >:



Friends

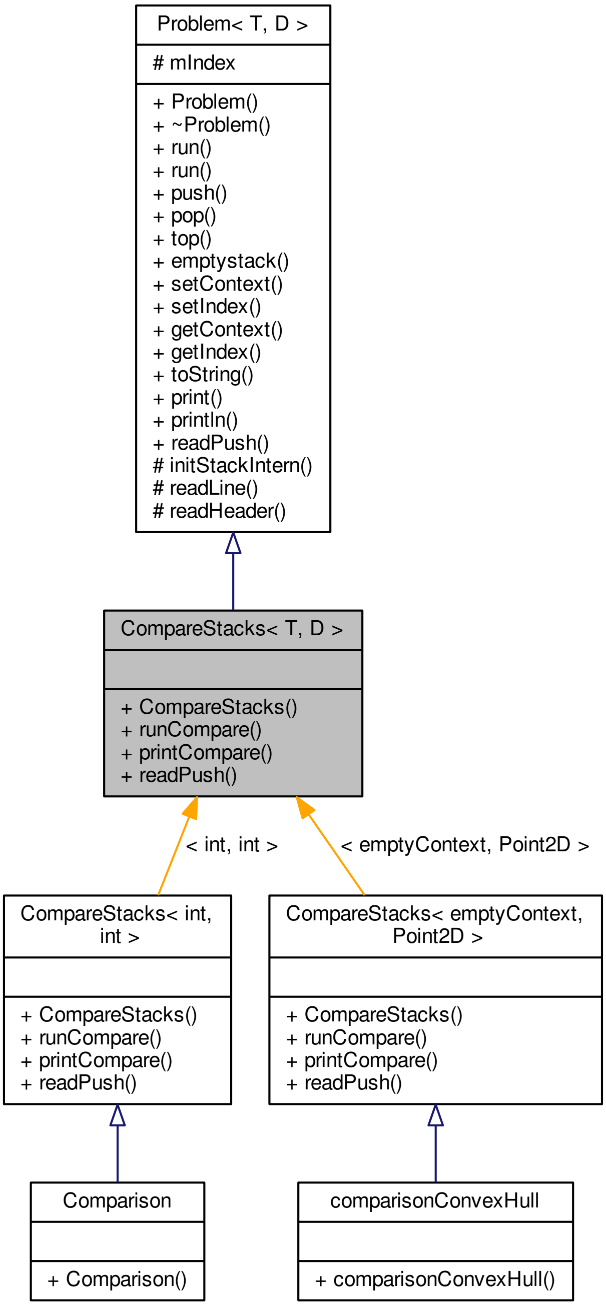
- class **CompressedStack< T, D >**
- class **NormalStack< T, D >**
- class **Signature< T, D >**
- class **Component< T, D >**

The documentation for this class was generated from the following file:

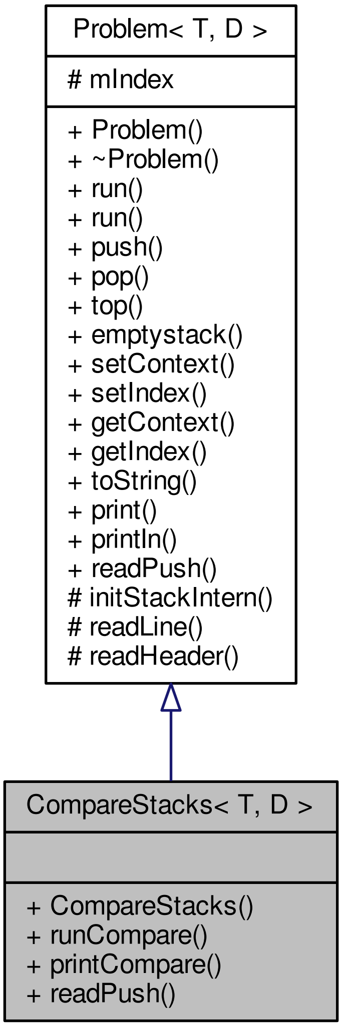
- include/buffer.hpp

4.2 CompareStacks< T, D > Class Template Reference

Inheritance diagram for CompareStacks< T, D >:



Collaboration diagram for CompareStacks< T, D >:



Public Member Functions

- **CompareStacks** (std::string fileName)
- void **runCompare** (int buffer=0)
- void **printCompare** ()
- void **readPush** (int iter=1)

