

Dian团队2021秋季招新笔试

说明部分

时间安排

笔试时间为 2021年9月27日晚 19:30-22:30

提交方式

将答卷 (pdf文件格式) 和自己的最终简历 (pdf文件格式) 压缩打包, 发送至邮箱 hr@dian.org.cn

压缩包命名格式为 选择的题目-学号-姓名, 例如: 2-U202166666-张三

同时请在答卷醒目的位置书写自己的个人信息

注意: 请按规定命名, 未按规定命名提交的试题不予批改!

试题部分

试题说明

1. 从笔试题1、2、3、4、5中**任选一道**作答, 在**完整解答一题**后可以选择写其他题 (请先保证选择的题目的**作答完整**, 多选的题仅仅可作为参考项)
2. **个人独立完成, 允许查阅资料**, 如有答案需要大量复制资料内容, 请**附上答案来源** (网址或书籍页数), 避免被误认为抄袭
3. **严禁抄袭** (一经发现答案雷同, 一同**取消资格**并会被**拉入团队招新黑名单**)
4. 所有年级统一试题。为了体现区分度, 题目设置了阶梯性难度, 对不同年级**评价标准会有不同**, 低年级零基础的同学尽力而为, 尽可能体现自己在整个过程的**思考探索**
5. 若对试题有任何疑问 (如题目有误、题意不明确等问题), 可联系QQ: **1011137273** (2021秋招招新群群主) 进行咨询
6. 笔试结果后续将**通过手机短信**进行通知发送

试题内容

笔试题1

考察方向: **设计模式、面向对象**

关键词: **UML类图**

试题1

请查询相关资料, 回答下列问题。

注意: "[***]" 内为你的回答中**理应涉及到并简要解释的内容**(也提示了你可能需要搜索的内容)

1. 大致了解面向对象思想, 了解任何支持面向对象语言中的类、对象等相关概念**(此题无须作答)**
2. 了解类的继承、函数重写的概念, 谈谈你对虚函数 (抽象函数) 重写的认识[函数重写的使用场景、重写的作用与意义]
3. 学习UML类图的相关概念**(此题无须作答)**

试题2

请根据你学习到的类、对象、UML等的相关知识，回答下列题目。（画图前先阅读附录中有关该题的一些简单约定）

1. 查询资料，了解设计模式中的策略模式。[设计模式、策略模式]

现有情景：某商场打折，现有3种打折方式A、B、C，可用DiscountA, DiscountB, DiscountC表示，针对顾客购买货物的总价，会选择不同的打折方式。

大致伪代码如下，请根据代码中表示的关系与逻辑画出UML图：

```
//示例代码，不用关系细节，仅表示类关系与大致逻辑
//Discount父类，提供函数Count接口，供子类重写
Class Discount
{
public:
    virtual int Count(int price) = 0;
}
//DiscountA类，继承Discount，并重写Count函数，实现对应的打折策略
Class DiscountA: public Discount
{
public:
    int Count(int price);
}
//DiscountB类，继承Discount，并重写Count函数，实现对应的打折策略
Class DiscountB: public Discount
{
public:
    int Count(int price);
}
//DiscountC类，继承Discount，并重写Count函数，实现对应的打折策略
Class DiscountC: public Discount
{
public:
    int Count(int price);
}

//伪代码
//假设输入的price值一定在1-10000之间
int CountNewPrice(int price)
{
    Discount* pDiscount = nullptr; //创建一个Discount对象指针，可以理解为它指向一个Discount对象，但此处指向空
    if(price < 1000)
    {
        pDiscount = new DiscountA;
    }
    else if(price < 5000 && price >=1000)
    {
        pDiscount = new DiscountB;
    }
    else
    {
        pDiscount = new DiscountC;
    }
    return pDiscount->Count(price);
}
```

2. 假设你正在使用C/C++语言学习服务器开发，在了解到一般服务器都具有日志系统后，作为C++程序员，你又开始折腾上了——造轮子，你决定写一个自己的日志系统（采用面向对象的思想），在你的程序里，该日志系统应该是一系列类的组合。请用 UML类图来表达出你的设计。

