APICAL SERVICES

Agile Developer Recruitment Quiz

Thank you for your interest in this role. At Apical Services, we strive to hire the best of the best. This quiz is your opportunity to put your best foot forward. We think this might take 4-6 hours of your time, but don’t be afraid to put more work into it. We want to see what you are capable of. The type of person we’re looking for will find this an enjoyable challenge!

Instructions

Please answer each question according to the directory structure provided with this document.

There are 2 blocks of questions:

- Questions 1-4 are related to Java
- Questions 5-8 are related to Javascript

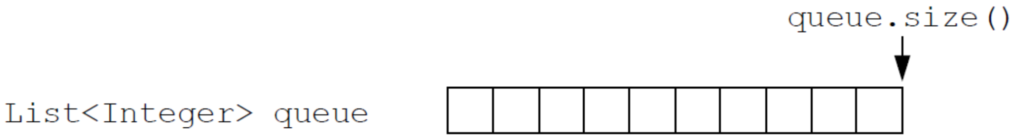
You are expected to complete just one of the blocks above, but if you feel that you can complete both, don't be afraid to do it!
All these questions can be answered in the provided Q#-answer.txt file, as code fragments are sufficient.

Question 9 is a full programming exercise whose outcome should be code that can be compiled (not if you do it Javascript, obviously) and executed. Here is where you can show your best technical skills. You may provide an IDE project (Eclipse, IntelliJ) or, if preferred, a maven or grunt project to build source code. Again, you may add additional information via a Q8-answer.txt file.

Question 1:

A producer thread periodically produces elements and puts them in a queue, and a consumer thread takes the elements from the queue and does some processing with them:

Consumer



Consumer

```
while(true)
{
        ___
        while(___)
        {
            ___
        }
        var el = queue.remove(0);
        ___
        processElement(el);
}
```

Producer

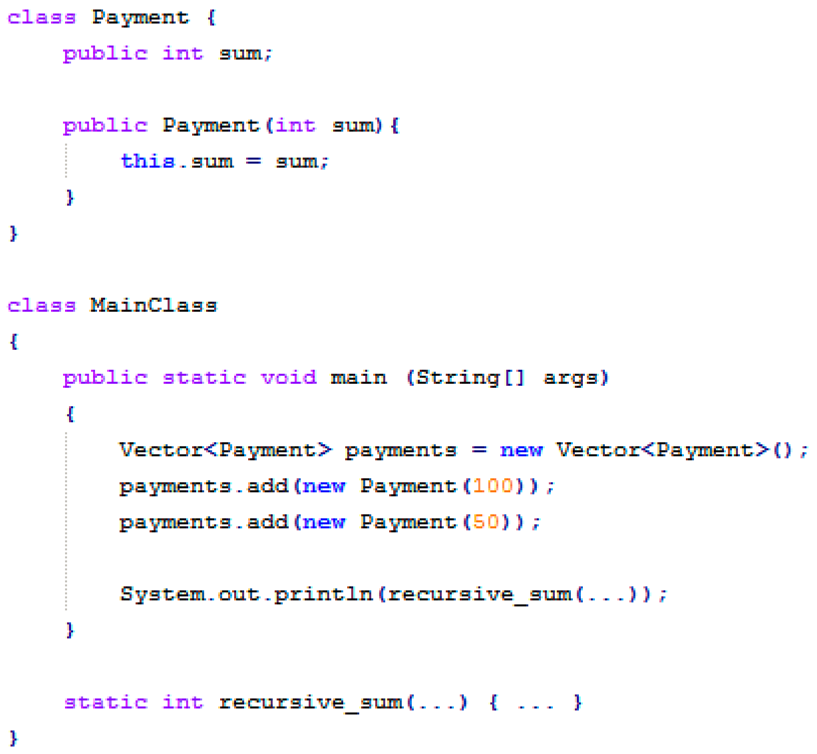
```
timer_tick()
{
        ___
        queue.Add(Math.random());
        ___
        ___
}
```

Question Outcomes

Please write the missing instructions for the producer and consumer threads:

Question 2:

Given the following code:



```
class Payment {
    public int sum;

    public Payment(int sum){
        this.sum = sum;
    }
}

class MainClass
{
    public static void main (String[] args)
    {
        Vector<Payment> payments = new Vector<Payment>();
        payments.add(new Payment(100));
        payments.add(new Payment(50));

        System.out.println(recursive_sum(...));
    }

    static int recursive_sum(...) { ... }
}
```

Question Outcomes

- Please complete the “recursive_sum()” method (recursively)

- What can go wrong with this recursive function if the number of payments becomes really big?

