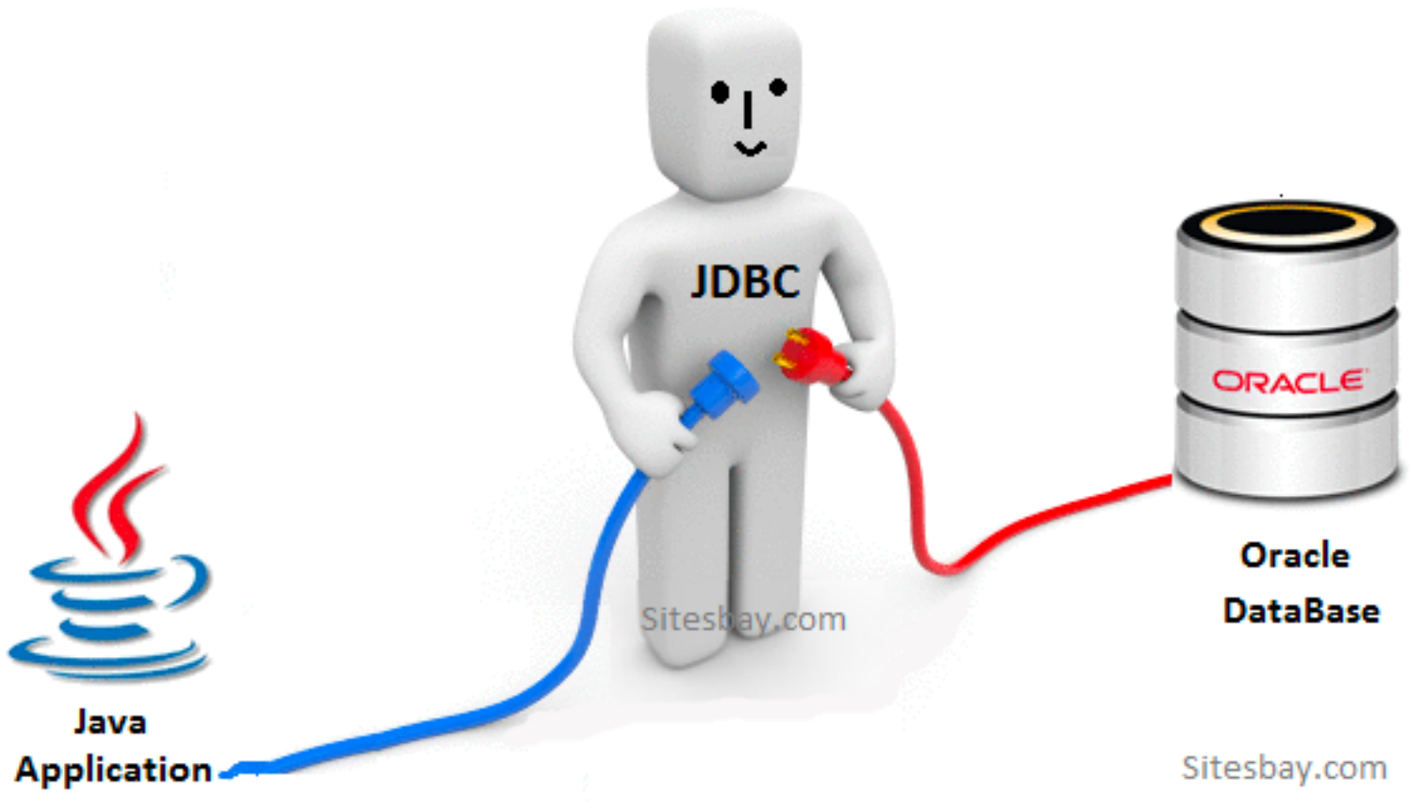
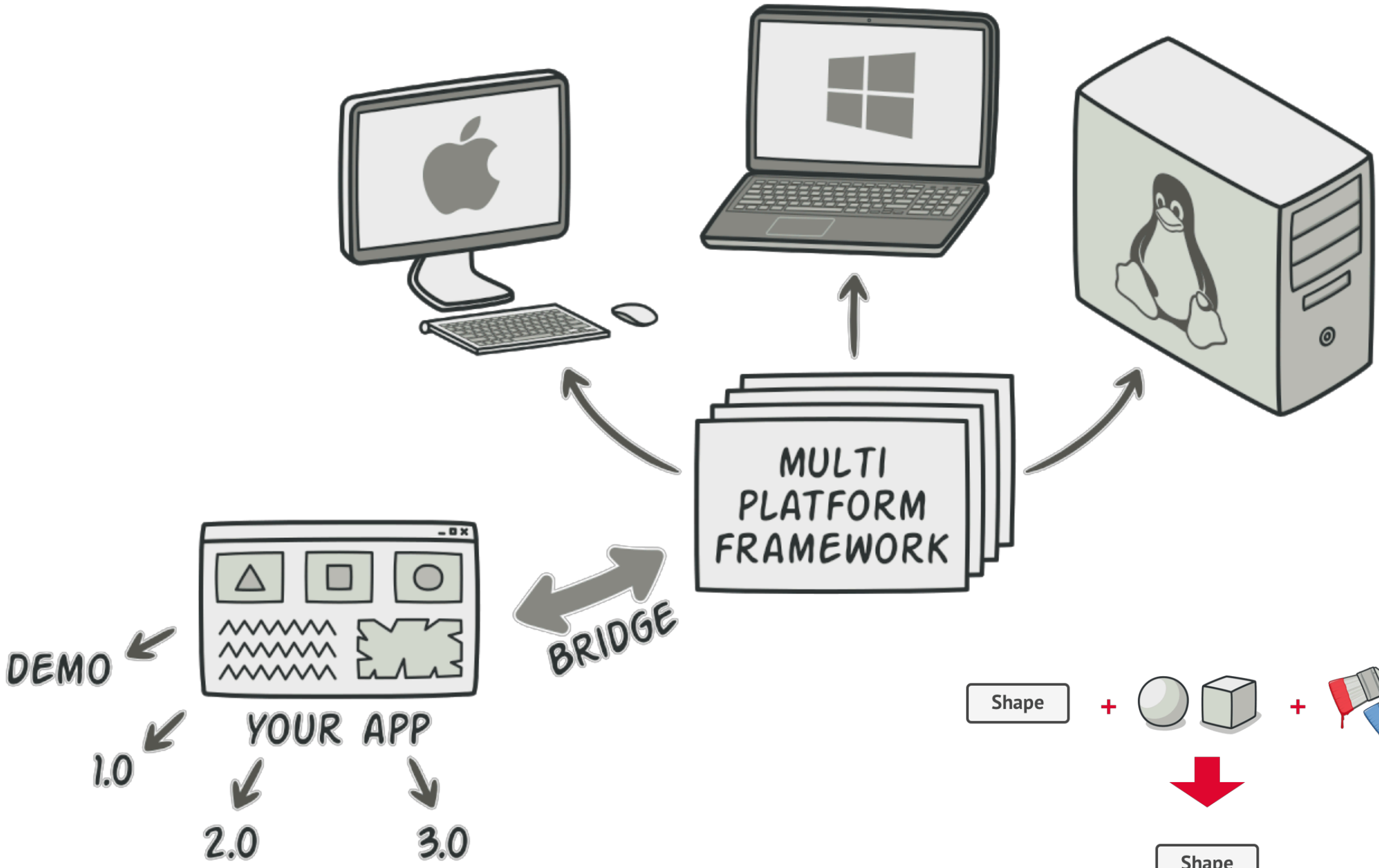
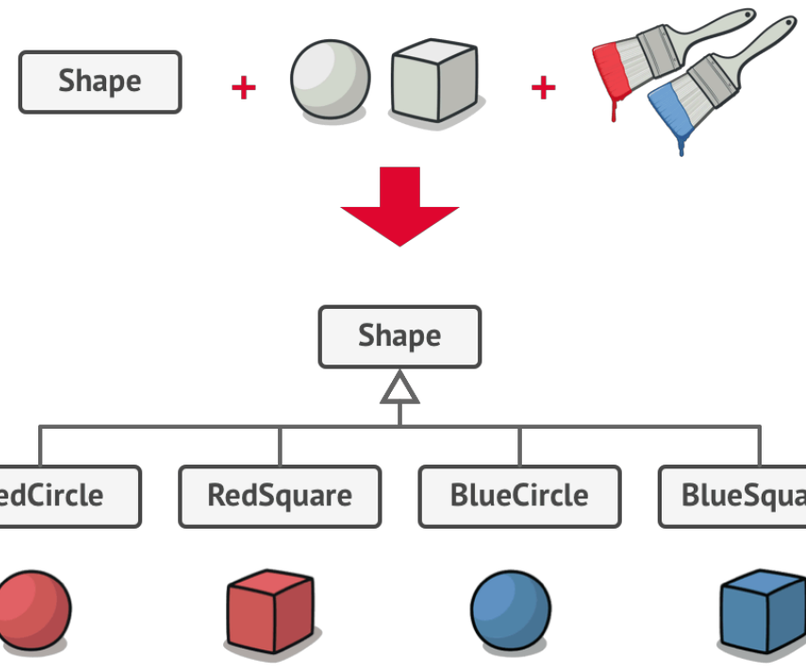
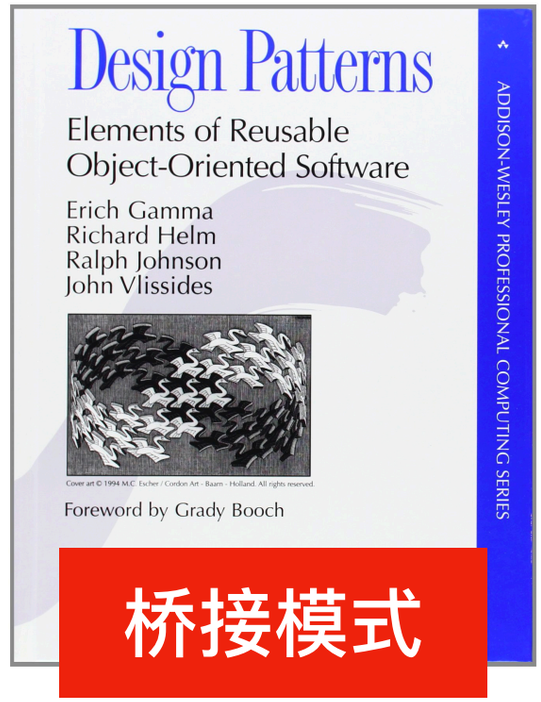
数据库访问

杨亮

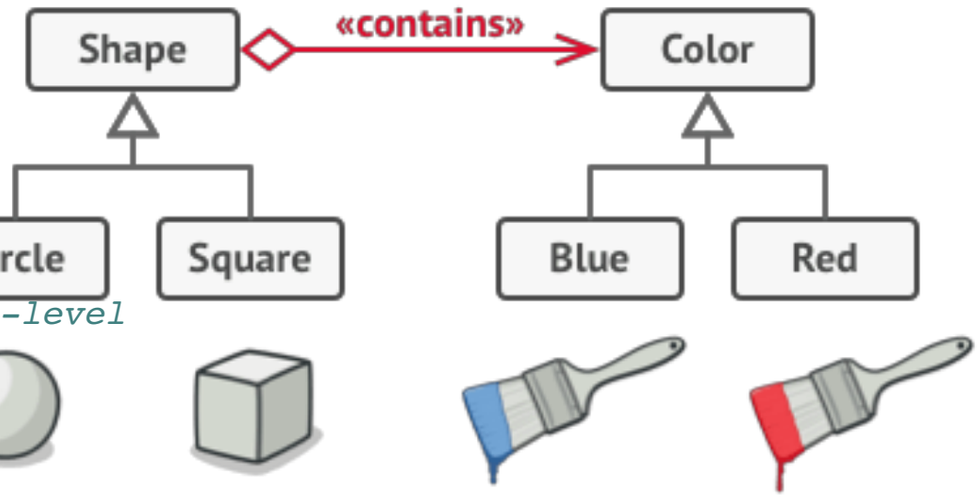

JDBC
ORACLE
Java
Application
Oracle
DataBase
Sitesbay.com
Sitesbay.com


MULTI
PLATFORM
FRAMEWORK
BRIDGE
DEMO
1.0
YOUR APP
2.0
3.0

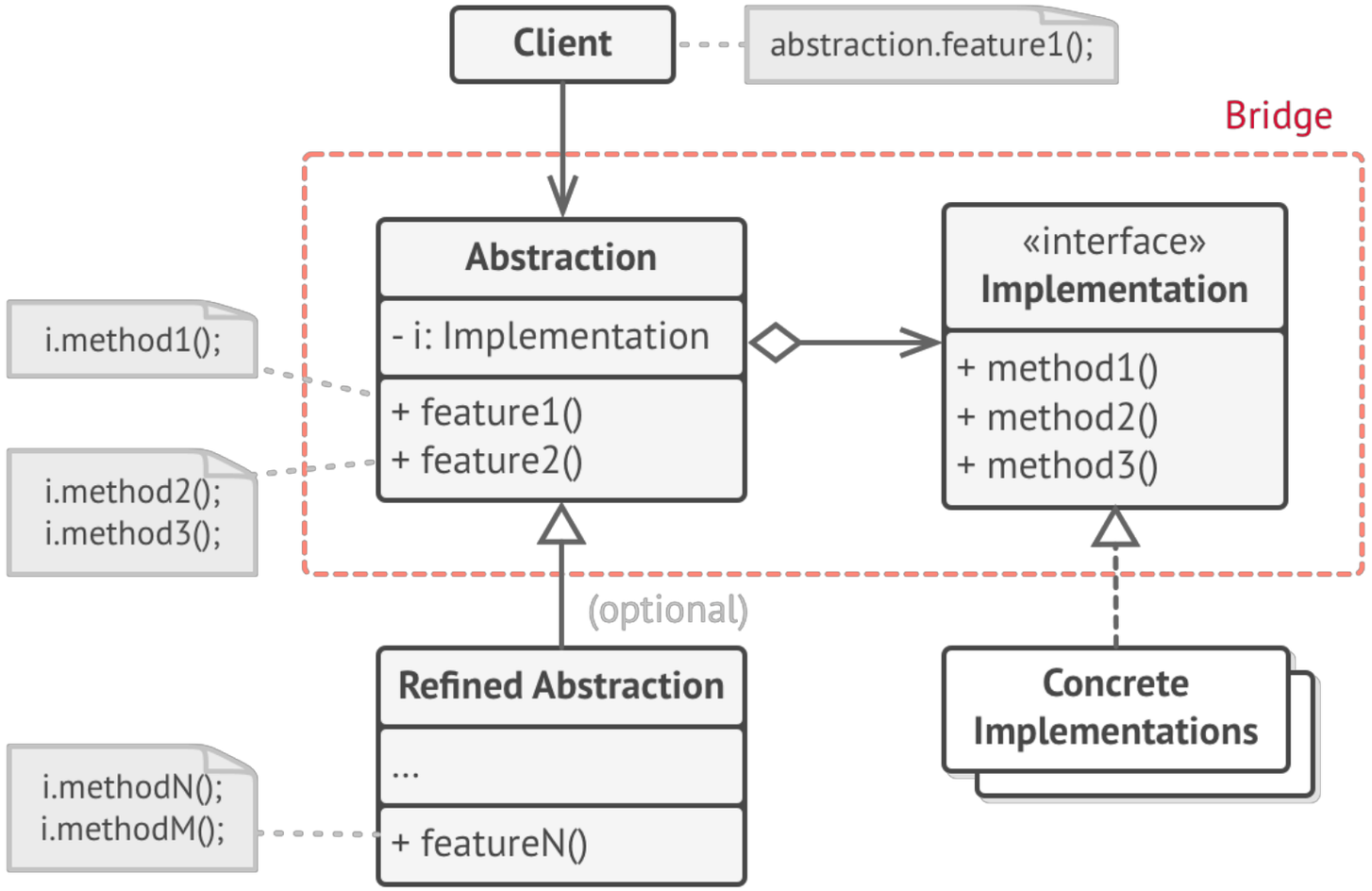

Shape
+
+
Shape
RedCircle