规定**至少**有如下3个类，并且在设计时尽量使得每个类简洁，功能单一化：

Logger：日志器，作为接口，用于让其他组件使用。

Appender：输出器，用于向固定的文件输出指定内容，一个输出器对应一个文件。

Formatter：格式器，用于将输出进来的日志信息按照事先规定好的格式化输出方式添加相应信息进行格式化。

说明：

仅设计，**无需实现**。但设计过程请**考虑到实现的难易程度**。

每一个**要求**完成后，请**保存**UML图并当作答卷的一部分进行提交，再进行后续的更改。

要求：

1. 为了方便服务器其他组件进行使用，我们会希望有一个 Logger 类，以如下的方式进行调用：

```
//简单调用逻辑代码，非完整代码

//定义日志级别枚举
enum Level
{
    INFO,
    DEBUG,
    WARNING,
    ERROR,
    FATAL
};

Logger logger; //定义一个logger对象
//若干操作，配置logger,此处略去
...

string logInfo("log a log");
logger.log(DEBUG, logInfo); //以DEBUG的日志级别，记录了logInfo日志(一个字符串)
```

2. 因为日志信息除了你需要打印的LogInfo以外，希望可以以固定的格式进行输出，并可以添加相应的信息，比如时间信息等，这个过程我们称之为格式化输出，我们希望能以如下方式进行设置。

```
//设置格式化方式
logger.SetFormatter("[%Y-%M-%D][%T][%m]"); //此处 %加一个字母表示格式化字符，
在函数中会使用相对应的值来替代该内容，无需过多关注
//此处格式化的效果大致如此 [2021-09-27][22:29:59][55555题目好难]，其中%m表示你
输入的LogInfo(一个字符串)
```

3. 我们同时希望能够一次日志可以打印到多处，所以希望对于一个logger，他可以向多处打印，且各处打印希望有不同的格式化输出。

```
//添加打印文件名
logger.AddAppender("/home/Dian/logFile/20210927.log");//此处传入的参数为
string
```

4. 完善功能。在完成上述3题要求后，你现在已经可以如下使用你的日志系统了。请先保存好你当前的UML图作为1-3要求的答案提交！

```
//日志系统的测试
int main(int argc, char* argv[])
{
    Logger logger;//创建一个Logger对象
    logger.SetFromatter("[%Y-%M-%D %T] [%m]");//设置该Logger对象的格式化方式
    logger.AddAppender("/home/Dian/logFile/20210927.log");//添加一个输出器
    logger.AddAppender("/home/Dian/Backup/logFile/20210927.log");//再添
    加一个输出器

    while(true)
    {
        logger.log(DEBUG, "test log system\n");//输出日志
        sleep(1000);
    }
}
```

但为了你的日志系统的易用性，你还有很多路要走，试着完成下列一些功能。

1. 比如你日志器中的Appender无法删除，你需要建立路径与appender的对应关系，保存在适当位置，可以采用如下方式删除某个Appender

```
//删除 输出器
logger.DeleteAppender("/home/Dian/logFile/20210927.log");
```

2. 目前你的所有Appender共用一个Fromatter，若你想定制每个输出器的格式化方式，你该如何设计？
3. 发挥你自己的创意，添加任何你觉得可以使用的上的功能，并在你的UML设计图中画出来（配上适当的文字说明）

