Katarina Simakina/Report/Assignment1

I have created the application which executes txt files. Running the program, a user needs to write a name of the file to execute it. The result of the execution is two files: first is the intermediate result, the name of it for example scenario1.out1 and the final result is scenario1.out.

My implementation has been done in the following steps:

1. The class Cell has the following parameters:

```
public class Cell {
private int start;
private int size;
private int id;
private boolean free;
   }
```

As well I have added the constructor and the following methods:
The method which helps to increase the size of free cells.

```
  public void addSize(int size){
          this.size += size; }
```

The method which helps to get finishing of the cell.

```
public int getFinish () {
   return start+size-1;// starting at 100, finishing 199
}
```

In the class Cell there are methods which help to get start, size, id of the cell and the methods which set start, size, id.

2. I have created the enum class where I have listed the strategies which need to be implemented.

```
public enum AllocateStrategy {
   First,
   Best,
   Worst
}
```

3. The class Memory which has the following parameters:

```
public class Memory {
   private ArrayList<Cell>cells = new ArrayList<>();
   private int maxSize;
   private AllocateStrategy strategy;
   private ArrayList<Error>errors=new ArrayList<>();
}
```

In this class I have implemented the following methods:

public boolean allocateFirst (int id, int sizeBlock){
Creating a variable. It is counter.
int start=0;
using the loop;
using if-else statement. If the cell is free and the size of the cell>=sizeBlock (amount of memory).
If the size of the cell equals to the size of the sizeBlock then setting that this cell is false (it is occupied).
Setting the id for this cell.
Else creating Cell cell = new Cell(start, sizeBlock, id);
Decreasing the size of the free cell.
cells.get(i).addSize(-sizeBlock);
adding to the list.
cells.add(i,cell);

In the same way the following methods:
public boolean allocateBest(int id, int sizeBlock){
using the loop;
using if-else statement
adding to the list}

public boolean allocateWorst(int id, int sizeBlock){
using the loop;
using if-else statement
adding to the list}

public boolean deallocate (int id){
using the loop;
using if;
remove cell from the list of the cells.
}
public void compact (){
using loop;
if-else statement;
removing from the list }

4.Creating the abstract class Error.
   attribute:
   private int numberofString;

5.Creating the AllocateError and Deallocate which are inherited from Error, they are subclasses.

```
public class AllocateError extends Error {
   private int maxBlock;
   private int id;}

public class DeallocateError extends Error{
   private int reasonError;}
```