# 全国计算机技术与软件专业技术资格（水平）考试

## 2020 年下半年 软件设计师 下午试卷

（考试时间 14:00～16:30 共 150 分钟）

### 请按下述要求正确填写答题卡

1．在答题纸的指定位置填写你所在的省、自治区、直辖市、计划单列市的名称。

2．在答题纸的指定位置填写准考证号、出生年月日和姓名。

3．答题纸上除填写上述内容外只能写解答。

4． 本试卷共 6 题，试题一至试题四是必答题，试题五至试题六选答 1 道。每题 15 分，满分 75 分。

5．解答时字迹务必清楚，字迹不清时，将不评分。

6．仿照下面的例题，将解答写在答题纸的对应栏内。

例题

2020 年下半年全国计算机技术与软件专业技术资格（水平）考试日期是（1）月（2）日。

因为正确的解答是“11 月 7 日”，故在答题纸的对应栏内写上“11”和“7”（参看下表）。

| 例题 | 解答栏 |
|---|---|
| （1） | 11 |
| （2） | 7 |

**试题一（共 15 分）**

阅读下列说明和数据流图，回答问题 1 至问题 4，将解答填入答题纸的对应栏内。

【说明】

某工厂制造企业开发了智能检测系统以有效提升检测效率，节约人力资源，该系统的主要功能包括：

（1）基础信息管理。管理员对检测标准和监控规则等基础信息设置。

（2）检测模型部署。管理员对常用机器学习方法建立检测模型分布。

（3）图像采集。实时将检测多样的产品待检测建分存储，包括产品结构，生产时间，图像信号和产品图像。

（4）缺陷检测。根据检测模型和检测质量标准对图像采集所收到的产品检测信息中所有图像进行检测或所有图像检测合格。若一个产品出现一张图像检测不合格，就表示该产品不合格，对不合格产品，其检测结果包括，产品型号和不合格类型。

（5）质量检测。根据监控规则对产品质量进行监控，将检测情况展示给检测业务员，若不满足条件，向检测业务员发送质量报警，检测是质量发起远程控制部分，向检测设备发送控制指令进行处理。

