

### 1. (20 Points) NumberFormatException

Write the `bin2Dec(String binaryString)` method to convert a binary string into a decimal number. Implement the `bin2Dec` method to throw a `NumberFormatException` if the string is not a binary string. Write a test program.

### 2. (20 Points) ArrayIndexOutOfBoundsException

Write a program meets the following requirements:

- (1) Create an array with 100 randomly chosen integers.
- (2) Prompt the user to enter the index of the array, then displays the corresponding element value. If the specified index is out of bounds, display the message **Out of Bounds**.
- (3) Write a test program.

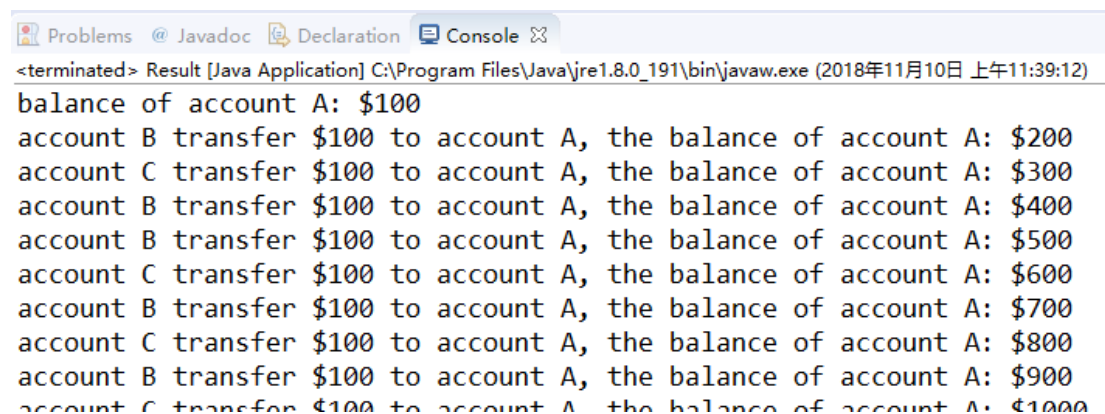
### 3. (30 Points) Find prime numbers with multiple threads

Design a program to find prime numbers using multiple threads. Each thread extends the **Thread** class and can find all the prime numbers in a certain range. In this exercise, you are required to create three threads: thread 1 finds all the prime numbers from 1 to 1000, thread 2 finds all the prime numbers from 1001 to 2000, and thread 3 finds all the prime numbers from 2001 to 3000. Print a prime number once it is found.

### 4. (30 Points) Synchronization of bank system

Design a program using interface **Runnable** to simulate a bank system. There is an account A with 100 dollars. Two different accounts, i.e., B and C, tend to transfer money into account A. When doing the transferring operation, the system gets the balance of account A and then update it by adding the money that another account transfers to it. An output example is shown in Fig. 1.

Finally, your program is required to do a statistics and print the result after all transfers to check whether it runs properly. An output example is shown in Fig. 2.



```
<terminated> Result [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (2018年11月10日 上午11:39:12)
balance of account A: $100
account B transfer $100 to account A, the balance of account A: $200
account C transfer $100 to account A, the balance of account A: $300
account B transfer $100 to account A, the balance of account A: $400
account B transfer $100 to account A, the balance of account A: $500
account C transfer $100 to account A, the balance of account A: $600
account B transfer $100 to account A, the balance of account A: $700
account C transfer $100 to account A, the balance of account A: $800
account B transfer $100 to account A, the balance of account A: $900
account C transfer $100 to account A, the balance of account A: $1000
```

Fig. 1: An output example of a bank system.

```
account B transfer $1000 to account A in total
account C transfer $1000 to account A in total
the banlance of account A: $2100
```

Fig. 2: An output example of statistics result.

**5. (Bonus Question: 20 Points) Deadlock: dining philosopher problem**

Five silent philosophers sit at a round table with bowls of spaghetti shown in Fig. 3. Forks are placed between each pair of adjacent philosophers. Each philosopher must alternately think and eat. However, a philosopher can only eat spaghetti when they have both left and right forks. Each fork can be held by only one philosopher and so a philosopher can use the fork only if it is not being used by another philosopher. After an individual philosopher finishes eating, they need to put down both forks so that the forks become available to others. A philosopher can take the fork on their right or the one on their left as they become available, but cannot start eating before getting both forks.

Now you are required to do the following tasks:

- 1) Write a program that could cause a deadlock.
- 2) Write another program that can avoid the deadlock so that all the philosophers can eat alternately.

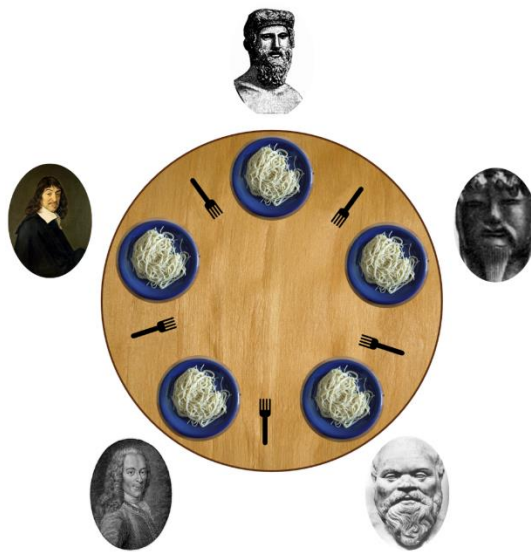


Fig. 3: The Dining Philosophers Problem.