Machine Problem 1

reading_with_exceptions(40 points)

Create a package named "reading_with_exceptions". Write a class named: ReadingWithExceptions with the following method:

void process(String inputFilename)

Your program will encounter errors, and we want you to gracefully handle them in such a way that you print out informative information and continue executing.

1. Your process routine will try to open a file with the name of inputFilename for input. If there is any problem (i.e. the file doesn't exist), then you should catch the exception, give an appropriate error and then return. Otherwise, your program reads the file for instructions.
2. Your process routine will read the first line of the file looking for an outputFilename String followed by an integer. i.e.:
   outputFilename   number_to_read
   Your program will want to write output to a file having the name outputFilename. Your program will try to read from "inputFilename" the number of integers found in "number_to_read".
3. Your process method will copy the integers read from inputFilename and write them to your output file (i.e. outputFilename). There should contain 10 numbers per line of output in your output file.
4. If you encounter bad input, your program should not die with an exception. For example:
   - If the count of the numbers to be read is bad or < 0 you will print out a complaint message and then read as many integers as you find.
   - If any of the other numbers are bad, print a complaint message and skip over the data
   - If you don't have enough input numbers, complain but do not abort
5. After you have processed inputFilename, I would like your program to then close the output file and tell the user that the file is created. Then Open up the output file and copy it to the Screen.

For example, if inputFilename contained:

```
MyOutput.txt 23
20 1 4 5 7
45
1 2 3 4 5 6 7 8 9 77 88 99 23 34
56 66 77 88 99 100 110 120
```

We would expect the output of your program to be (Note that after 23 numbers we stop printing numbers):

```
MyOutput.txt created with the following output:
20 1 4 5 7 45 1 2 3 4
5 6 7 8 9 77 88 99 23 34
56 66 77
```

The main program will access the command line to obtain the list of filenames to call your process routine with.

To prove that your program works, I want you to run your program with the following command line parameters:

```
file1.txt non-existent-file file2.txt file3.txt
```

Where `non-existent-file` does not exist.

file1.txt contains:

```
MyOutput1.txt 22
20 1 4 5 7 8 9 10 11 12 13 14
45 46 47 48 49 50 51
1 2 3 4 5 6 7 8 9 77 88 99 23 34
56 99 88 77 66 55 44 33 22 11
```

file2.txt contains:

```
niceJob.txt 40
20 1 x 5 7 45 1 2 3 4 5 6 7 8 9 77 88 99 23 34
56
```

file3.txt contains:

```
OneLastOutput.txt x0
20 1 5 7 45 1 2 3 4 5 6 7 8 9 77 88 99 23 34
56 99 88 11 22 33 44 55
66 77
```

PersonIO (25 points)

Write a program that can write/read person objects to/from a binary file.

- In the program, a main menu is displayed as shown in the sample run. You can enter a choice 1 for adding a person, 2 for displaying the list of person, 0 for exiting the main menu.
- You should have a “add” method that writes person objects into a binary file.
- When adding a person, the user will be prompted to enter the person’s name and age from keyboard.
- You should have a “display” method that reads the binary file and display all person object in the console.
- Person class should have two instance variables: name and age
- Person class should contain one constructor that takes name and age as parameters. e.g.

  public Person(String name, int age)
- Person class should have a toString method that displays the person’s name and age.

Sample output:

```
Please choose an option:
0: quit
1: add
2: display
1
Please enter the person's name:
Mike
Please enter the person's age:
39
Please choose an option:
0: quit
1: add
2: display
1
Please enter the person's name:
Jenny
Please enter the person's age:
12
```

```
Please choose an option:
0: quit
1: add
2: display
2
***********************
Person [name=Mike, age=39]
Person [name=Jenny, age=12]
***********************
Please choose an option:
0: quit
1: add
2: display
0
Bye
```