笔试题2

考察方向：C、嵌入式

关键词：IIC协议

试题1

通过搜索引擎查找IIC协议相关介绍，简单介绍一下IIC协议，从协议的总线描述，时序图，以及设备主从关系图方面描述一下自己的理解。

试题2

阅读附件TMP75的datasheet，回答以下问题

1. TMP75的用途是什么？
2. 其测量的范围是？精度是多少？测量所需时间有多少？

3. TMP内部有几个寄存器，其**作用**分别是什么？

试题3

编码题，请根据TMP75的数据手册以及自己了解的IIC协议内容，解答以下问题。

假设你现在有一块TMP75且已经正确连接到上位机上。

1. 找出使用IIC协议向TMP75寄存器写数据的**时序图**，以及使用IIC协议从TMP75寄存器读取数据的**时序图**。
2. 假设给出以下接口，请使用这些接口写出初始化TMP75并且从TMP75中读取测量到的数据的代码，补充在main函数中（仅可能地加上注释）

```
//接口说明
typedef enum {
    SDA,
    SCL
} pinType; // IO口种类定义
typedef enum {
    HIGH,
    LOW
} electricalLevel; // 电平高低，HIGH为高电平，LOW为低电平

/**
 * @brief 设置IO口电平
 * @param pin: 目标IO口
 * @param level: 目标电平，高电平或低电平
 */
void writePin(pinType pin, electricalLevel level);

/**
 * @brief 读取目标电平
 * @param pin: 目标IO口
 * @return 返回读取到的IO口电平，高电平或低电平
 */
electricalLevel readPin(pinType pin);

/**
 * @brief 延迟一定时间
 * @param us: 延迟的微秒数
 */
void delayTime(int us);
```

接口的一些使用说明以及main函数如下，请补充main函数内容，完成初始化TMP75并且从TMP75中读取测量到的数据的功能。

```
typedef enum {
    SDA,
    SCL
} pinType; // IO口种类定义
typedef enum {
    HIGH,
    LOW
} electricalLevel; // 电平高低，HIGH为高电平，LOW为低电平

void writePin(pinType pin, electricalLevel level);
electricalLevel readPin(pinType pin);
```

```

void delayTime(int us);

/**
 * @brief 接口使用用例
 */
void testFunc(void) {
    electricalLevel pinLevel;
    writePin(SDA, HIGH);    // 将SDA线设为高电平
    writePin(SCL, HIGH);    // 将SCL线设为高电平
    delayTime(2);           // 延时2us
    pinLevel = readPin(SDA); // 读取SDA线的电平
    writePin(SDA, LOW);     // 将SDA线设为低电平
    return;
}

/**
 * @brief main函数，请编写main函数以完成要求
 */
int main(void) {
    // testFunc(); // 接口使用用例
    return 0;
}

```

笔试题3

考察方向：区块链

关键词：区块链

试题1

请查询相关资料，回答下列问题。

1. 根据查找的资料，列出区块链的特点和核心技术，再简单聊聊自己对于区块链产品的认识。
2. 请简述区块链工作原理,并通过作图表示区块与区块之间的关系。
3. 请列出比特币中每个区块的构成并注明对应字节数。

试题2

自行了解PoW工作机制原理，**阅读区块结构，并根据注释补全程序**。设缺省值为零。

注：写清楚逻辑即可，若暂时没有学过go语言则不需要用go语言完成该题，可通过文字补充说明，不需要考虑复杂区块情景。

(1)区块结构

```

type Block struct {
    PreHash string
    HashCode string
    TimeStamp string
    Diff int
    Data string
    Nonce int
}

```

(2)

```
package main
import (
    "crypto/sha256"
    "encoding/hex"
    "fmt"
    "strconv"
    "strings"
    "time"
)

//定义区块
type Block struct {
    /*add your code here*/
}

//创世区块
func GenerateFirstBlock(data string) Block {
    var firstblock Block
    /*add your code here*/
    return firstblock
}

//计算区块的哈希值
func GenerationHashValue(block Block) string {
    //请仿照比特币的计算方式计算哈希
    var hashdata string
    /*add your code here*/
    var sha = sha256.New()
    sha.Write([]byte(hashdata))
    hashed := sha.Sum(nil)
    return hex.EncodeToString(hashed)
}

//通过pow共识算法生成新区块
func GenerateNextBlock(data string, oldBlock Block) Block {
    var newBlock Block
    /*add your code here*/
    return newBlock
}

//实现简易pow算法
func pow(diff int, block *Block) string {
    for {
        var hash string
        //在循环内写出判断挖矿是否成功的依据
        /*add your code here*/
    }
}

//主函数
func main() {
    //该部分不需要实现
}
```