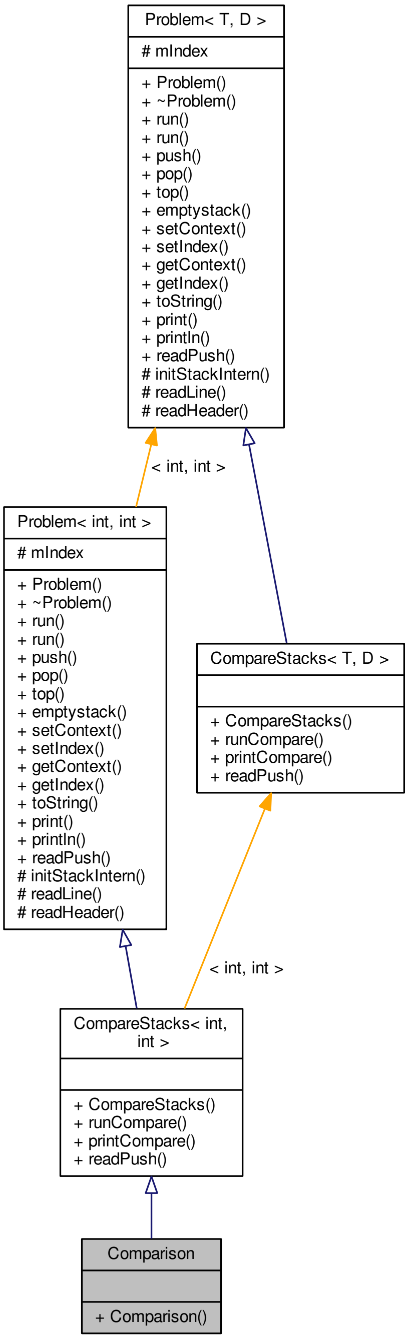
Additional Inherited Members

The documentation for this class was generated from the following file:

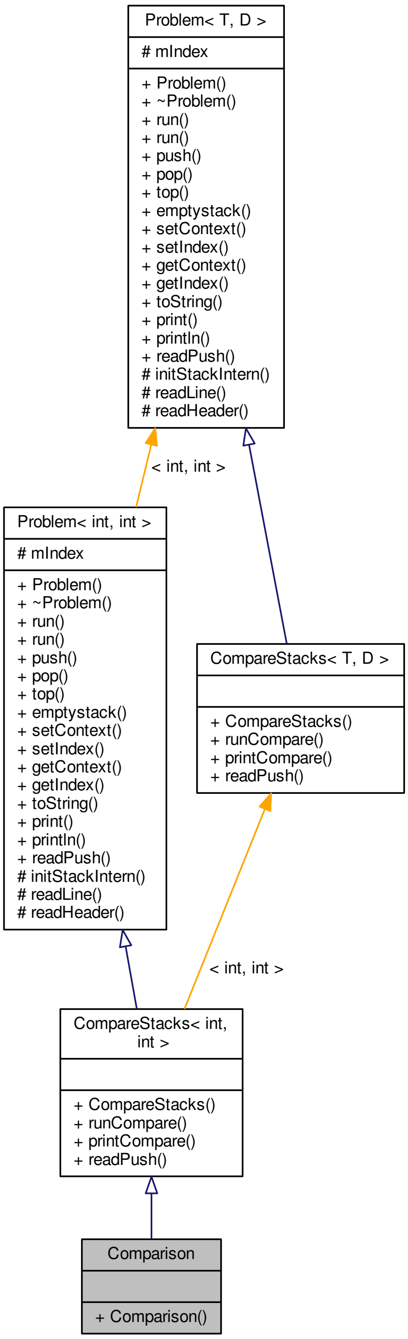
- include/compare.hpp

4.3 Comparison Class Reference

Inheritance diagram for Comparison:



Collaboration diagram for Comparison:



Public Member Functions

- **Comparison** (std::string filePath)