RedSquare
BlueCircle
BlueSquare

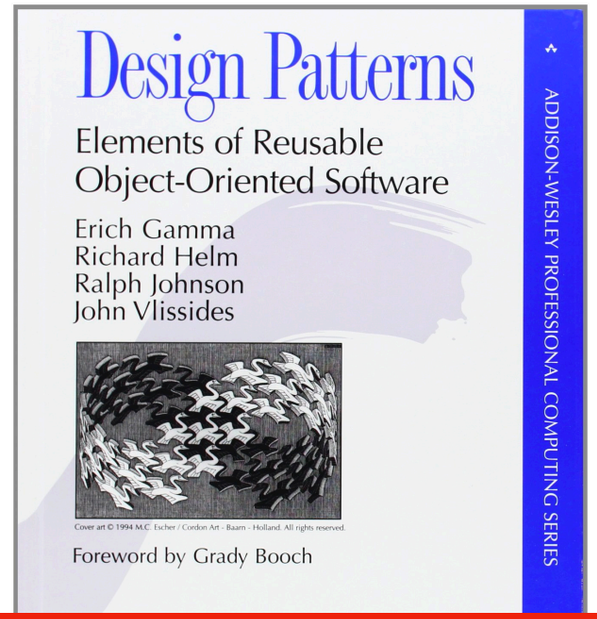



```java
/** "Implementor" */
interface DrawingAPI {
    public void drawCircle(final double x, final double y, final double radius);
}
/** "ConcreteImplementor"  1/2 */
class DrawingAPI1 implements DrawingAPI {
    public void drawCircle(final double x, final double y, final double radius) {
        System.out.printf("API1.circle at %f:%f radius %f%n", x, y, radius);