（6）模型监控。在系统中部署的模型、产品的检测信息结合基础信 息进行监测分析，将模型运行情况发给监控人员。

现采用结构化方法对智能检测系统，进行分析与设计，获得如图 1-1 的上下文数据流图和图 1-2 的数据流图。

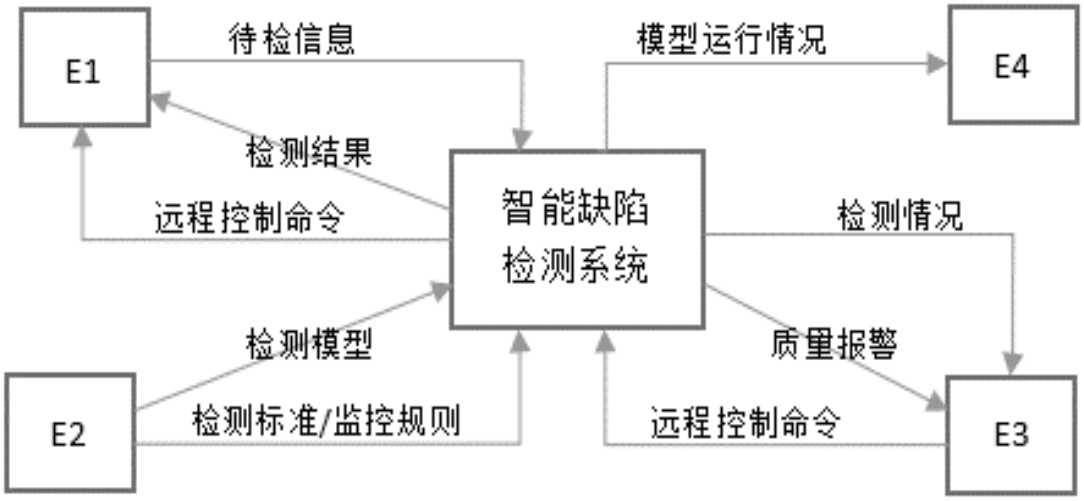


图 1-1

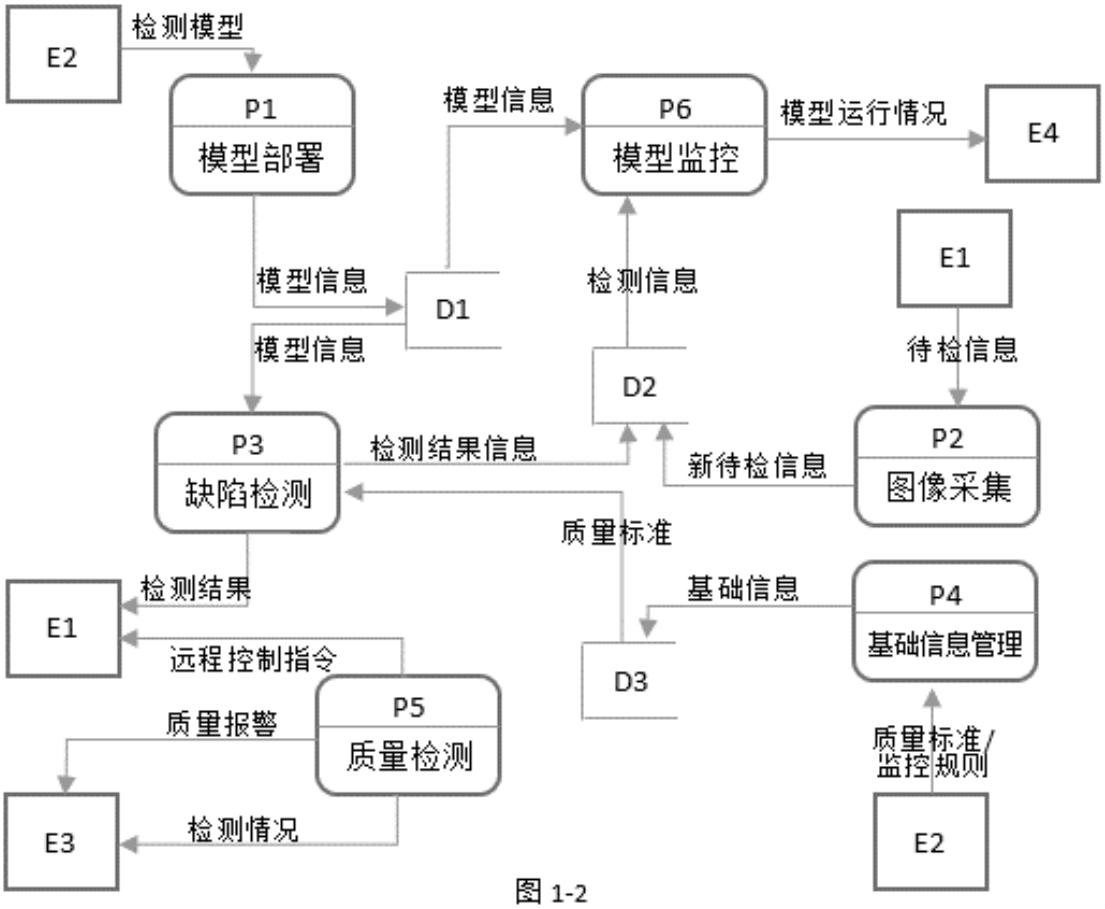


图 1-2

【问题 1】（4 分）

使用说明中的语句给出图 1-1 中的实体 E1～E4 的名称。

【问题 2】（3 分）

使用说明中的语句给出图 1-2 中的数据存储 D1～D3 的名称。

【问题 3】（5 分）

根据说明和图中术语，补齐图 1-2 中缺失的数据及起点和终点。

【问题 4】（3 分）

根据说明，采用结构化语言对缺陷检测的加工逻辑进行描述。

**试题二（共 15 分）**

阅读下列说明，回答问题 1 至问题 4，将解答填入答题纸的对应栏内。

【说明】

M 集团拥有多个分公司，为了方便集团公司对各个分公司职员进行有效管理，集团公司决定构建一个信息平台以满足公司各项业 务管理需求。

【需求分析结果】

（1）分公司关系模式需要记录的信息包括分公司编号、名称、经理号、联系地址和电话。分公司编号唯一标记分公司关系模式中的每一个元组，每个分公司各有一名经理，负责分公司的管理工作，每个分公司设立仅为本分公司服务的多个业务部，业务部包括：研发部、财务部、采购部、交易部等。

（2）业务部关系模式需要记录的信息包括业务部编号、名称、地址、电话和分公司编号。业务部编号唯一标记业务部关系模式中的每 一个元组，每个业务部各有一名主管负责业务部的管理工作，每个业务部有多名职员，每个职员只能来源于一个业务部。