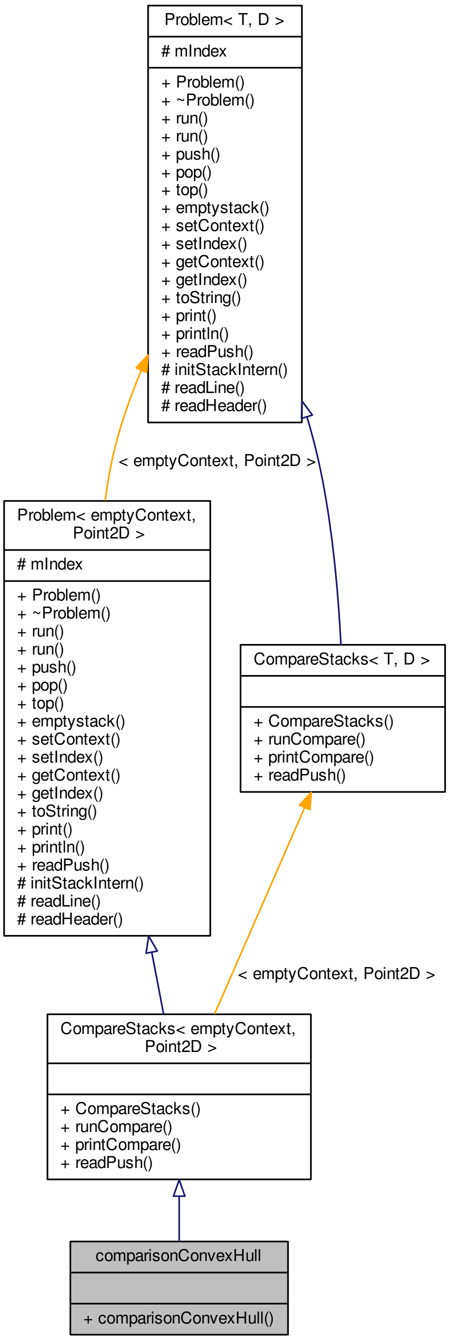
Additional Inherited Members

The documentation for this class was generated from the following file:

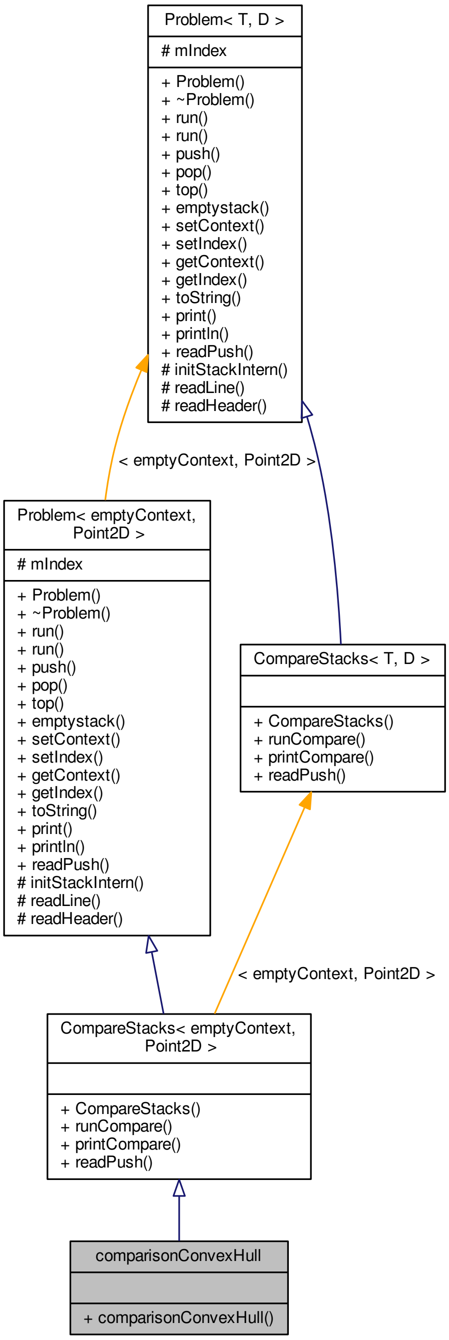
- src/main.cpp

4.4 comparisonConvexHull Class Reference

Inheritance diagram for comparisonConvexHull:



Collaboration diagram for comparisonConvexHull:



Public Member Functions

- **comparisonConvexHull** (std::string filePath)

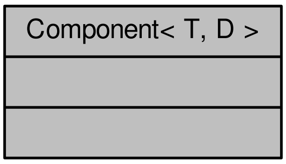
Additional Inherited Members

The documentation for this class was generated from the following files:

- include/convexHull.hpp
- src/convexHull.cpp

4.5 Component< T, D > Class Template Reference

Collaboration diagram for Component< T, D >:



Friends

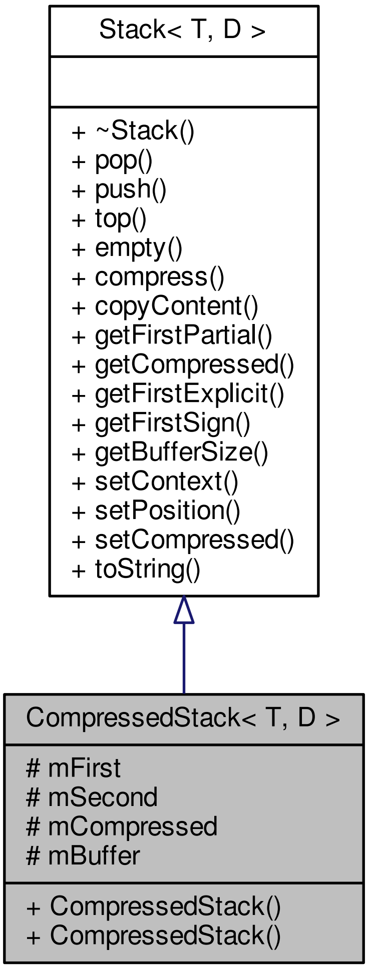
- class **CompressedStack< T, D >**

The documentation for this class was generated from the following files:

