## 1.通用问题:

* 语言设计中空引用([null reference](http://programmers.stackexchange.com/questions/12777/are-null-references-really-a-bad-thing" \t "/home/duoyi/文档\\x/_blank))的存在有什么问题？假设你想要将空引用的概念从你的首选语言中移除，可能导致什么结果？
* 为什么函数式编程重要？什么时候适用函数式语言？
* 设计(design)、架构(architecture)、功能(functionality)和美学(aesthetic)之间有什么区别？讨论一下。
* 微软、谷歌、欧朋(opera)和火狐这类公司是如何从他们的浏览器中获利的？
* 为什么打开TCP套接字有很大的开销？
* 封装的重要性体现在哪儿？
* 什么是实时系统？它与普通系统有什么区别？
* 实时语言(real-time language)和堆内存分配(heap memory allocation)之间的关系是什么？
* 不变性(Immutability)是指: (变量的)值只能在创建的时候被设置一次，之后就不能被改变。为什么不变性对写更加安全的代码有帮助？
* 可变值(mutable values)和不可变值(immutable values)有哪些优缺点？
* 什么是O/R阻抗失衡(Object-Relational impedence mismatch)？
* 如果你需要使用缓存，你使用哪些原则来确定缓存的大小？
* TCP和HTTP有什么区别？
* 在客户端渲染(client-side rendering)和服务端渲染(server-side rendering)之间，你是如何权衡的？
* 如何在一个不可靠的协议之上构建一个可靠的通信协议？

## 2.开放式问题:

* 为什么人们会抵制变化？
* 如何向你的祖母解释什么是线程？
* 作为一个软件工程师，你想要既要有创新力，又要产出具有可预测性。采用什么策略才能使这两个目标可以共存呢？
* 什么是好的代码？
* 解释什么是流(Streaming)和如何实现一个流？
* 假设你的公司给你一周的时间，用来改善你和同事的生活: 你将如何使用这一周？
* 本周你学了什么？
* 所有的设计中都会有美学元素(aesthetic element)的存在。问题是，你认为美学元素是你的朋友还是敌人？
* 列出最近你读过的5本书。
* How would you introduce Continue Delivery in a multi million waterfall shop?
* 我们来谈谈"重复造轮子","非我发明症", "吃自己做出来的狗粮"的这些做法吧。

(注: 重复造轮子: Reinventing the wheel; 非我发明症:Not Invented Here Syndrome; 吃自己做出来的狗粮: Eating Your Own Dog Food)

* 在你当前的工作流中，什么事情是你计划下一步需要自动化的？
* 为什么写软件是困难的？是什么使软件的维护变得困难？
* 你更喜欢在全新项目（Green Field Project）上工作还是在已有项目(Brown Field Project)基础上工作？为什么？
* [当你在浏览器地址栏输入google.com回车之后都发生了什么?](https://github.com/alex/what-happens-when" \t "/home/duoyi/文档\\x/_blank)
* What does your computer do when you wait?
* 如何向一个5岁的孩子解释什么是Unicode/数据库事务？
* 如何维护单体架构(monolithic architecture)？
* 一个"专业的开发者"意味着什么？
* 软件开发是艺术、是技艺还是工程？你的观点是什么？
* "喜欢这个的人也喜欢..."，如何在一个电子商务商店里实现这种功能？
* 为什么在创新上，企业会比创业公司慢些？

## 3. 代码设计相关问题:

* What is the tests influence on design?
* 内聚和耦合的区别是什么？
* 重构在哪些场景下有用？
* 代码中的注释有用吗？
* 设计和架构有什么区别？
* 为什么在测试驱动开发(TDD)中是先写测试，再写代码？
* C++支持多继承，Java允许类实现多个接口。这些特性对正交性有什么影响？使用多继承和使用多接口有区别吗？[这个问题来自Andrew .Hunt 和 David Thomas写的《程序员修炼之道》]

## 4.web相关问题:

* 为什么"第一方cookie(first-party cookie)"和"第三方cookie(third-party cookie)"被如此不同的对待？

## 5.代码管理相关问题:

* 为什么在Mercurial或者git中(管理)分支比SVN容易？
* 分散式版本控制系统（比如git），相比集中式版本控制系统（如svn）有哪些优势和劣势？

(注:集中式版本控制系统: Centralized Version Control Systems；分散式版本控制系统: Distributed Version Control Systems)

* 能描述一下什么是GitHubFlow和GitFlow工作流吗？
* 什么是rebase？
* 为什么合并操作(merge)在Mercurial和git中比在SVN和CVS中容易？

## 6.并发相关问题:

* 为什么我们需要并发呢？解释一下。
* 为什么测试多线程/并发代码这么困难？
* 什么是竞争条件（Race Condition）？用任何一个语言写一个例子。
* 什么是死锁？用代码解释一下。
* 什么是饿死？
* 什么是Wait-Free算法？