（3）职员关系模式需要记录的信息包括职员号、姓名、所属业务部编号、岗位、电话、家庭成员姓名和成员关系。其中岗位包括：经理、主管、研发员、业务员等。

【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图（不完整）如图 2-1 所示。

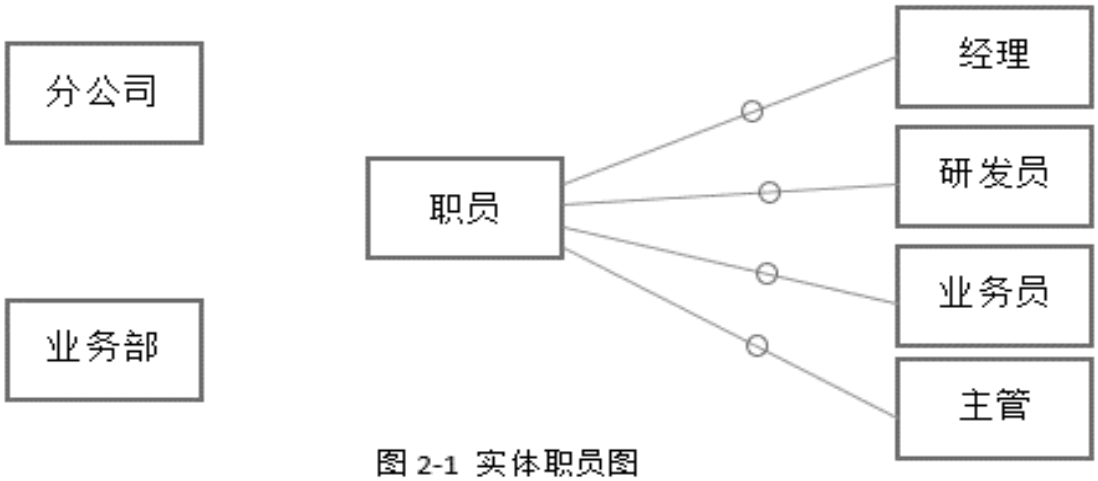


图 2-1 实体职员图

【逻辑结构设计】

根据概念模型设计阶段完成的实体联系图，得出如下关系模式（不完整）：

分公司（分公司编号，名称，__（a）__，联系地址）

业务部（业务部编号，名称，__（b）__，电话），

职员（职员号，姓名，岗位，__（c）__，电话，家庭成员姓名，成员关系）

【问题 1】（4 分）

根据问题描述，补充 4 个联系，完善图 2-1 的实体联系图，联系名可用联系 1、联系 2、联系 3 和联系 4 代替，联系的类型为 1:1、1:n 和 m:n（或 1:1、1:*和*:*）

【问题 2】（3 分）

根据题意将以上关系模式中的空（a）～（c）的属性补充完整，并填入对应位置。

【问题 3】（4 分）

（1）分析分公司关系模式的主键和外键

（2）分析业务部关系模式的主键和外键

【问题 4】（4 分）

在职员关系模式中，假设每个职员有多名家属成员，那么职员关系模式存在什么问题？应如何解决？

**试题三（共 15 分）**

阅读下列说明和 UML 图，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某房产公司欲开发一个房产信息管理系统，其主要功能描述如下：

（1）公司销售的房产（Property）分为住宅（House）和公寓（Cando）两类。针对每套房产，系统存储房产证明、地址、建造年份、建筑面积，销售报价、房产照片以及销售状态（在售，售出，停售）等信息。对于住宅，还需存储楼层、公摊面积、是否有地下室等信息；对于公寓，还需存储是否有阳台等信息。

（2）公司雇佣了多名房产经纪（Agent），负责销售房产，系统中需要存储房产经纪的基本信息，包括：姓名、家庭住址、联系电话、受雇的起止时间等。一套房产同一时段仅由一名房产经纪负责销售，系统中会记录房产经纪负责每套房产的起始时间和终止时间。

（3）系统用户（User）包括房产经纪和系统管理员（Manager），用户需经过系统身份验证之后才能登录系统。房产经纪登录系统之后，可以录入负责销售的房产信息，也可以查询所负责的房产信息。房产经纪可以修改其负责的房产信息，但需要经过系统管理员的审批授权。

（4）系统管理员可以从系统中导出所有房产的信息列表，系统管理员定期将售出和停售的房产信息进行归档，若公司确定不再销售某套房产，系统管理员将该房产信息从系统中删除。

