**ARRAYSORTER.CPP**

#include "ArraySorter.h"

using namespace std;

ArraySorter::ArraySorter( const int aArrayOfNumbers[], unsigned int aArraySize )

{

*// copy array into sorter*

fArrayOfNumbers = new int[aArraySize];

for ( unsigned int i = 0; i < aArraySize; i++ )

{

fArrayOfNumbers[i] = aArrayOfNumbers[i];

}

fArraySize = aArraySize;

}

ArraySorter::~ArraySorter()

{

*// delete memory associated with array*

delete [] fArrayOfNumbers;

}

void ArraySorter::stepCompleted(std::ostream &aOStream)

{

aOStream << "State: "<< \*this << endl;

}

void ArraySorter::swapElements( unsigned int aSourcIndex, unsigned int aTargetIndex)

{

int temp = at(aSourcIndex);

fArrayOfNumbers[aSourcIndex] = at(aTargetIndex);

fArrayOfNumbers[aTargetIndex] = temp;

}

const unsigned int ArraySorter::at(unsigned int aIndex) const

{

if (aIndex > fArraySize)

{

throw range\_error("The index's length is not within the array size. Please modify the index");

}

return fArrayOfNumbers[aIndex];

}

const unsigned int ArraySorter::getRange() const

{

return fArraySize;

}

void ArraySorter::sort(ostream& aOStream)

{

stepCompleted(aOStream);

}

ostream& operator<<(std::ostream& aOStream, const ArraySorter& aObject)

{

aOStream << "[";

for (unsigned int i = 0; i < aObject.getRange(); i++)

{

aOStream << aObject.at(i);

if (i < aObject.getRange() - 1)

aOStream << ", ";

}

aOStream << "]";

return aOStream;

}

**SELECTIONSORT.CPP**

#include "SelectionSort.h"

using namespace std;

SelectionSort::SelectionSort(int aArrayOfNumbers[], unsigned int aArraySize): ArraySorter::ArraySorter(aArrayOfNumbers, aArraySize)

{

}

void SelectionSort::sort(std::ostream& aOStream)

{

for (unsigned int i = 0; i < getRange() - 1; i++)

{

unsigned int b = i;

for (unsigned int a = i + 1; a < getRange(); a++)

{

if (at(a) < at(b))

{

b = a;

}

}

swapElements(i, b);

stepCompleted(aOStream);

}

}

**INSERTIONSORT.CPP**

#include "InsertionSort.h"

using namespace std;

InsertionSort::InsertionSort(int aArrayOfNumbers[], unsigned int aArraySize) : ArraySorter::ArraySorter(aArrayOfNumbers, aArraySize)

{

}

void InsertionSort::sort(std::ostream& aOStream)

{

for (unsigned int i = 0; i < getRange()-1; i++)

{

for (unsigned int b = i+1; b > 0; b--)

{

if (at(b) < at(b-1))

{

swapElements(b, b-1);

}

}

stepCompleted(aOStream);

}

}

A close up of text on a black background

Description automatically generated