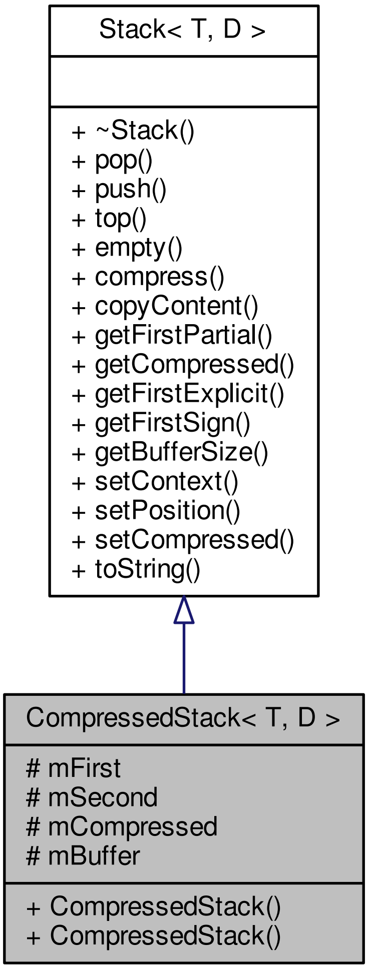
- include/buffer.hpp
- include/component.hpp

4.6 CompressedStack< T, D > Class Template Reference

Inheritance diagram for CompressedStack< T, D >:



Collaboration diagram for CompressedStack< T, D >:



Public Member Functions

- **CompressedStack** (int size, int space, int buffer, std::shared_ptr< T > context, std::streampos position=std::streampos(0))
- **CompressedStack** (int size, int space, int buffer, const Signature< T, D > &sign)

Protected Attributes

- Component< T, D > **mFirst**
- Component< T, D > **mSecond**
- Block< T, D > **mCompressed**
- Buffer< T, D > **mBuffer**

Friends

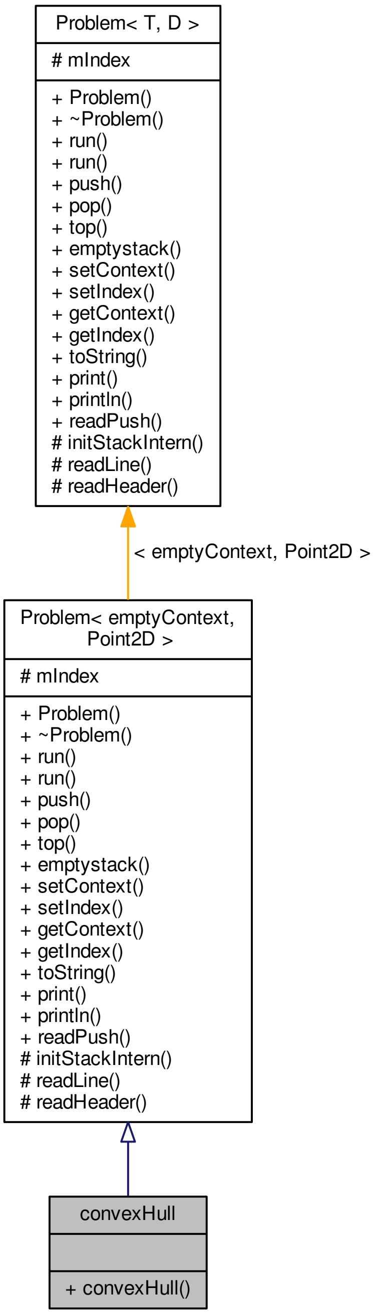
- class **Problem< T, D >**

The documentation for this class was generated from the following files:

