

# Q1.在有了规范的开发文档前提下，代码注释的必要性，详细程度？（如何写注释）

---

## 1.代码注释的必要性

---

- 在团队中，我们写的程序不只是给自己看的，而是大家共同开发，如果没有注释，代码理解起来比较让人头疼。
- 软件开发通常是一个长期的过程，有时候过了很长时间再看自己的代码，如果没有注释，理解起来也很困难。
- 如果离开公司，新的继任者接手自己的工作，如果没有足够多的文档和代码注释，接手难度会很大。
- 代码注释是写代码的人和读的代码的人的通信桥梁，最大限度的提高团队开发合作效率。
- 代码注释是程序代码可维护性的重要环节之一。
- 代码没注释，看起来比较恶心，久而久之代码就会发臭。
- 代码注释可以让人更直观的理解代码功能，达到高效开发的目的。

## 2. 详细程度

---

一般情况下，源程序的有效注释必须在总代码行数的百分之20以上。注释的原则是有助于对程序的阅读理解，在该加的地方都加上，注释不宜太多。也不能太少。注释语言必须准确、易懂、简洁。

### 2.1 注释的原则

- 注释形式统一：按照项目文档规定，严格按照统一格式书写代码和注释；
- 注释内容简单明确：注释语言必须准确，简单明了，避免二义性或者错误的注释；
- 代码修改后注释一定也要相应的跟着修改；
- 必须加的基本注释：类和接口的注释，构造函数的注释；方法的注释；全局变量的注释；字段属性的注释。
- 在一些循环和逻辑分支中要加注释：因为有些循环逻辑分支较长，读起来比较费力，如果加一些说明性的注释可以使读者读起来更加友好。

### 2.2 如何写注释

按照开发文档的要求写代码和注释，并且开发团队严格依照文档上面的格式。

注释分为单行注释，快注释和文档注释，在java中的注释格式分别为：

- 单行注释：`// + 注释内容`
- 块注释：`/* 注释内容 */`
- 文档注释：`/** 注释内容 */`

#### 2.2.1 类和接口的注释

- 类功能函数说明
- 作者
- 最后修改人
- 最后修改日期

### 2.2.2 构造方法的注释

- 详细的参数说明

### 2.2.3 方法的注释

- 必要的参数说明，包括传入参数和返回参数
- 函数功能说明
- 一些函数使用的说明，如说明函数中可能存在的风险
- 作者，修改人以及修改日期

### 2.2.4 全局变量的注释

- 作用域
- 参数解释

### 2.2.5 字段/属性的注释

- 参数解释

## 3. 总结

一般在项目文档确定的情况下，关于代码和注释的规范也相应确定下来了，团队中每个人都按照这个格式写注释，大家看起来赏心悦目，会为项目开发提供不少的帮助。在个人实际的编程体验中，我也深刻的体会到代码注释的必要性，自己一周之前写的代码拿到现在看就得重新梳理一遍，其实如果有了注释，我想这个梳理时间可能会大大减少，增加自己coding的效率。所以，在接下来的编程过程中，我会逐渐养成写注释的习惯，以使得自己的代码更加易读。代码如果没有注释就会发臭，希望自己以后的代码永远香甜可，勉励自己多多写有效的注释。

## 4. 实际项目的代码注释

19年暑假，我在一个公司实习，我所在的开发小组主要编写前后端数据接口，主要用到了springboot 和 mybatis，接口请求采用RESTfull风格，注释采用 swagger 注释。下面是部分代码和注释：

开发工作完全是按照开发说明书进行，注释采用swagger风格，要求对每个接口功能和参数做必要说明，注明作者和日期，函数名统一规范: (get, delete, update, insert) + 有意义的名称，少做或不做额外注释。

```
@Api(tags = "7其它功能接口")
@RestController
public class MoneyAccountController {
    @Autowired
    IMoneyAccountService moneyAccountService;
    /**
     * @author 后永胜
     * @since 2019-07-05
     */
    @ApiOperation(value = "7.3款项入账信息")
    @ApiImplicitParams({
        @ApiImplicitParam(name="propertyBasicId",value="财物关键字",required = true ,
dataType="String", paramType = "query")
    })
    @RequestMapping(value="/BasicQuery/Money_AccountDetail",method = RequestMethod.GET)
    public AjaxResult MoneyAccountDetail(String propertyBasicId) {
        AjaxResult res = new AjaxResult();
        if(propertyBasicId == null || propertyBasicId=="") {
```

```

        res.setCode(ResultState.Error);
        res.setMsg(FrontMessage.ParamError);
        return res;
    }else {
        res = moneyAccountService.getMoneyAccountDetail(propertyBasicId);
        return res;
    }
}
}
}

```

```

/**
 * <p>
 * Mapper 接口
 * </p>
 *
 * @author 王杰
 * @since 2019-07-04
 */
@Mapper
public interface MoneyAccountMapper extends BaseMapper<MoneyAccount> {
    // @author 后永胜 @since 2019-07-05
    MoneyAccount getMoneyAccountDetailByParam(@Param("propertyBasicId") String propertyBasicId);

    // @author 王杰 @since 2019-07-09
    MoneyAccount getMoneyAccountRecord(@Param("propertyBasicId") String propertyBasicId);

    // @author 王杰 @since 2019-07-09
    List<PropertyItem> selectMoneyAccountPropertyItemById(@Param("propertyBasicId") String propertyBasicId);
}

```

```

/**
 * <p>
 * 服务类
 * </p>
 *
 * @author 王杰
 * @since 2019-07-04
 */
public interface IMoneyAccountService extends IService<MoneyAccount> {
    /**
     * @param propertyBasicId
     * @return
     * @author 后永胜
     * @since 2019-07-05
     */
    public AjaxResult getMoneyAccountDetail(String propertyBasicId);
}

```

```

/**
 * <p>
 * 服务实现类
 * </p>
 *
 * @author 王杰

```

```
* @since 2019-07-04
*/
@Service
public class MoneyAccountServiceImpl extends ServiceImpl<MoneyAccountMapper, MoneyAccount>
implements IMoneyAccountService {
    @Autowired
    private MoneyAccountMapper moneyAccountMapper;

    /**
     * @author 后永胜
     * @since 2019-07-05
     */
    @Override
    public AjaxResult getMoneyAccountDetail(String propertyBasicId) {
        // TODO Auto-generated method stub
        AjaxResult res = new AjaxResult();
        MoneyAccount data = moneyAccountMapper.getMoneyAccountDetailByParam(propertyBasicId);

        res.setCode(ResultState.Correct);
        res.setData(data);
        res.setMsg(FrontMessage.SearchSuccess);
        return res;
    }
}
```