    }
}
/** "ConcreteImplementor" 2/2 */
class DrawingAPI2 implements DrawingAPI {
    public void drawCircle(final double x, final double y, final double radius) {
        System.out.printf("API2.circle at %f:%f radius %f%n", x, y, radius);
    }
}
/** "Abstraction" */
abstract class Shape {
    protected DrawingAPI drawingAPI;
    protected Shape(final DrawingAPI drawingAPI){
        this.drawingAPI = drawingAPI;
    }
    public abstract void draw();                                    // low-level
    public abstract void resizeByPercentage(final double pct);      // high-level
}
/** "Refined Abstraction" */
class CircleShape extends Shape {
    private double x, y, radius;
    public CircleShape(final double x, final double y, final double radius, final DrawingAPI drawingAPI) {
        super(drawingAPI);
        this.x = x;  this.y = y;  this.radius = radius;
    }
    // low-level i.e. Implementation specific
    public void draw() {
        drawingAPI.drawCircle(x, y, radius);
    }
    // high-level i.e. Abstraction specific
    public void resizeByPercentage(final double pct) {
        radius *= (1.0 + pct/100.0);
    }
}
```

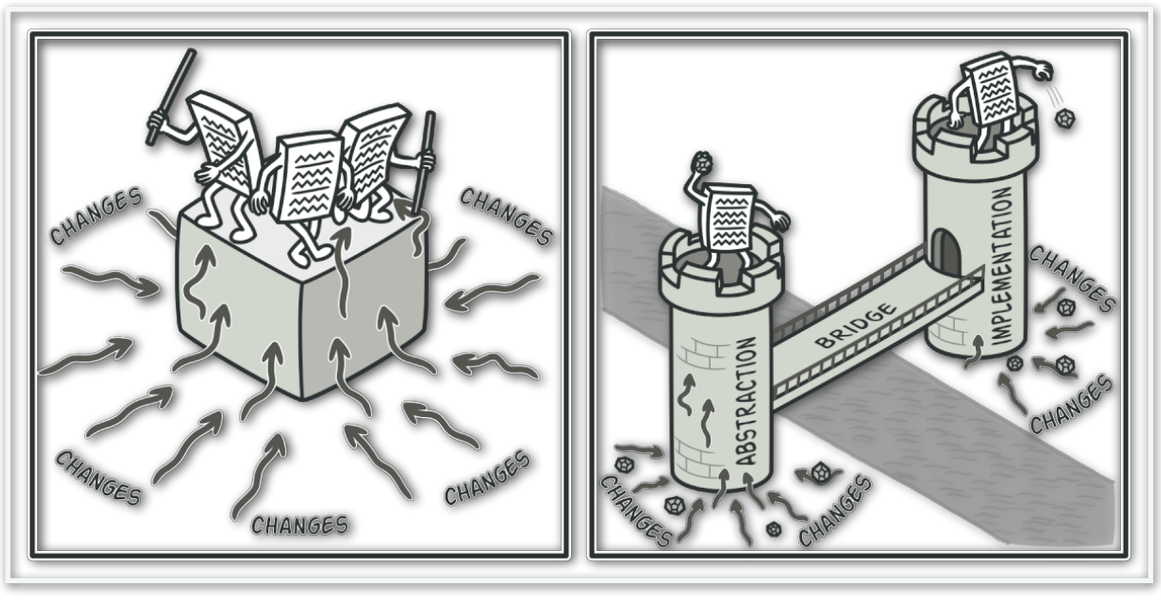