笔试题4

考察方向：计算机系统

关键词：死锁

试题1

请查询相关资料，回答下列问题。

注意：“[***]” 内为你的回答中**理应涉及到**并简要解释的内容(也提示了你可能需要搜索的内容)

1. 计算机概念中的进程和线程是什么？【进程、线程】
2. 计算机概念中的死锁是什么？造成死锁的条件有哪些？【死锁的产生】
3. 如何避免死锁？
4. 运用相关的数据结构说明如何进行死锁检测
5. 编码一个**死锁情景**（可写伪代码，添加适当的注释），并针对你写的死锁提出解决办法

试题2

请看如下伪代码，回答以下问题

1. **最终结果**是打印 I get r2 还是 I get r1？
2. 说明资源 r1, r2和r3的状态，状态有：**已被占用**（要说明被什么占用），**空闲** 两种
3. 判断是否会产生死锁？如果有请说明理由，并提出解决办法，如果没有，请说明理由

```
//假定每个线程获取资源与释放资源的速度很快（不超过1s）且一致，暂不考虑外界因素
// var 申明变量
var resource(资源) r1;
var resource(资源) r2;
var resource(资源) r3;
// get 获取资源,then获取到资源后执行的代码块,release 释放资源
Thread1(线程1执行的代码块) {
    Thread1 get r1;then{
        Thread1.sleep(1000ms);
        Thread1 get r3;then{
            Thread1.sleep(100ms);
            Thread1 get r2;then{
                //打印
                print("I get r2")
                Thread1 release r1;
            }
        }
    }
}

Thread2(线程2执行的代码块) {
    Thread2 get r3;then{
        Thread2.sleep(5000ms);
        Thread2 get r2;then{
            Thread2.sleep(100ms);
            Thread2 get r1;then{
                //打印
                print("I get r1")
                Thread2 release r3;
            }
        }
    }
}
```



```
}
```

注意：Thread1 和 Thread2两个线程同时启动

笔试题5

考察方向：AI

关键词：机器学习

试题1

请查询资料，按要求回答下列问题，**需要时可结合画图、公式说明**

注意：“**[***]**” 内为你的回答中**理应涉及到**并简要解释的内容(也提示了你可能需要搜索的内容)

1. 机器学习是如何数据驱动的？**[监督学习、回归&分类、训练/验证/测试集各自的作用]**
2. 初步学习线性模型（感知器、逻辑回归、支持向量机等）**（此题无需作答）**
3. 通过以下例子初步认识线性模型的优化：已知四个数据点(x,y): (1,1)、(2,3)、(3,3)、(4,2)，使用**最小二乘法**求解w、b使得 $y=wx+b$ 与上述四点最匹配，并**给出求解过程**
4. 上述例子中，通过最小二乘法来使直线与四个点拟合的误差最小，便是**优化**问题的一个简单例子。请继续学习并解释**神经网络**（可简单理解为多层的线性模型）中，如何通过**损失函数loss**与**梯度下降gradient descent**（或者牛顿法Newton）等**优化**方法进行学习的？**[前向传播，反向传播，如何更新模型]**
5. 众所周知，卷积神经网络CNN才是目前计算机视觉领域的核心算法。对于输入的图像，相比于全连接神经网络对同时对所有位置的特征进行计算，**卷积**会以类似滑动窗口的方式滑过每一块像素并输出特征。试从你**对图像/视觉的理解出发**，谈谈你对CNN在视觉任务中为何得以应用的想法。

试题2

学习目标检测任务中mAP（mean Average Precision）指标，并依次回答以下问题：

1. 目标检测object detection任务是什么，任务目标是如何用数值表示的？
2. TP（True Positive）、TN、FP在**目标检测任务中具体指什么**？**[IoU (Intersection over Union)]**
3. mAP是如何计算的？**[precision、召回率recall、P-R curve]**