Question 3:

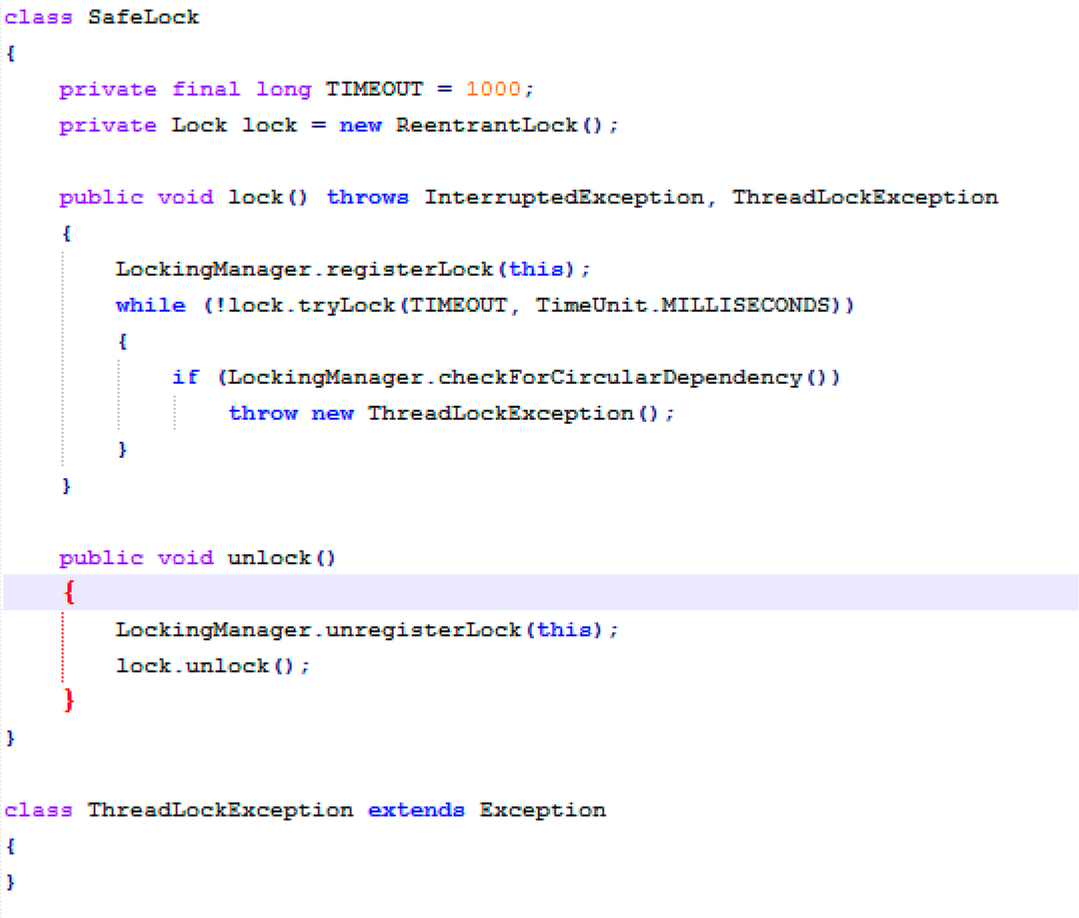
Consider the code presented in folder Q3/src/

Question Outcomes

- Given the unit tests written in TestOption.java, write type Option<T>
- Can you imagine some situations where these types could be useful?