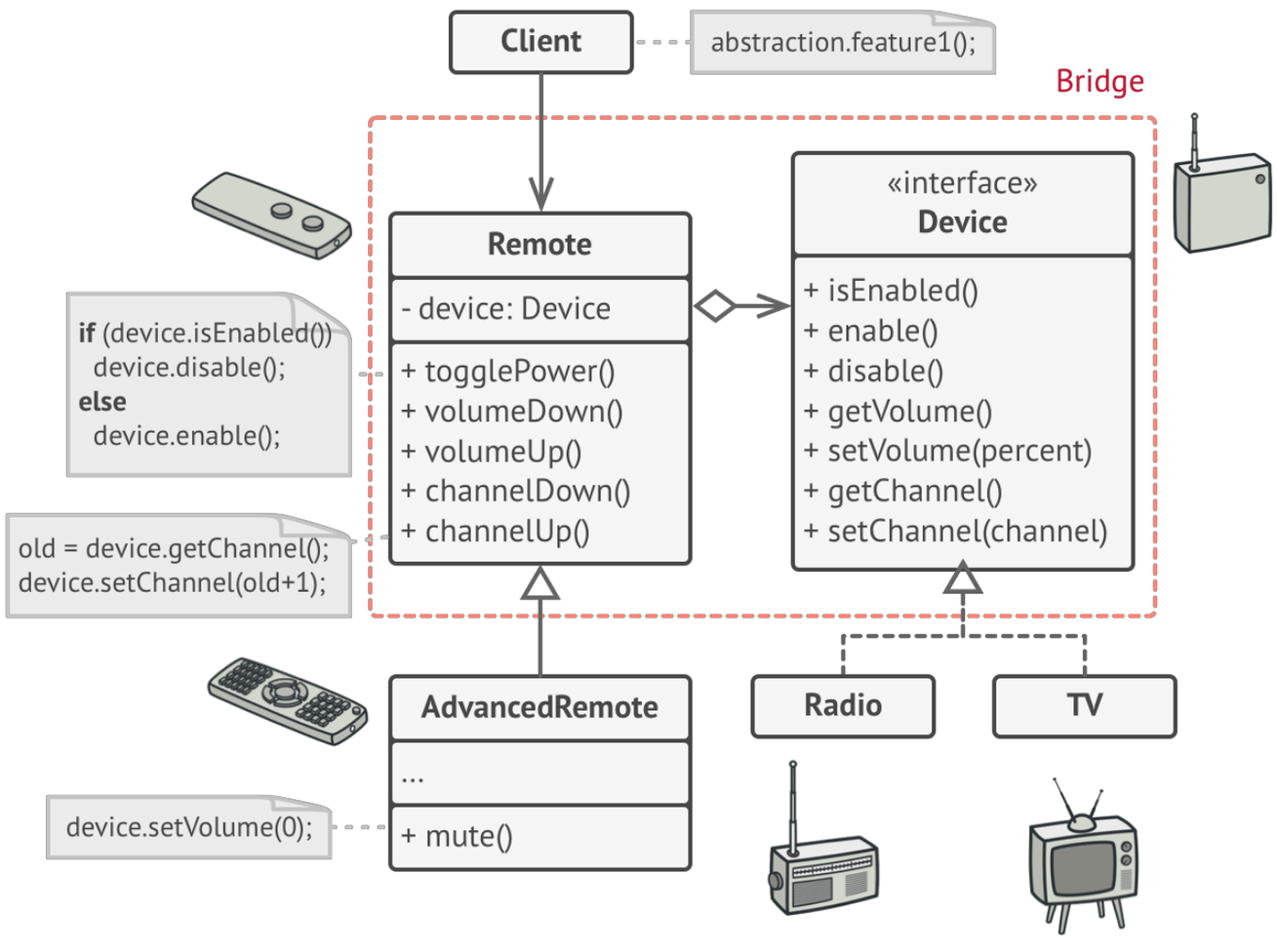

```java
class BridgePattern {
    public static void main(final String[] args) {
        Shape[] shapes = new Shape[] {
            new CircleShape(1, 2, 3, new DrawingAPI1()),
            new CircleShape(5, 7, 11, new DrawingAPI2())
        };
        for (Shape shape : shapes) {
            shape.resizeByPercentage(2.5);
            shape.draw();
        }
    }
}
```


Client
