Lab #4

Notes

Folder organization for solutions on the student's account:

```
z:\
|
|--Programming02
|----Lab01Problem01
|----Lab01Problem02
|----…
|----Lab02Problem01
|----…
|--Programming01
```

Task #1: Class ArrayList from standard library 1 (0.5%)

Write a program using class java.util.ArrayList which reads unknown amount of integer numbers from standard input; prints them, reverses and prints them again. You have to use an object of class ArrayList<Integer> to store and reverse list of integer numbers entered by user.

Example:
```
1 2 3 4 5 6 7
^-Z
Before reversing:
1 2 3 4 5 6 7
After reversing:
7 6 5 4 3 2 1
```

Task #2: Class ArrayList from standard library 2 (0.5%)

Write a program using class java.util.ArrayList which reads unknown amount of integer numbers from standard input; inserts value 0 before each even number in the list; prints the content of list; removes all odd values; prints content of the list again. You have to use an object of ArrayList<Integer> to work with integer numbers entered by user.

Example:
```
2 2 3 4 5 6 7 7 8
^-Z
After insertions:
0 2 0 2 3 0 4 5 0 6 7 7 0 8
After deletions:
0 2 0 2 0 4 0 6 0 8
```

Task #3: Class ArrayListInt (1.5%)

You have to implement class ArrayListInt (simple version of ArrayList from standard library) and do 3 simple experiments with it.

Class has to have following methods:

Method	Meaning
ArrayListInt()	Creates empty, ready to use list of integer numbers
int size()	Returns the current size of the list
int get(int i)	Return value of element with index i
void set(int i, v)	Sets value of element with index i to v
void add(int e)	Adds value e at the end of the list
void add(int i, int v)	Inserts value v before element with index I
void remove(int I)	Remove element with index I

1 experiment (0.5%): Show that the simplest way to add elements to a list (each time when add new element to create a new array the size of which is greater by 1, copy element from old array, reassign reference) is very inefficient comparing to standard ArrayList.

2 experiment (0.5%): Find a way that standard ArrayList uses to add elements. Implement this approach for class ArrayListInt. Compare efficiency of this operation for classes ArrayList and ArrayListInt. Explain the difference.

3 experiment (0.5): Solve Task1 and Task 2 of this lab using class ArrayListInt.

Home Reading: Liang Introduction to Java Programming 8th ed. Chapter 11.