♥ ♦ Java coding practice problems – prime and powerful

Posted on April 8, 2015 by Arulkumaran Kumaraswamipillai

2 Like Share 4 G+1

There is no other way to pass the Java coding tests other than practicing. I have failed Java coding tests due to lack of practice. How often do you get work with tree and graph data structures? Ho often do you work with algorithms?

50+ Java coding practice problems Links:

Can you write code in Java? | Designing your classes & interfaces in Java | Java Data Structures & Algorithms | Passing the unit tests | What is wrong with this code?

Q1. Can you write code to check if a given number is prime?
A1. A prime number is a number that is divisible only by **itself**

and of course 1.

More facts about prime numbers

- 1 is not a prime number. Prime numbers start from 2
- 2 is the first and only **even** prime number
- all the other prime numbers apart from 2 are odd numbers starting from 3. **E.g.** 3, 5, 7

A naive solution

```
1  public static boolean isPrimeNaive(int number) {
2      if (number % 2 == 0 || number == 1) {
3          return (number == 2);
4      }
5
6      //if the number is divisible by other than i
7      //it is not a prime
8      for (int i = 3; i < number; i++) {
9          if(number % i == 0){
10             return false;
11         }
12     }
13
14     return true;
15 }
```

Q. What is wrong with the above solution?
A. It is not optimal. if the number is 97, you will end up dividing 97 by from 3 to 96. Goes through the loop 93 times.

A better solution

Two improvements can be made

1) Instead of i++, do **i+2**, to check for only odd numbers as 2 is the only even prime number. All the others are odd
2) Instead of i < number, do **i*i < number** because if you look at factors of 99, the factors are repeated **half way mark**

```
1 99 = 1 * 99 == 3 * 33 == 9 * 11 == 11 * 9  == 33
```

You can see repeated factors –> 9 * 11 and 11 * 9, 3 * 33 and 33 * 3, etc. So, the revised solution will take advantage of this finding. So, for 97, it will only loop through 4 times for

16 Technical Key Areas

numbers 3, 5, 7, 9. When it gets to 11, it exits the loop as 11 * 11 = 121 which is > 97.

```
1  public static boolean isPrime(int number) {
2          // even numbers stop here.
3          // Only 2 is an even prime number & the
4          // 1 is not a prime number
5          if (number % 2 == 0 || number == 1) {
6              return (number == 2);
7          }
8
9          // odd numbers from 3 to n get here
10         // goes inside a loop only if i*i <= num
11         // so, numbers 3, 5, 7 skip this for loo
12         // 9, 11, 13, 15, etc get in as 3*3 = 9,
13         for (int i = 3; i * i <= number; i += 2)
14             // divisible by other than itself
15             if (number % i == 0){
16                 return false;
17             }
18         }
19
20         // if gets here, it is a prime
21         return true;
22 }
```

Another more effective approach for isPrime(int number) is:

```
1  private static boolean isPrime(int number) {
2          // even numbers stop here.
3          // Only 2 is an even prime number & the
4          // 1 is not a prime number
5          if (number % 2 == 0 || number == 1) {
6              return (number == 2);
7          }
8
9          int i = 2;
10         while (i < Math.sqrt(number)) {
11             if (number % i == 0) {
12                 return false;
13             }
14             i++;
15         }
16
17         // if gets here, it is a prime
18         return true;
19 }
```

Q2. Can you write a method that gives a list of prime numbers within a given range?
A2. The range will be supplied via “from” and “to”.

```
1  import java.util.List;
2
```


80+ step by step Java Tutorials

open all | close all

- Setting up Tutorial (6)
- Tutorial - Diagnosis (2
- Akka Tutorial (9)
- Core Java Tutorials (2
- Hadoop & Spark Tuto
- JEE Tutorials (19)
- Scala Tutorials (1)
- Spring & HIbernate Tu
- Tools Tutorials (19)
- Other Tutorials (45)

100+ Java pre-interview coding tests

open all | close all

- Can you write code? (
- ♦ Complete the given
- Converting from A to I
- Designing your classe
- Java Data Structures
- Passing the unit tests
- What is wrong with th
- Writing Code Home A
- Written Test Core Jav
- Written Test JEE (1)

How good are your .....?