## 7.软件生命周期和团队管理相关问题:

* 什么是敏捷（Agility）？
* 你是如何处理遗留代码（Legacy Code）的？
* 假设我是你们公司的CEO，请向我解释什么是看板，并且说服我在它上面投资。
* 敏捷（Agility）和瀑布（Waterfall）之间的最大区别是什么？
* 作为团队管理者，你对会议太多这个问题是如何处理的？
* 你会如何处理延期很长时间了的项目？

"个体与交互重于过程和工具"和"客户协作重于合同谈判"占了敏捷宣言（Agile Manifesto）的一半，谈论一下这两个观念。

如果你是你们公司的CTO，你会采取什么样的决策？

* 你觉得项目经理有用吗？
* 如果要你组织一个弹性工作制的开发团队（即没有强制工作时间的要求），并且假期制度是"按需休假"，你会如何做？
* 你会如何管理一个人员流动非常高的团队？如何在不加薪的条件下说服团队成员不要离开？
* 除了代码之外，你最关注你的同事的哪3项素质？
* 关于代码，你最希望非技术人员能知道的的三件事是什么？

## 8.安全相关问题:

* 什么是双因素认证(Two Factor Authentication)？在一个已有的Web应用中，你如何实现这种机制？

## 9.比尔盖茨式问题:

* 如果你把一面镜子放在扫描仪上，会发生什么？
* 假设有一个和你完全一样的克隆人，而他是你的上司，你愿意和他工作吗？
* 现在请你面试一下我。
* 为什么Quora上的回答会比Yahoo Answer上的回答好？
* 对手是现代语言，你的任务是要为Cobol辩护，你会如何进行？
* 10年后的你是什么样子？
* 假设你是我老板，我被解雇了。你会如何通知我？
* 我想要重构一个系统，而你想要从头重写。我们来争论一下该怎么弄吧。然后我们反转角色，再争论一下。
* 老板要你对公司撒谎，你的反应是什么？
* 如果你可以穿越到以前，你会给年轻时候的你什么建议？

## 10.代码示例问题:

* 这段Javascript函数的输出是什么？

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | function hookupevents() {    for (var i = 0; i < 3; i++) {      document.getElementById("button" + i)        .addEventListener("click", function() {          alert(i);        });    }  } |

* 关于类型擦除(Type Erasure)，这段Java代码的输出是什么？为什么？

|  |  |
| --- | --- |
| 1  2  3  4 | ArrayList<Integer> li = new ArrayList<Integer>();  ArrayList<Float> lf = new ArrayList<Float>();  if (li.getClass() == lf.getClass()) // evaluates to true    System.out.println("Equal"); |

* 你能指出哪儿有内存泄漏吗？

public class Stack {

    private Object[] elements;

    private int size = 0;

    private static final int DEFAULT\_INITIAL\_CAPACITY = 16;

    public Stack() {

        elements = new Object[DEFAULT\_INITIAL\_CAPACITY];

    }

    public void push(Object e) {

        ensureCapacity();

        elements[size++] = e;

    }

    public Object pop() {

        if (size == 0)

            throw new EmptyStackException();

        return elements[--size];

    }

    /\*\*

     \* Ensure space for at least one more element, roughly

     \* doubling the capacity each time the array needs to grow.

     \*/

    private void ensureCapacity() {

        if (elements.length == size)

            elements = Arrays.copyOf(elements, 2 \* size + 1);

    }

}

* if语句，或者更加通用点，条件表达式通常是过程式编程/命令式编程的形式。你能去掉这段代码中的switch语句，用面向对象的方式来修改这段代码吗？

public class Formatter {

    private Service service;

    public Formatter(Service service) {

        this.service = service;

    }

    public String doTheJob(String theInput) {

        String response = service.askForPermission();

        switch (response) {

        case "FAIL":

            return "error";

        case "OK":

            return String.format("%s%s", theInput, theInput);

        default:

            return null;

        }

    }

}

* 你能去掉这里的if语句，将它改成更加面向对象吗？

public class TheService {

    private final FileHandler fileHandler;

    private final FooRepository fooRepository;

    public TheService(FileHandler fileHandler, FooRepository fooRepository) {

        this.fileHandler = fileHandler;

        this.fooRepository = fooRepository;

    }

    public String Execute(final String file) {

        final String rewrittenUrl = fileHandler.getXmlFileFromFileName(file);

        final String executionId = fileHandler.getExecutionIdFromFileName(file);

        if ((executionId == "") || (rewrittenUrl == "")) {

            return "";

        }

        Foo knownFoo = fooRepository.getFooByXmlFileName(rewrittenUrl);

        if (knownFoo == null) {

            return "";

        }

        return knownFoo.DoThat(file);

    }

}

* 如何重构这段代码？

function()

{ HRESULT error = S\_OK; if(SUCCEEDED(Operation1()))

{ if(SUCCEEDED(Operation2()))

{ if(SUCCEEDED(Operation3()))

{ if(SUCCEEDED(Operation4()))

{

} else {

error = OPERATION4FAILED;

}

} else {

error = OPERATION3FAILED;

}

} else {

error = OPERATION2FAILED;

}

} else {

error = OPERATION1FAILED;

} return error;

}