

我们公司的ASP.NET 笔试题,你觉得难度如何 - 文章 - 伯乐在线



本套试题共8个题,主要考察C#面向对象基础,SQL和ASP.NET MVC基础知识。

第1-3题会使用到一个枚举类,其定义如下:

```
C#  
public enum QuestionType  
{  
    Text = 0,  
    MultipleChoice = 1  
}
```

第1题:请定义一个接口IQuestion,有【标题】和【问题种类】两个属性,其中【问题种类】是只读的枚举类型QuestionType,另外还有一个方法获取该问题的答案(无参,返回字符串)。

第2题:请定义一个抽象类QuestionBase,实现第一题中的IQuestion接口,其中【问题种类】属性不在该抽象类中实现,而留在该抽象类的子类中实现;获取答案的方法有默认实现,返回字符串“默认答案”。

第3题:请定义一个TextQuestion类,继承自第2题中的QuestionBase;获取答案的方法返回字符串“文本答案”。再定义一个MultipleChoiceQuestion类,继承自第2题中的QuestionBase;获取答案的方法返回字符串“单选答案”。

第4题:假设有实体类Product定义如下:

```
C#  
public class Product  
{  
    public string Name { get; set; }  
    public bool IsDeleted { get; set; }  
}
```

现在有一个方法从IQueryable中获取没有删除的Product列表,该方法实现如下:

```
C#  
public List GetActiveProducts(IQueryable query)  
{  
    return query.WhereNotDeleted().ToList();  
}
```

```
}
```

请完成扩展方法: WhereNotDeleted

第5题: 假设数据库中有User和Income两张表如下, 请仔细分析下方的示例数据, 然后写出SQL得到右方的查询结果。

User表	
Id	Name
1	Leo
2	Mike

Income表				
Id	UserId	Amount	Year	Month
1	1	100	2014	1
2	1	200	2014	1
3	1	200	2014	2
4	2	100	2014	2
5	2	200	2014	3

查询结果			
Name	Year	Month	Income
Leo	2014	1	300
Leo	2014	2	200
Mike	2014	2	100
Mike	2014	3	200

第6题: 根据第5题的数据结构, 有如下两个实体类和查询结果类的定义:

C#

```
public class User
{
    public int Id { get; set; }
    public string Name { get; set; }
}

public class Income
{
    public int Id { get; set; }
    public int UserId { get; set; }
    public decimal Amount { get; set; }
    public int Year { get; set; }
    public int Month { get; set; }
}

public class UserIncomeDto
{
    public string Name { get; set; }
    public int Year { get; set; }
    public int Month { get; set; }
    public decimal Income { get; set; }
}
```

现有一个方法用LINQ的方式得到第5题的查询结果, 该方法定义如下:

C#

```
public List GetUserIncomeDtos(IQueryable users, IQueryable incomes)
{
    throw new NotImplementedException();
}
```

请完成该方法的实现。

第7题：在ASP.NET MVC应用程序中，假设有如下HTML表单：

C#

```
<form action="/admin/mobile/user/login">
    <input type="text" placeholder="username"/>
    <input type="password" placeholder="password"/>
    <input type="submit" value="login"/>
</form>
```

当该表单同步提交的时候，如何修改以上HTML和路由配置以使该请求进入下方的action中：

C#

```
public class UserController : Controller
{
    [HttpPost]
    public ActionResult Login(string username, string password)
    {
        throw new NotImplementedException();
    }
}

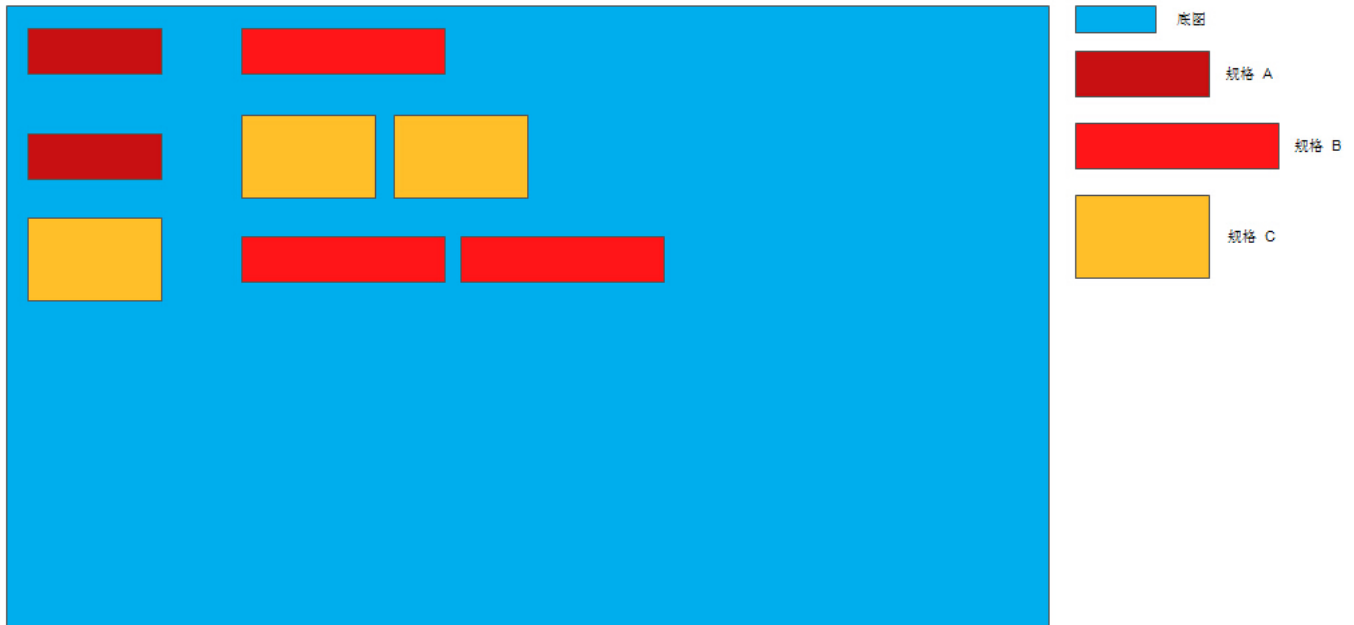
}C#

public class Product
{
    public string Name { get; set; }
    public string Description { get; set; }
    public void Validate1()
    {
        if (string.IsNullOrEmpty(this.Name))
        {
            throw new Exception("please enter a name for the product");
        }
        if (string.IsNullOrEmpty(this.Description))
        {
            throw new Exception("product description is required");
        }
    }
    public void Validate2()
    {
        this.Require(x => x.Name, "please enter a name for the product");
        this.Require(x => x.Description, "product description is required");
    }
}
```

