**哈尔滨工业大学（威海）**

Java语言实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 课程名称 | Java程序设计 | 课程编号 | SE33601 |
| 实验名称 | 实验1：类的定义及继承的应用 | | |
| 实验类型 | 验证与编程 | 实验学时 | 2学时 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 班号 | 191110323 | 学号 | 2191110323 | 姓名 | 张凯文 |
| 开始 | 2020.10.09 | 截止 | 2020.10.14 | 成绩 |  |
| 批语 |  | | | | |

目录

[1 实验目的、内容和要求 1](#_Toc53126212)

[1.1 实验目的 1](#_Toc53126213)

[1.2 实验内容 1](#_Toc53126214)

[1.3 实验要求 1](#_Toc53126215)

[2 实验内容 1](#_Toc53126216)

[2.1 需求描述 1](#_Toc53126217)

[2.2 具体类设计 2](#_Toc53126218)

[2.3 重构：利用继承消除重复代码 3](#_Toc53126219)

# 1 实验目的、内容和要求

## 1.1 实验目的

1. 掌握单个类、类的属性和方法的设计方法
2. 掌握继承的基本特性和超类与子类的定义方法
3. 能够利用StarUML绘制简单类图

## 1.2 实验内容

1. 下载、安装Eclipse/intellij Idea; StarUML。
2. 利用StarUML进行类的设计，并使用快速开发工具Eclipse或者intellj idea实现代码；

## 1.3 实验要求

1. 除了增删表格的行外，不允许改变本文档的页边距及其他格式；
2. 程序代码要求直接从快速开发工具中拷贝代码，并粘贴到本报告相应的表格内，调整程序代码采用consolas字体，五号，去除多余空行，并注意缩进等基本格式；
3. 实验结果可以截图
4. 由于每个同学写的程序长短不一，必然导致页码的变换，因此在正式提交本文档前，需要在目录部分，右键更新目录的页码，否则会出现“未定义标签”错误。

# 2 实验内容

## 2.1 需求描述

为了满足不同用户数据量的要求，某软件产品拟采用多种数据库管理系统对数据进行管理，目前采用MySQL，Oracle两种数据库。

数据库的连接操作严格按照如下顺序进行：

* MySQL的数据库连接操作
  1. 注册MySQL驱动（Registration MySQL Driver）
  2. 建立MySQL连接（Establish MySQL Connection）
  3. 创建执行语句（Create Execute Statements）
  4. 执行语句（Execute Statement）
  5. 处理结果（Processing Results）
  6. 释放资源 （Free Resource）
* Oracle的数据库连接操作

1. 注册Oracle驱动（Registration Oracle Driver）
2. 建立Oracle连接（Establish Oracle Connection）
3. 创建执行语句（Create Execute Statements）
4. 执行语句（Execute Statement）
5. 处理结果（Processing Results）
6. 释放资源 （Free Resource）

要求：根据以上流程，为公司的数据库连接编写代码框架，实现两种数据库的连接（在方法中打印一条提示语句即可，不必真正编写数据库的相关代码）。为设计和实现方便，假设以上数据库的连接操作步骤中没有注入参数，也没有返回值。

## 2.2 具体类设计

* **实验内容1：参考讲义中《咖啡与茶冲泡机案例》和2.1节需求的描述，对Mysql，Oracle两种数据库的连接进行设计**。

1. **使用StarUML软件绘制具体类的类图。**

|  |
| --- |
|  |

根据以上设计的类图，完成各具体类的代码，并按照1.3 实验要求的格式粘贴到下方对应表格中。

1. **MySQL的数据库连接操作类代码：**

|  |
| --- |
| public static class MySQL{  public void CreateExecute() {  System.out.println("Create Execute Statements");  }  public void ExecuteStatements() {  System.out.println("Execute Statement");  }  public void Processing(){  System.out.println("Processing Results");  }  public void Free(){  System.out.println("Free resource");  }  public void Registration\_MySQL(){  System.out.println("Registration MySQL Driver");  }  public void Establish\_MySQL(){  System.out.println("Establish MySQL Connection");  }  public void Prepare(){  Registration\_MySQL();  Establish\_MySQL();  CreateExecute();  ExecuteStatements();  Processing();  Free();  }  } |

1. **Oracle的数据库连接操作类代码：**

|  |
| --- |
| public static class Oracle {  public void CreateExecute() {  System.out.println("Create Execute Statements");  }  public void ExecuteStatements() {  System.out.println("Execute Statement");  }  public void Processing(){  System.out.println("Processing Results");  }  public void Free(){  System.out.println("Free resource");  }  public void Registration\_Oracle(){  System.out.println("Registration Oracle Driver");  }  public void Establish\_Oracle(){  System.out.println("Establish Oracle Connection");  }  public void Prepare(){  Registration\_Oracle();  Establish\_Oracle();  CreateExecute();  ExecuteStatements();  Processing();  Free();  }  } |

1. **测试类的代码：**

|  |
| --- |
| public static void main(String args[]){  MySQL mysql = new MySQL();  mysql.Prepare();  System.out.println("=============");  Oracle oracle = new Oracle();  oracle.Prepare();  } |

1. **运行结果截图：**

|  |
| --- |
|  |

## 2.3 重构：利用继承消除重复代码

* **实验内容2：为达到代码复用的目的，需要对本报告第2.2节中的设计进行优化，对具体类进行抽象，从而形成继承结构，利用继承的特性消除重复代码。**

1. **类图绘制如下：**

|  |
| --- |
|  |

根据以上重构后的类图，完成各具体类的代码，并按照1.3 实验要求的格式粘贴到下方对应表格中。

1. **重构后的超类代码实现**

|  |
| --- |
| public static class Commons {  public void CreateExecute() {  System.out.println("Create Execute Statements");  }  public void ExecuteStatements() {  System.out.println("Execute Statement");  }  public void Processing(){  System.out.println("Processing Results");  }  public void Free(){  System.out.println("Free resource");  }  } |
|  |

1. **重构后的MySQL数据库连接操作类代码：**

|  |
| --- |
| public static class MySQL extends Commons {  public void Registration\_MySQL(){  System.out.println("Registration MySQL Driver");  }  public void Establish\_MySQL(){  System.out.println("Establish MySQL Connection");  }  public void Prepare(){  Registration\_MySQL();  Establish\_MySQL();  CreateExecute();  ExecuteStatements();  Processing();  Free();  }  } |

1. **重构后的Oracle数据库连接操作类代码：**

|  |
| --- |
| public static class Oracle extends Commons {  public void Registration\_Oracle(){  System.out.println("Registration Oracle Driver");  }  public void Establish\_Oracle(){  System.out.println("Establish Oracle Connection");  }  public void Prepare(){  Registration\_Oracle();  Establish\_Oracle();  CreateExecute();  ExecuteStatements();  Processing();  Free();  }  } |

1. **重构后的测试类代码：**

|  |
| --- |
| public static void main(String args[]){  MySQL mysql = new MySQL();  mysql.Prepare();  System.out.println("=============");  Oracle oracle = new Oracle();  oracle.Prepare();  } |

1. **运行结果截图：**

|  |
| --- |
|  |