现采用面向对象方法开发该系统，得到如图 3-1 所示的用例图和图 3-2 所示的初始类图。

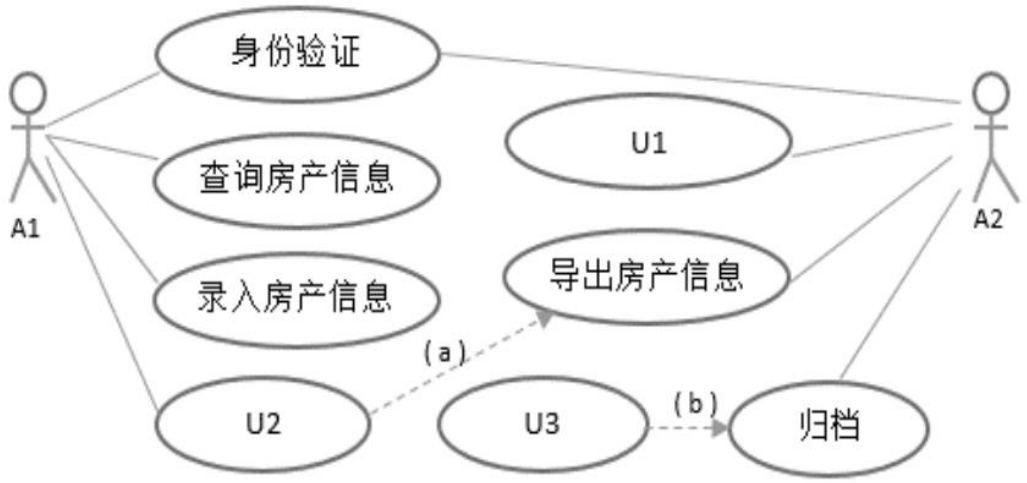


图3-1 用例图

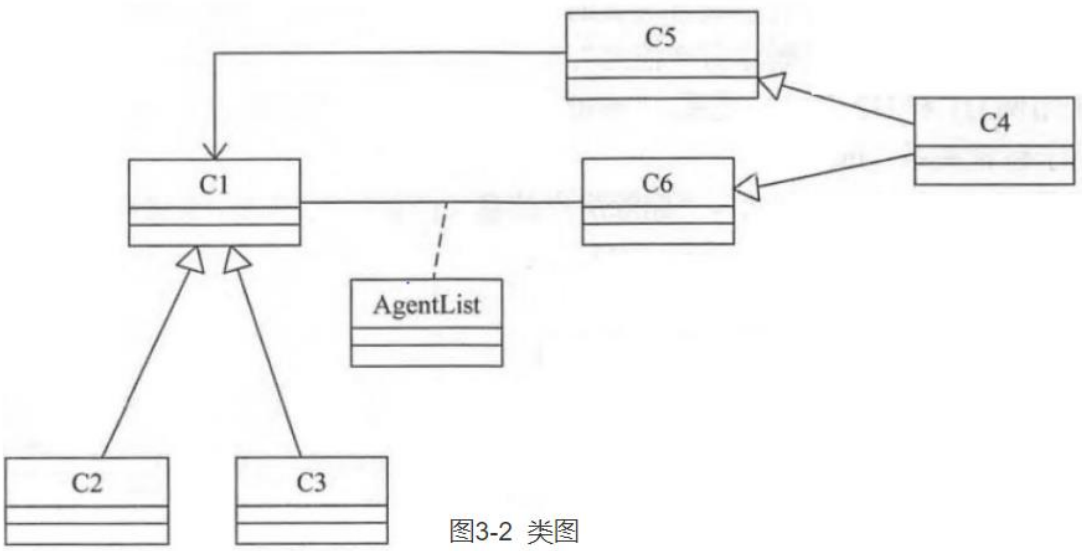

图3-2 类图

【问题 1】（7 分）

（1）根据说明中描述，分别给出图 3-1 中 A1 到 A2 所对应的参与者名称以及 U1 到 U3 所对应的用例名称。

（2）根据说明中描述，分别给图 3-1 中（a）和（b）用例之间的关系。

【问题 2】（6 分）

根据说明中描述，分别给图 3-2 中 C1～C6 所对应的类名称。

【问题 3】（2 分）

图 3-2 中 AgentList 是一个英文名称，用来进一步阐述 C1 和 C6 之间的关系，根据说明中的描述，给出 AgentList 的主要属性。

**试题四（共 15 分）**

阅读下列说明和 C 代码，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

希尔排序算法又称最小增量排序算法，其基本思想是：

步骤 1：构造一个步长序列 delta1，delta2，....，deltak，其中 delta1 = n / 2，后面的每个 delta 是前一个的 1 / 2，deltak = 1；