请完成Validate2方法中Require方法的定义和实现，从而使得Validate2与Validate1方法实现同样的效

果。

第9题: 现有蓝色底的纸板, 可以自定义长宽(W*H), 定义好之后不能修改。现在提供N种规格的矩形小纸板, A ,B ,C ,D 实际上最多不超过10种规格, 选择其中的1种 或者 N种规格的纸板填充到底板上, 如下图所示:



为了计算出最佳的选择方案: 比如A 5个 C 1个 D 2个 使得底板的填充率最高, 请提供具体的算法思路。

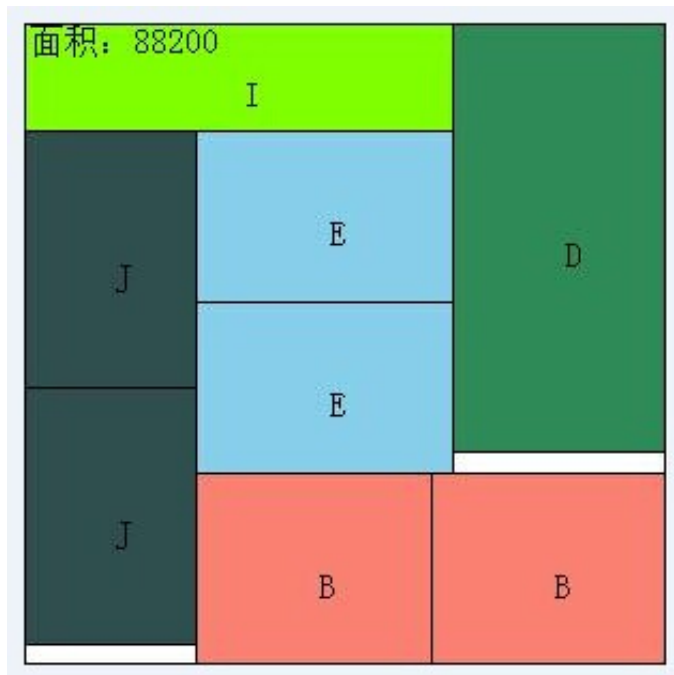
提示: 比如下面这样的输入和输出:

```
var background = new Block("Background", 300, 300);

var blocks = new List<Block>();

blocks.Add(new Block("A", 90, 110, Color.SaddleBrown));
blocks.Add(new Block("B", 110, 90, Color.Salmon));
blocks.Add(new Block("C", 200, 200, Color.SandyBrown));
blocks.Add(new Block("D", 100, 200, Color.SeaGreen));
blocks.Add(new Block("E", 120, 80, Color.SkyBlue));
blocks.Add(new Block("F", 400, 300, Color.SlateBlue));
blocks.Add(new Block("G", 250, 150, Color.Yellow));
blocks.Add(new Block("H", 170, 150, Color.Blue));
blocks.Add(new Block("I", 200, 50, Color.Chartreuse));
blocks.Add(new Block("J", 80, 120, Color.DarkSlateGray));
```

输出:



拿高薪，还能扩大业界知名度！优秀的开发工程师看过来 -> 《[高薪招募讲师](#)》

1 赞 2 收藏 [1 评论](#)



合作联系

Email: bd@jobbole.com

QQ: 2302462408 (加好友请注明来意)

更多频道

[小组](#) - 好的话题、有启发的回复、值得信赖的圈子

[头条](#) - 分享和发现有价值的内容与观点

[相亲](#) - 为IT单身男女服务的征婚传播平台

[资源](#) - 优秀的工具资源导航

[翻译](#) - 翻译传播优秀的外文文章

[文章](#) - 国内外的精选文章

[设计](#) - UI, 网页, 交互和用户体验

[iOS](#) - 专注iOS技术分享

[安卓](#) - 专注Android技术分享

[前端](#) - JavaScript, HTML5, CSS

[Java](#) - 专注Java技术分享

[Python](#) - 专注Python技术分享