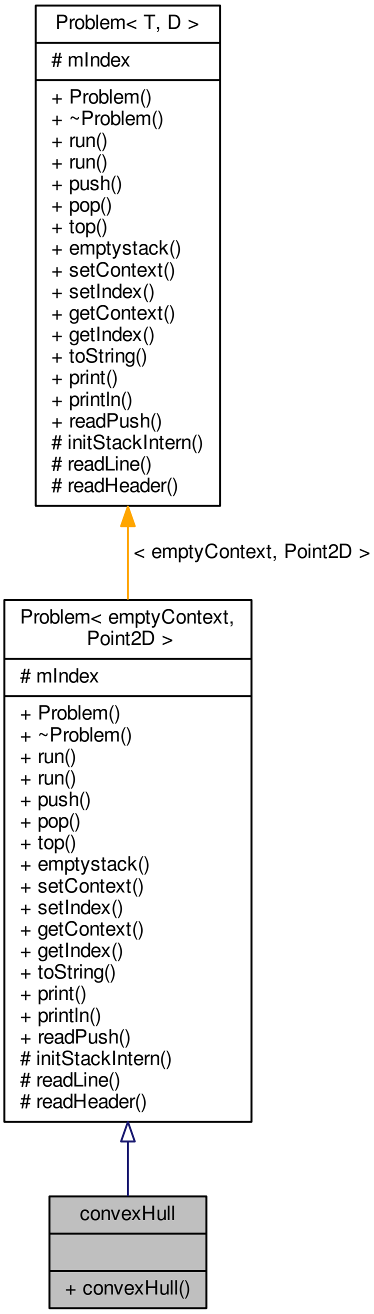
- include/buffer.hpp
- include/compressedStack.hpp

4.7 convexHull Class Reference

Inheritance diagram for convexHull:



Collaboration diagram for convexHull:



Public Member Functions

- **convexHull** (std::string filePath)

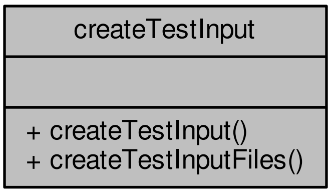
Additional Inherited Members

The documentation for this class was generated from the following files:

- include/convexHull.hpp
- src/convexHull.cpp

4.8 createTestInput Class Reference

Collaboration diagram for createTestInput:



Public Member Functions

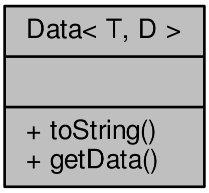
- void **createTestInputFiles** (int code, int stacktype, std::string fileName, int n, int p, int min=0, int max=100, double prob=0)

The documentation for this class was generated from the following files:

- include/createTestInput.hpp
- src/createTestInput.cpp

4.9 Data< T, D > Class Template Reference

Collaboration diagram for Data< T, D >:



Public Member Functions

- std::string **toString** ()
- D **getData** ()

Friends

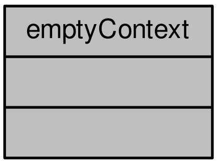
- class **Component**$<$ **T, D** $>$
- class **CompressedStack**$<$ **T, D** $>$
- class **Problem**$<$ **T, D** $>$
- class **CompareStacks**$<$ **T, D** $>$
- class **Comparison**

The documentation for this class was generated from the following file:

- include/data.hpp

4.10 emptyContext Class Reference

Collaboration diagram for emptyContext:

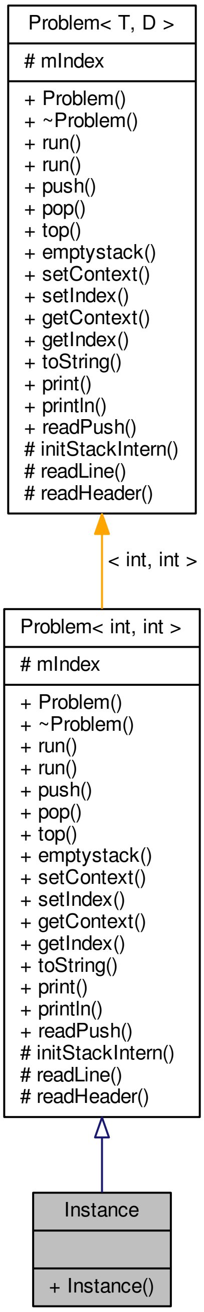


The documentation for this class was generated from the following file:

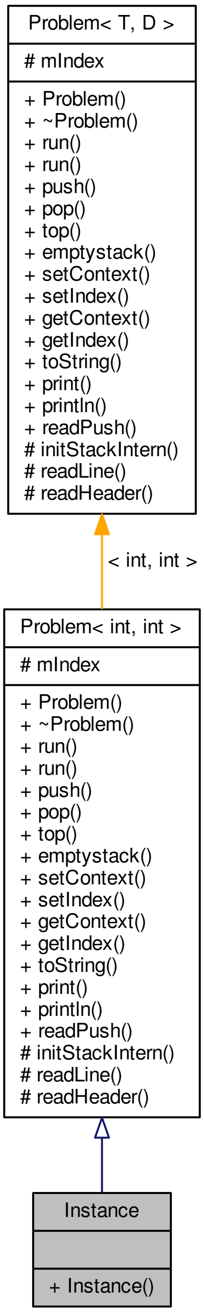
- include/convexHull.hpp

4.11 Instance Class Reference

Inheritance diagram for Instance:



Collaboration diagram for Instance:



Public Member Functions

- **Instance** (std::string filePath)

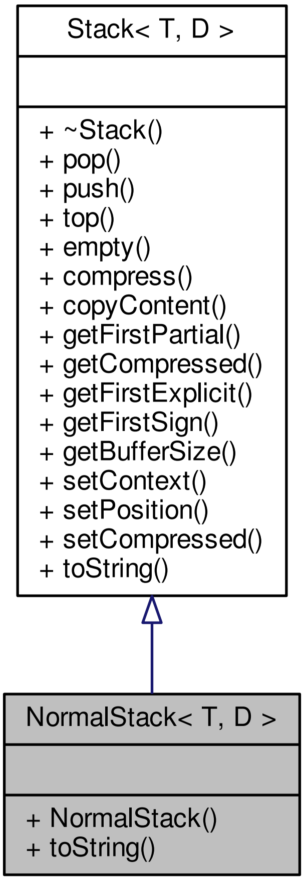
Additional Inherited Members

The documentation for this class was generated from the following file:

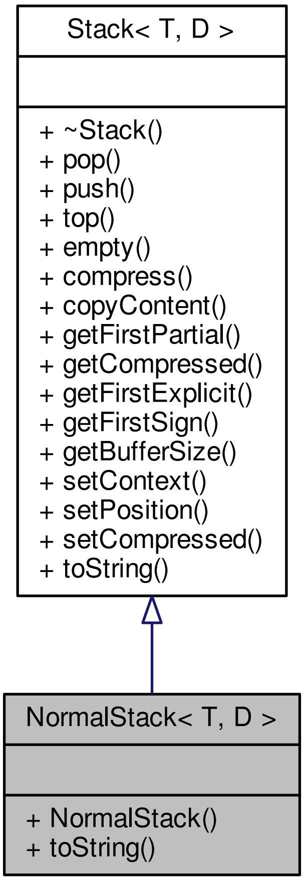
- src/main.cpp

4.12 NormalStack< T, D > Class Template Reference

Inheritance diagram for NormalStack< T, D >:



Collaboration diagram for NormalStack< T, D >:



Public Member Functions

- std::string **toString** ()

Friends

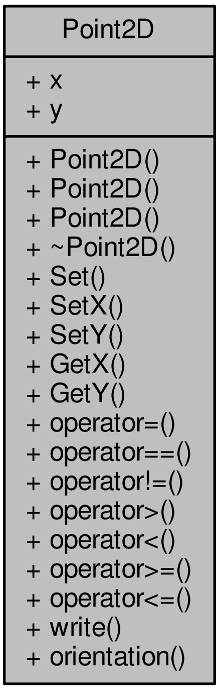
- class **Problem**< **T, D** >
- class **CompareStacks**< **T, D** >
- class **Comparison**

The documentation for this class was generated from the following files:

- include/buffer.hpp
- include/normalStack.hpp

4.13 Point2D Class Reference

Collaboration diagram for Point2D:



Public Member Functions

- **Point2D** (double x, double y)
- **Point2D** (const Point2D &other)
- void **Set** (const double x, const double y)
- void **SetX** (double x)
- void **SetY** (double y)
- double **GetX** () const
- double **GetY** () const
- void **operator=** (const Point2D &other)
- bool **operator==** (const Point2D &other) const
- bool **operator!=** (const Point2D &other) const
- bool **operator>** (const Point2D &other) const
- bool **operator<** (const Point2D &other) const
- bool **operator>=** (const Point2D &other) const
- bool **operator<=** (const Point2D &other) const
- void **write** (std::ostream &os)

Static Public Member Functions

- static int **orientation** (Point2D p1, Point2D p2, Point2D p3)

