Taboola
Content You May Like.

R&D Engineer Home Test

Instructions

The purpose of this test is to evaluate your technical abilities in various aspects of Software Engineering, Optimizations, Concurrency Issues and Architecture. Also allowing common grounds for discussion in the interview that takes place after you complete this test.

The test is a home test and you are encouraged to use any resource at your disposal - online, consulting friends etc. You will be asked to explain your solution in the interview, emphasizing on correctness, design and coding decisions.

You should submit via email, working and tested code. The code should be in any language you feel comfortable with (Java is preferred, but definitely not required) and include unit testing.

For any questions/remarks, please feel free to approach Rami Stern rami.s@taboola.com , David Tsur david.t@taboola.com, Ruthy Goldberg ruthy@taboola.com, Lior Chaga lior.c@taboola.com and Tal Sliwowicz tal@taboola.com.

Questions

Question 1

Objective: Implement a text based calculator application. Usage of Rhino, Nashorn and other similar solutions is not allowed.
Input: The input is a series of assignment expressions. The syntax is a subset of Java numeric expressions and operators.
Output: At the end of evaluating the series, the value of each variable is printed out.

Example:
Input: Following is a series of valid inputs for the program:

```
i = 0
j = ++i
x = i++ + 5
y = 5 + 3 * 10
i += y
```

Output:

```
(i=37,j=1,x=6,y=35)
```

Question 2

The following class has several memory and runtime inefficiencies and bugs. Locate and fix as many as you can.

```
import java.util.Date;
import java.util.List;

public class MyClass
{
	private Date m_time;
	private String m_name;
	private List<Long> m_numbers;
	private List<String> m_strings;

	public MyClass(Date time, String name, List<Long> numbers, List<String>
strings) {
		m_time = time;
		m_name = name;
		m_numbers = numbers;
		m_strings = strings;
	}

	public boolean equals(Object obj) {
		if (obj instanceof MyClass) {
			return m_name.equals(((MyClass)obj).m_name);
		}
		return false;
	}

	public String toString() {
		String out = m_name;
		for (long item : m_numbers) {
			out += " " + item;
		}
		return out;
	}

	public void removeString(String str) {
		for (int i = 0; i < m_strings.size(); i++) {
			if (m_strings.get(i).equals(str)) {
				m_strings.remove(i);
			}
		}
	}
	public boolean containsNumber(long number) {
		for (long num : m_numbers) {
			if (num == number) {
				return true;
			}
		}
		return false;
	}

	public boolean isHistoric() {
		return m_time.before(new Date());
	}
}
```

Question 3

Jimmy was tasked with writing a class that takes a base list of strings and a series of transformations and applies them, returning the end result.
To better utilize all available resources, the solution was done in a multi-threaded fashion.
Explain the problems with this solution, and offer 2 alternatives. Discuss the advantages of each approach.

```
import java.util.*;

public class StringsTransformer {

    private List<String> data = new ArrayList<String>();

    public StringsTransformer(List<String> startingData) {
        this.data = startingData;
    }

    private void forEach(StringFunction function) {
        List<String> newData = new ArrayList<String>();
        for (String str : data) {
            newData.add(function.transform(str));
        }
        data = newData;
    }

    public List<String> transform(List<StringFunction> functions) throws
InterruptedException {
        List<Thread> threads = new ArrayList<Thread>();
        for (final StringFunction f : functions) {
            threads.add(new Thread(new Runnable() {
                @Override
                public void run() {
                    forEach(f);
                }
            }));
        }
        for (Thread t : threads) {
            t.join();
        }
        return data;
    }

    public static interface StringFunction {
        public String transform(String str);
    }
}
```

Question 4

You're tasked with writing a spec for a generic local cache with the following property: If the cache is asked for a key that it doesn't contain, it should fetch the data using an externally provided function that reads the data from another source (database or similar).

What features do you think such a cache should offer? How, in general lines, would you implement it?

Question 5

You are tasked with improving the efficiency of a cache heavy system, which has the following properties/architecture:

The system has 2 components, a single instance backend and several frontend instances.

1. The backend generates data and writes it to a relational database that is replicated to multiple data centers.

2. The frontends handle client requests (common web traffic based) by reading data from the database and serving it. Data is stored in a local cache for an hour before it expires and has to be retrieved again. The cache's eviction policy is LRU based.

There are two issues with the implementation above:

1. It turns out that many of the database accesses are redundant because the underlying data didn't actually change.
2. On the other hand, a change isn't reflected until the cache TTL elapses, causing staleness issues.

Offer a solution that fixes both of these problems. How would the solution change if the data was stored in Cassandra and not a classic database?