Exercise 7 (10 points) – can be done individually or in pair

- The first lines of all source files must be comment containing names & IDs of all members. Also create file readme.txt containing names & IDs of all members.
- Put all files (source, input, output) in folder Ex7_xxx where xxx = your full ID. That is, your source files must be in package Ex7_xxx and input/output files (if there is any) must be read from/write to this folder. From now on, you'll get point deduction for wrong package & folder structure.
- The group representative zips Ex7_xxx & submits it to Google Classroom. The other members submit only readme.txt. Email submission is not accepted.
- The exercise is graded only once, and after graded, members can't be added.

==

Complete the given source file to make the program work as follows:

1. Complete class BankThread. You can add more variables & methods, change method headers, but don't change the visibility of existing ones
    - Use Exchanger to exchange Account between depositing BankThreads
    - Use CyclicBarrier to make threads start some tasks at the same time

2. Complete class Account. You can add more variables & methods, change method headers, but don't change the visibility of existing ones
    - Use Semaphore or monitor to let only 1 thread update balance and print to System.out at a time. To get correct result, balance & System.out should be protected together

3. Complete method runSimulation for main thread's activities
    - Use CyclicBarrier to make threads start some tasks at the same time
    - Use Join to make main thread wait until all BankThreads complete their works before printing final balances

4. Every output line must be labeled by the name of the thread who prints it. Don't hard code thread name, but use Thread.currentThread() to get the printing thread

```
main >> Enter #rounds for a new simulation(-1 to quit)
3
W2   >> manage     ..................................account B (balance = 0)
D1   >> manage     account A (balance = 0)
D2   >> manage     ..................................account B (balance = 0)
W1   >> manage     account A (balance = 0)
W1   >> round 1    account A  cannot withdraw
W2   >> round 1    ..................................account B  cannot withdraw
D1   >> round 1    account A  +51  balance =  51
D2   >> round 1    ..................................account B  +43  balance =  43
W1   >> round 2    account A   -8  balance =  43
D2   >> round 2    ..................................account B  +95  balance = 138
D1   >> round 2    account A  +75  balance = 118
W2   >> round 2    ..................................account B  -69  balance =  69
W2   >> round 3    ..................................account B   -4  balance =  65
W1   >> round 3    account A  -32  balance =  86
D2   >> round 3    ..................................account B   +6  balance =  71
D1   >> round 3    account A  +58  balance = 144
```

Each BankThread identifies account it is managing in this simulation

Account update in each line must be correct

```
main >> Enter #rounds for a new simulation(-1 to quit)
4
D2   >> exchange account
D1   >> exchange account
D1   >> manage     ..................................account B (balance = 71)
W1   >> manage     account A (balance = 144)
W2   >> manage     ..................................account B (balance = 71)
D2   >> manage     account A (balance = 144)
D2   >> round 1    account A   +3  balance = 147
D1   >> round 1    ..................................account B  +26  balance =  97
W1   >> round 1    account A  -18  balance = 129
W2   >> round 1    ..................................account B  -14  balance =  83
D1   >> round 2    ..................................account B  +22  balance = 105
W2   >> round 2    ..................................account B  -12  balance =  93
D1   >> round 3    ..................................account B  +76  balance = 169
D2   >> round 2    account A  +37  balance = 166
W1   >> round 2    account A  -15  balance = 151
D1   >> round 4    ..................................account B   +2  balance = 171
W2   >> round 3    ..................................account B  -31  balance = 140
W1   >> round 3    account A  -34  balance = 117
W1   >> round 4    account A   -7  balance = 110
W2   >> round 4    ..................................account B  -43  balance =  97
D2   >> round 3    account A  +93  balance = 203
D2   >> round 4    account A  +60  balance = 263
```

Only depositing BankThreads (i.e. D1, D2) exchange accounts

Account balance must continue from previous simulation

```
main >> Enter #rounds for a new simulation(-1 to quit)
3
D2   >> exchange account
D1   >> exchange account
D2   >> manage     ..................................account B (balance = 97)
W1   >> manage     account A (balance = 263)
W2   >> manage     ..................................account B (balance = 97)
D1   >> manage     account A (balance = 263)
D1   >> round 1    account A  +42  balance = 305
D2   >> round 1    ..................................account B  +48  balance = 145
W1   >> round 1    account A  -80  balance = 225
W2   >> round 1    ..................................account B  -31  balance = 114
W1   >> round 2    account A  -78  balance = 147
D2   >> round 2    ..................................account B  +24  balance = 138
W2   >> round 2    ..................................account B  -37  balance = 101
D1   >> round 2    account A  +94  balance = 241
W1   >> round 3    account A  -77  balance = 164
W2   >> round 3    ..................................account B  -13  balance =  88
D2   >> round 3    ..................................account B  +37  balance = 125
D1   >> round 3    account A  +11  balance = 175


main >> Enter #rounds for a new simulation(-1 to quit)
-1
main >> final balance    account A = 175
main >> final balance    ..................................account B = 125
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
```