Public Attributes

- double **x**
- double **y**

Friends

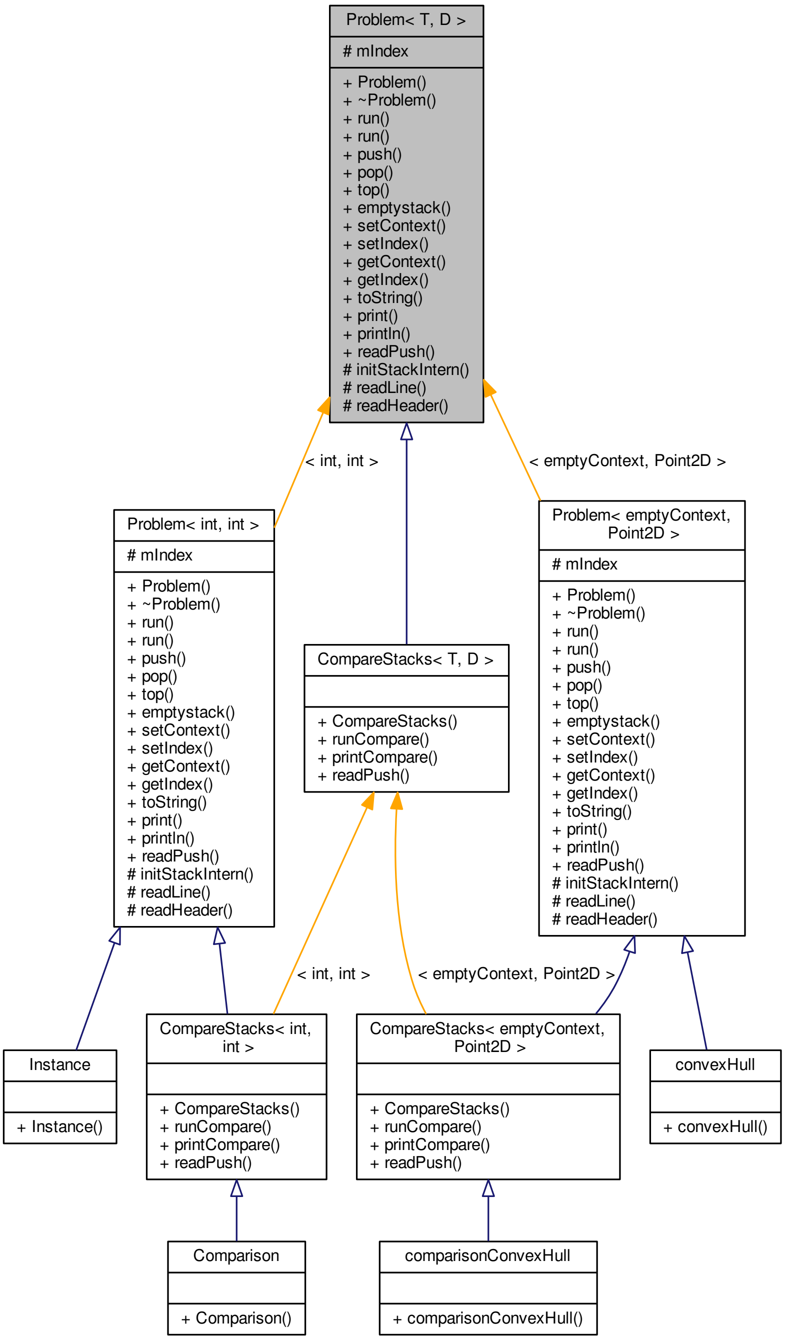
- std::ostream & **operator**<< (std::ostream &os, Point2D p)

The documentation for this class was generated from the following files:

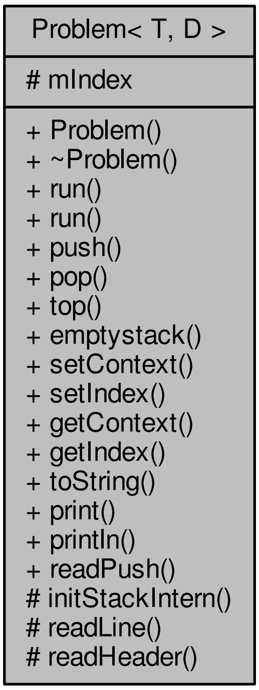
- include/Point2D.hpp
- src/Point2D.cpp

4.14 Problem< T, D > Class Template Reference

Inheritance diagram for Problem< T, D >:



Collaboration diagram for Problem< T, D >:



Public Member Functions

- **Problem** (std::string fileName)
- void **run** ()
- void **run** (int limit)
- void **push** (Data< T, D > elt)
- Data< T, D > **pop** ()
- Data< T, D > **top** (int k)
- bool **emptystack** ()
- void **setContext** (const T &context)
- void **setIndex** (int index)
- T **getContext** ()
- int **getIndex** ()
- std::string **toString** ()
- void **print** ()
- void **println** ()
- void **readPush** (int iter=1)

Protected Member Functions

- void **initStackIntern** ()
- std::vector< std::string > **readLine** ()
- std::vector< std::string > **readHeader** ()

