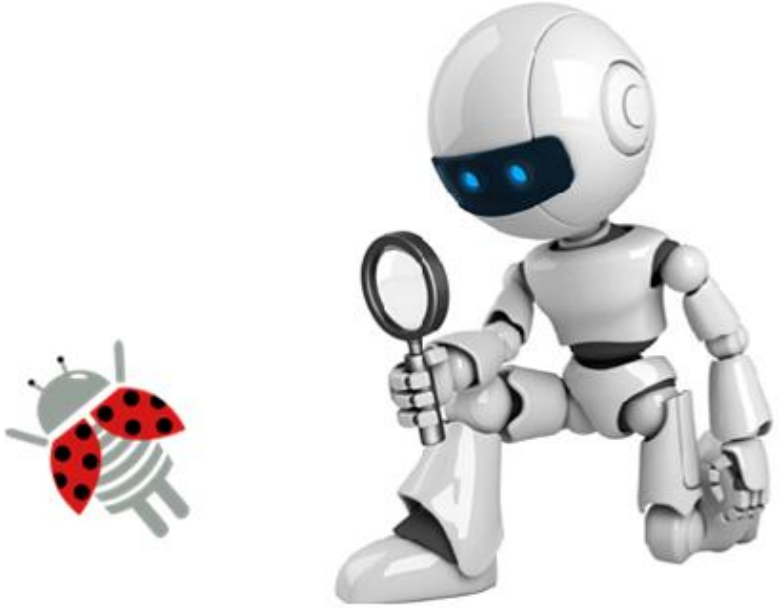
QATestify.com

Test automation course



Lesson 3

(practice)

Table of contents

Recommendations for solving practical challenges

- Use knowledge of two previous lessons and Java basics, studied by yourself
- Break non-obvious tasks into several ones and think how to solve them
- Create your programs with minimum lines of code
- Avoid code duplication. Place frequently used functionality to separate methods
- Be honest with yourself and don’t google the solution. Try to refresh a small piece of code instead. For example, “How to get array length?”

Beginner level

1. Given an array of integer numbers. Create a method (program) which returns a new one where each element is multiplied by 3
2. Given two integers, x and y. Create a method (program) which returns True if one if them is 10 or if their sum is 10
3. Create a method (program) which returns True if the given non-negative number is a multiple of 3 or a multiple of 5
4. Given an array of integers. Create a method (program) which takes two arguments - this array and number that you are looking for - and returns quantity of this number in the array
5. There are several marks in the school: from 1 to 5 (the highest). Create a method (program) which takes a mark and return your result (for ex. "Excellent", "Good", "Failed" etc.). Please do not use if...else condition

Beginner level

6. Print all even numbers from range (0..50). Try to solve this in more than one way
7. Given an array of integers. Find and print the max element in it
8. Given a string and an int n. Return a string made of the first n characters of the string, followed by the first n-1 characters of the string, and so on. Example:

   yourMethod("Testing", 4) → "TestTesTeT"
   yourMethod("Testing", 3) → "TesTeT"
   yourMethod("Testing", 2) → "TeT“

9. Create method which takes two integers (a and b), performs their division (a/b) and outputs accurate result of division (this means the result should have a decimal part, if any)

Beginner level

10. Create method which gets a number form 1 to 9 (for example 4) and then prints the follow output:

For number 4:

```
*
**
***
****
***
**
*
```

For number 3:

```
*
**
***
**
*
```

Pre-intermediate level

1. A couple is going to go to the cinema. That's makes sense, if both of them have the same mood. There are two boolean parameters *heSmile* and *sheSmile* which indicate if each is smiling or not. Write a method which returns True if they're eventually going to go to the cinema, of False otherwise:

   visitCinema(true, true) → true
   visitCinema(false, false) → true
   visitCinema(true, false) → false

2. Given a string. Write a method which swaps first and last characters and returns a new modified string:

   swapChar("Cool text") → "tool texC“

3. Given 2 integers, a and b. Write a method which returns true if one if them is 21 or if their sum is 21.

4. Given an array of integers. Write a method which prints the second half of the array. Include the middle element to output, if the array has odd number of elements

Pre-intermediate level

5. You don't like number 7 and number 9. Write a method which checks input array and returns True if the array doesn't contain your unloved numbers
6. Given an array of integers. Write a method which calculates sum of all elements and arithmetic average of them
7. Given an array of integers. Write a method which finds max and min elements in it and multiplies them by 2
8. Write a method which calculates a sum of first 100 odd elements of Fibonacci sequence
9. Write a method which prints multiplication table for the input number. Like this:

2 x 2 = 4

2 x 3 = 6

2 x 4 = 8

...