步骤 2：根据步长序列进行 k 趟排序；

步骤 3：对第 i 趟排序，根据对应的步长 delta，将等步长位置元素分，对同一组内元素在原位置上进行直接插入排序。

【C 代码】

下面的算法用 C 语言实现

（1）常量和变量说明

Data：待排序数组 data，长度为 n，待排序数据在 data[0]，data[1]，data[2]...，data[n - 1]中。

N：数组 data 中的元素个数

delta：步长数组

（2）C 程序

```
#include <stdlib.h>

void ShellSort(int data[], int n) {
    int* delta, k, i, t, dk, j;
    k = n;
    delta = (int*) malloc(sizeof(int) * (n / 2));
    i = 0;
    do {
        __(1)__;
        delta[i ++ ] = k;
    } while __(2)__;
```

```
    i = 0;
    while (  (3)  ) {
        for (k = delta[i]; k < n; ++ k )
            if (data[k] < data[k - dk]) {
                t = data[k];
                for (j = k - dk; j >= 0 && t < data[j]; j -= dk)
                    data[j + dk] = data[j];
                 (4) ;
            }
        ++ i;
    }
}
```

【问题 1】（8 分）

根据说明和 C 代码，填充 C 代码中的空（1）～（4）。

【问题 2】（4 分）

根据说明和 C 代码，该算法的时间复杂度__（5）__ $O(n^2)$（填写小于、等于或大于）。

该算法是否稳定__（6）__（是或否）。

【问题 3】（3 分）

对数组(15，9，7，8，20，-1，4)用希尔排序方法进行排序，经过第一趟排后得到的数组为__（7）__。

**试题六（共 15 分）**

阅读下列说明和 Java 代码，将应填入__（n）__处的字句写在答题纸的对应栏内。

【说明】

在线支付是电子商务的一个重要环节，不同的电子商务平台提供了不同的支付接口。现在需要整合不同电子商务平台的支付接口，使得客户在不同平台上购物时，不需要关心具体的支付接口。拟采用中介者（Mediator）设计模式来实现该需求，所设计的类图如图 6-1 所示。

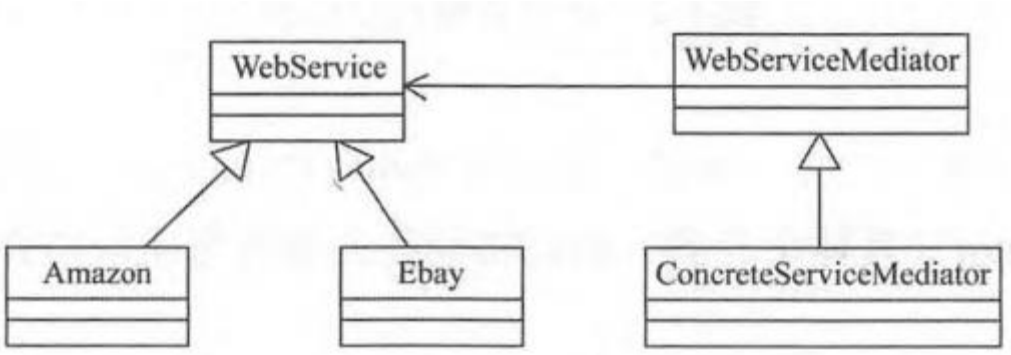


图 6-1　类图

【Java 代码】

```
import java.util.*;

interface WebServiceMediator {
    public __(1)__;
    public void SetAmazon(WebService amazon);
    public void SetEbay(WebService ebay);
}

abstract class WebService {
    protected __(2)__ mediator;
    public abstract void SetMediator(WebServiceMediator mediator);
    public __(3)__;
    public abstract void search(double money);
}
```

```
class ConcreteServiceMediator implements WebServiceMediator {
    private WebService amazon;
    private WebService ebay;

    public ConcreteServiceMediator() {
        amazon = null;
        ebay = null;
    }

    public void SetAmazon(WebService amazon) {
        this.amazon = amazon;
    }

    public void SetEbay(WebService ebay) {
        this.ebay = ebay;
    }

    public void buy(double money, WebService service) {
        if (service == amazon)
            amazon.search(money);
        else
            ebay.search(money);
    }
}
```

```
class Amazon extends WebService {
    public void SetMediator(WebServiceMediator mediator) {
        this.mediator = mediator;
    }

    public void buyService(double money) {
        __(4)__;
    }

    public void search(double money) {
        System.out.println("Amazon receive: " + money);
    }
}

class Ebay extends WebService {
    public void SetMediator(WebServiceMediator mediator) {
        this.mediator = mediator;
    }

    public void buyService(double money) {
        __(5)__;
    }

    public void search(double money) {
        System.out.println("Ebay receive: " + money);
    }
}
```