Protected Attributes

- int **mIndex**

Friends

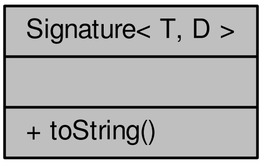
- class **CompressedStack**< **T, D** >
- class **CompareStacks**< **T, D** >
- class **Comparison**

The documentation for this class was generated from the following files:

- include/compressedStack.hpp
- include/problem.hpp

4.15 Signature< T, D > Class Template Reference

Collaboration diagram for Signature< T, D >:



Public Member Functions

- std::string **toString** ()

Friends

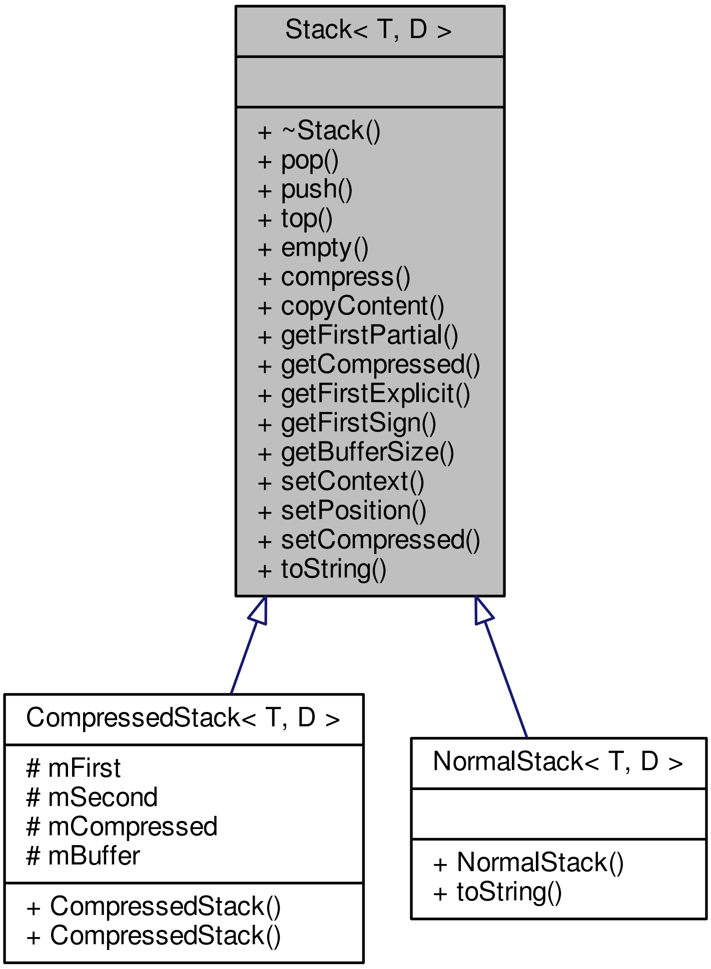
- class **Component**< **T, D** >
- class **CompressedStack**< **T, D** >
- class **NormalStack**< **T, D** >

The documentation for this class was generated from the following files:

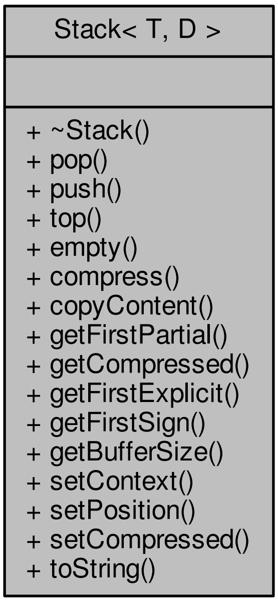
- include/buffer.hpp
- include/sign.hpp

4.16 Stack< T, D > Class Template Reference

Inheritance diagram for Stack< T, D >:



Collaboration diagram for Stack< T, D >:



Public Member Functions

- virtual Data< T, D > **pop** (Problem< T, D > &problem)=0
- virtual void **push** (const Data< T, D > &data)=0
- virtual Data< T, D > **top** (int k=1)=0
- virtual bool **empty** (int lvl=-1, int component=0)=0
- virtual void **compress** ()=0
- virtual void **copyContent** (CompressedStack< T, D > &stack)=0
- virtual Block< T, D > **getFirstPartial** (int lvl)=0
- virtual Block< T, D > **getCompressed** ()=0
- virtual ExplicitPointer< T, D > **getFirstExplicit** ()=0
- virtual Signature< T, D > **getFirstSign** ()=0
- virtual int **getBufferSize** ()=0
- virtual void **setContext** (std::shared_ptr< T > context)=0
- virtual void **setPosition** (std::streampos position)=0
- virtual void **setCompressed** (Block< T, D > block)=0
- virtual std::string **toString** ()=0

The documentation for this class was generated from the following file:

- include/stack.hpp

Index