Pre-intermediate level

10. The three adjacent digits in the 1000-digit number that have the greatest product are 9 × 9 × 8 = 648.

73167176531330624919225119674426574742355349194934969835203127745063262395783180169848018694788518438586156078911294949545950173795833195285320880551112540698747158523863050715693290963295227443043557668966489504452445231617318564030987115121722383113622298934233803081353362766142828064444866452387493035890729629049156044077239071381051585930796086670172427121883998797908792274921901699720888093776657273330010533678812202354218097512545405947522435258490771167055601360483958644670632441572215539753697817977846174064955149290862569321978468622482839722413756570560574902614079729686524145351004748216637048440319989000889524345065854122758866688116427171479924442928230863465674813919123162824586178664583591245665294765456828489128831426076900422421902267105562632111109370544217506941658960408071984038509624554443629812309878799272442849091888458015616609791913387549920052406368991256071760605886116467109405077541002256983155200055935729725716362695618826704282524836008232575304207529634503

Find the nine adjacent digits in the 1000-digit number that have the greatest product. What is the value of this product?

Pre-intermediate level

11. Given an array of integer positive numbers. For example, {4, 6, 0, 1, 2, 3, 1, 9}, but it can be any random. Write Java-program which returns True, if sequence {1, 2, 3} appears somewhere in the array. Provide additional boundary checks
12. Imagine that you have a special skill which allows to calculate amount of digits in any integer positive number. For example, if number is 12345, amount=15. Write a method which checks your superpower for any hard-coded number.
13. Write a method for calculating quadratic equation. It takes a, b, c numbers and print the solution
14. Write a method which takes a word and prints it like a diagonal matrix. See example:

Input: "Test"

Output: T

e

s

t

Intermediate (middle) level

1. **Superpower** (complicated). Imagine that you have a special skill which allows to calculate amount of digits in any integer positive number. For example, if number is 12345, amount=15. Write a program which checks your superpower **when user enters necessary number via console (in IDE)**. Use error guessing technique for provide additional checks
2. **Lucky tickets.** As you may know tram tickets have numbering in the range from 000 001 to 999 999. Many people try to find a lucky ticket, where the sum of the first three digits = the sum of the second three digits. For example, 205 151 is a lucky ticket. Write a program for calculating of how many lucky tickets people can find in the one roll
3. **Reversed array.** Write a method which takes an array and returns inverted one
4. **Sum of numbers in the string.** Given a string. Write a method which returns the sum of the numbers appearing in the string, ignoring all other characters. Provide all necessary checks. Example:

yourMethod("krm236abw") → 11
yourMethod("aa49b55") → 23

Intermediate (middle) level

5. **Temperature converter.** Write a program for temperature converter between Celsius and Kelvin and vice versa. Use OOP for realizing separate class with appropriate methods and call them in the main class of your program
6. **The longest word.** Write a method which returns the longest word in the input string. Compare only letters separated by space symbol
7. **Addition of matrices.** Add two matrix according to the common math rules and print the output to the console
8. **CamelCaseSeparator.** Some testing frameworks can read names of test methods and include them to the report like a beautiful sentence. As you know, we use camelCaseNamingPolicy during writing our tests. Please write a method which takes a test name and returns beautiful sentence for report. Example:

   Input: "checkChangingProfilePicture“

   Output: "Check changing profile picture"

Advanced (more complicated) level

1. **Password generator.** Apple ID requires a password strong enough to protect your account, data and money. It's difficult to come up a good example of password which meets the following requirements from Apple:

    - should include uppercase and lowercase Latin letters

    - should include digit (-s)

    - should be at least 8 characters.

Please write a program which generates a random password considering the above-mentioned requirements. Choose the max length of password within reasonable limits (for ex. set mas password's length = 15)

2. **Years converter.** Write a program which takes Arabic numerals in range (0..2015) and returns their Romanian analogue. Do not create dual converter, just Arabic > Romanian

Advanced (more complicated) level

3. Happy Test Manager

Pre-conditions:

Your Test Manager is a good guy but unfortunately he doesn't understand how to decide whether the last build is stable for release or it isn't. Let's imagine that he has a test report with the follow data:

- *passed/failed number of test cases (TC)*
- *complete number of unresolved issues*
- *number of unresolved issues (bugs) with HIGH priority*
- *number of unresolved issues (bugs) with Blocker and Critical severity*

The customer demands that every successful release has ***less than 20% failed TC***. In addition he doesn't accept application with ***more than 10 unresolved issues with HIGH priority*** or ***5 unresolved issues with Blocker/Critical severity***. His wife throws in her two cents and insists that number of ***unresolved Blocker issues should be less than 5%*** of complete issues' number

Challenge:

Please help your Test Manager and create a program which will read above-mentioned test report data as a user input, calculate all metrics and print build result (stable or not) to the console. In case if build isn't stable, output some additional explanation for Test Manager.

Additional requirements:

Please realize logic for build metrics calculation and their reference values in the separate class (for ex. Metrics). User input, comparison and the others action should be in the main class