open all | close all

- Career Making Know-

Job Hunting & Resum

```
3  public class PrimeNumber {
4
5      public static void main(String[] args) {
6          System.out.println("Prime numbers=" + ge
7      }
8
9      public static List<Integer> getPrimeNumbers(i
10         List<Integer> primeNumbers = new ArrayLi
11         for (int number = from; number <= to; nu
12             if(isPrime(number)){
13                 primeNumbers.add(number);
14             }
15         }
16
17         return primeNumbers;
18     }
19
20     private static boolean isPrime(int number) {
21         // even numbers stop here.
22         // Only 2 is an even prime number & the
23         // 1 is not a prime number
24         if (number % 2 == 0 || number == 1) {
25             return (number == 2);
26         }
27
28         int count = 0;
29         // odd numbers from 3 to n get here
30         // goes inside a loop only if i*i <= num
31         // so, numbers 3, 5, 7 skip this for loo
32         // 9, 11, 13, 15, etc get in as 3*3 = 9,
33         for (int i = 3; i * i <= number; i += 2)
34             System.out.println("i=" + i);
35             count++;
36             // divisible by other than itself
37             if (number % i == 0){
38                 return false;
39             }
40         }
41
42         System.out.println("count=" + count);
43
44         // if gets here, it is a prime
45         return true;
46     }
47 }
```

Output:

```
1 Prime numbers=[2, 3, 5, 7, 11, 13, 17, 19, 23, 29
2 97, 101, 103, 107, 109, 113, 127, 131, 137, 139,
```

Q3. Can you list “powerful numbers” between a given range, where the definition of a powerful number is — A positive integer **m** which is for every” **p**” that divides “m”, “p*p” must also divide m, where “p” is a prime number

The powerful numbers from 1 to 40 are: 1, 4, 8, 9, 16, 25, 27, 32, 36, …

12 is not a powerful number because: 3 divides 12 but 3*3=9 does not.
18 is not a powerful number because: 2 divides 18, but 2*2=4 does not.
1 is a powerful number because: neither p nor p*p divides it.

A3.

```
1  package algorithms;
2
3  import java.util.ArrayList;
4  import java.util.List;
5
6  public class PowerfulNumber {
7
8      public static void main(String[] args) {
9          System.out.println("powerfulNums= " + ge
10     }
11
12     public static List<Integer> getPowerfulNumbe
13         List<Integer> powerfulNums = new ArrayLi
14         List<Integer> primeNumbers = getPrimeNum
15         for (int m = from; m <= to; m++) {
16             boolean isPowerfulNumber = true;
17             for (Integer p : primeNumbers) {
18                 // every p that divides m, p*p m
19                 if(m % p == 0 && m % (p*p) != 0)
20                     isPowerfulNumber = false;
21                     break;
22                 }
23             }
24
25             if(isPowerfulNumber){
26                 powerfulNums.add(m);
27             }
28         }
29
30         return powerfulNums;
31     }
32
33     private static List<Integer> getPrimeNumbers
34         List<Integer> primeNumbers = new ArrayLi
35         for (int number = from; number <= to; nu
36             if(isPrime(number)){
37                 primeNumbers.add(number);
38             }
39         }
40
41         return primeNumbers;
42     }
43
44     private static boolean isPrime(int number) {
45         if (number % 2 == 0 || number == 1) {
46             return (number == 2);
47         }
48
49         for (int i = 3; i * i <= number; i += 2)
50             // divisible by other than itself
51             if (number % i == 0){
52                 return false;
```

```
53                 }
54             }
55
56             return true;
57         }
58
59 }
```

Output:

```
1 powerfulNums= [1, 4, 8, 9, 16, 25, 27, 32, 36]
```

50+ Java coding practice problems Links:

Can you write code in Java? | Designing your classes & interfaces in Java | Java Data Structures & Algorithms | Passing the unit tests | What is wrong with this code?

Popular Posts

♦ 11 Spring boot interview questions & answers

823 views

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

765 views

18 Java scenarios based interview Questions and Answers

399 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

388 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

295 views

♦ 7 Java debugging interview questions & answers

293 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

285 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

279 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

239 views

001B: ♦ Java architecture & design concepts interview questions & answers

201 views

Bio | Latest Posts

Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

About Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

‹ Transaction management in SQL — Sybase example

Auditing database record changes in Oracle ›

Posted in Algorithms, Can you write code?

Tags: Free Content, Free FAQs

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function?
☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

Prepare to succeed

☀ Turn readers of your Java CV go from “Blah blah” to “Wow”? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category ▼

© Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.

© 2016 Java-Success.com ↑ Responsive Theme powered by WordPress