Question 4:

Given the following code:



```
class SafeLock
{
    private final long TIMEOUT = 1000;
    private Lock lock = new ReentrantLock();

    public void lock() throws InterruptedException, ThreadLockException
    {
        LockingManager.registerLock(this);
        while (!lock.tryLock(TIMEOUT, TimeUnit.MILLISECONDS))
        {
            if (LockingManager.checkForCircularDependency())
                throw new ThreadLockException();
        }
    }

    public void unlock()
    {
        LockingManager.unregisterLock(this);
        lock.unlock();
    }
}

class ThreadLockException extends Exception
{
}
```

Question Outcomes

- This class is designed to cope with a problematic situation when dealing with locks, what is it?
- What techniques can be used to avoid running into such a situation?
- What does checkForCircularDependency() possibly do?
- What particular characteristic do the static methods in LockingManager require to work properly?

Question 5

Can you implement the following function?

```
function areAnagrams(str1, str2) {
      // implementation...
}
```

Tip: an anagram is a word, phrase, or sentence formed from another by rearranging its letters. For instance, “Angel” is an anagram of “glean”.

Question 6

Given the following code:

```
var Person = function(name) {
   this.name = name;
};
Person.prototype.getName = function() {
  return this.name;
};
var greet = function(nameProducer) {
   console.log("hello " + nameProducer());
}

var joe = new Person("Joe");
greet(joe.getName);
```

Question Outcomes

- This code prints “hello “ instead of “hello Joe”. Why do you think that happens?
- How would you fix it so that in prints “hello Joe”?

Question 7

Can you write a sum() function that produces the same output when called like this?

```
console.log(sum(2,3));
console.log(sum(2)(3));
```

(the expected output is 5 in both cases)

Question 8

Take a look at the following piece of code:

```
$http.get("http://domain.com?q=1", function(result1)) {
        $http.get("http://domain.com?q=2", function(result2)) {
                $http.get("http://domain.com?q=3", function(result3)) {
                        processResults([result1, result2, result3]);
                };
        };
};
```

It makes 3 HTTP calls, and then passes their results as an argument to a processResults() function.

Can you rewrite it with the following considerations?

- Avoid nested functions
- Try to run the HTTP calls in parallel, not sequentially

Question 9:

Please write a Java console application or a Javascript web application with the following behavior:

1. When the user enters the name of a shape followed by the corresponding number of numeric parameters, define that shape and keep it in memory. The numbers may be of type float. Examples:

```
circle 1.7 -5.05 6.9

square 3.55 4.1 2.77

rectangle 3.5 2.0 5.6 7.2

triangle 4.5 1 -2.5 -33 23 0.3

donut 4.5 7.8 1.5 1.8
```

- For the circle, the numbers are the x and y coordinates of the centre followed by the radius.
- For the square it is x and y of one corner followed by the length of the side.
- For the rectangle it is x and y of one corner followed by the two sides.
- For the triangle it is the x and y coordinates of the three vertices (six numbers in total).
- For the donut it is the x and y of the centre followed by the two radiuses.

In addition, every time such a line is entered, the application should give it a unique identifier and print it out in a standardised form, for example:

```
=> shape 1: circle with centre at (1.7, -5.05) and radius 6.9
```

2. When the user enters a pair of numbers, the application should print out all the shapes that include that point in the (x, y) space, i.e. it should print out shape X if the given point is inside X. (A point is inside a donut shape if it is inside the outer circle but not inside the inner one.)

It should also print out the surface area of each shape found, and the total area of all the shapes returned for a given point.

3. It should accept the commands “help” for printing instructions and “exit” for terminating the execution (“exit” is not necessary in the Javascript app).

4. If the user enters anything unexpected (including errors like too few/many arguments, incorrect number format, etc.), it should print a meaningful error message and continue the execution.

5. Feel free to add additional shapes (e.g. ellipsis) or operations (e.g. to delete a given shape). An advanced option could be to find all the shapes that overlap one that’s named by the user.

6. Allow input from a file as well as the console. It should be possible, for example, to read a file with shape definitions and then to continue with an interactive session.

7. Think about implementing it in a way which would perform well even for a very large number shapes (e.g., tens of millions, but assuming it can still be held in the program memory).

8. Please provide a sample input file for testing.