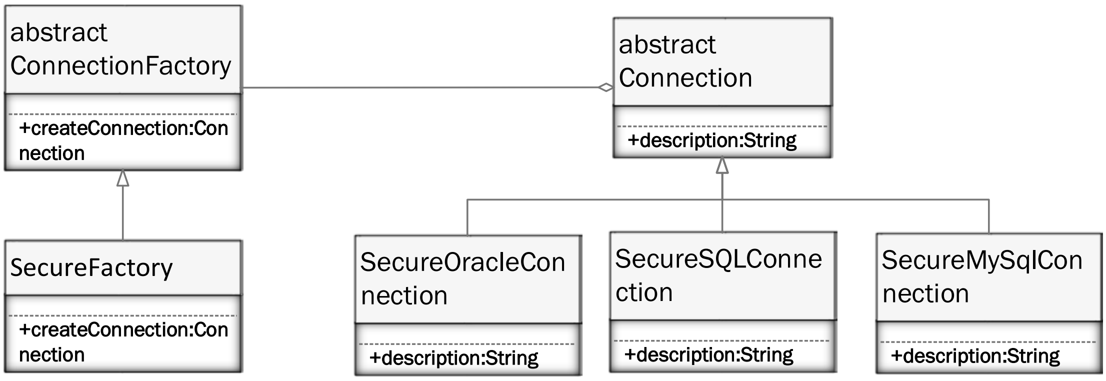
设计题（共 50 分）

1、利用工厂方法模式实现下面的实例：某软件产品开发，需要我们实现针对三种不同的数据库建立安全连接 (secure connection) ， 数据库类型分别是 Oracle, SQL, MySql， 每一个安全链接建立以后：

- **Oracle 数据库的安全链接类叫做 SecureOracleConnection, 其 description 方法需要输出字符串 “Oracle Secure”**
- **SQL 数据库的安全链接类叫做 SecureSQLConnection，其 description 方法需要输出 字符串 “SQL Secure”**
- **MySql 数据库的安全链接类叫做 SecureMySqlConnection，其 description 方法需要输出字符串 “MySql Secure”**

请画出 UML 类图并给出相应的示意性代码，并给出你所写的程序的测试代码，并说明该模式体现了哪些面向对象设计的基本原则。（25 分）

解答：



```
//以下定义两个抽象类
public abstract class Connection
{
    public Connection(){}
    public String description() {return “Generic”; }
}
public abstract class ConnectionFactory
{
    public ConnectionFactory() {}
    protected abstract Connection createConnection{String type);
}

//以下定义三种数据库的安全链接对象
public class SecureOracleConnection extends Connection
{   public SecureOracleConnection(){}
   public String description()
   {return “Oracle secure”; }
}
```

```
public class SecureSQLConnection extends Connection
{   public SecureSQLConnection(){}
  public String description () {return “SQL secure”;}
}
public class SecureMySqlConnection extends Connection
{   public SecureMySqlConnection() {}
  public String description (){return “MySql secure”; }
}
//以下定义这个工厂
public class SecureFactory extends ConnectionFactory
{   public Connection createConnection（String type)
 {
    if(type.equals(“Oracle”)){return                        new
SecureOracleConnection();}
    else if(type.equals(“SQL”)) ”))
    {
       return new SecureSQLConnection();
    }
    else {return new SecureMySqlConnection();}
  }
}

//测试代码
public class TestFactory
{   public static void main(String args[])
   {
     SecureFactory factory;
     factory = new SecureFactory();
     Connection connection1=factory.createConnection(“Oracle”);
     Connection connection2=factory.createConnection(“SQL”);
     Connection connection3=factory.createConnection(“MySql”);
     System.out.println(“connecting
                        with “+ connection1.description());
     System.out.println(“connecting
                        with “+ connection2.description());
     System.out.println(“connecting
                         With “+ connection3.description());
   }
}
```

工厂方法模式体现了开放一闭合原则， 和依赖倒转原则