abstraction.feature1();
Bridge
Abstraction
- i: Implementation
+ feature1()
+ feature2()
«interface»
Implementation
+ method1()
+ method2()
+ method3()
i.method1();
i.method2();
i.method3();
(optional)
Refined Abstraction
...
+ featureN()
Concrete
Implementations
i.methodN();
i.methodM();

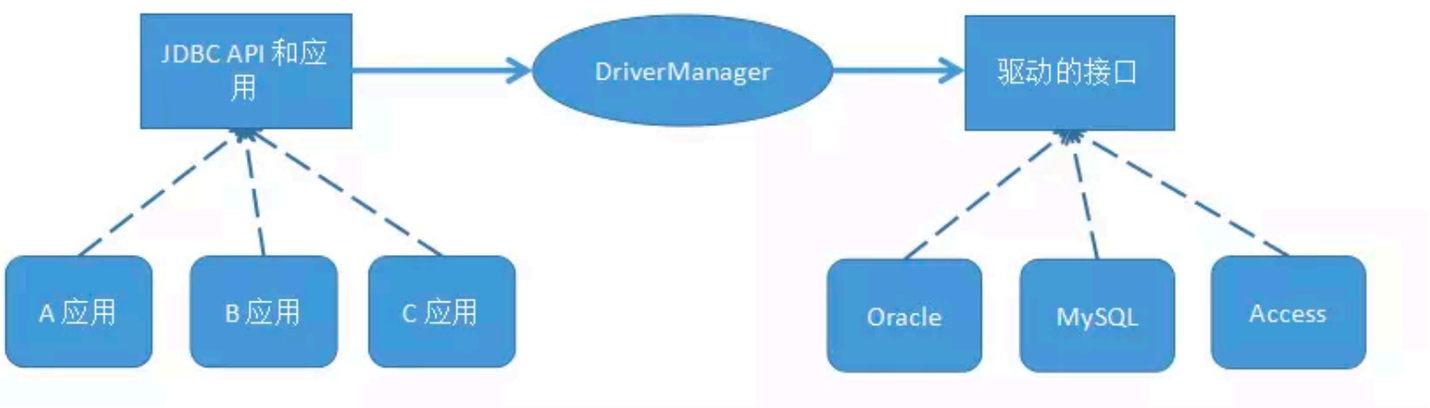

Design Patterns
Elements of Reusable
Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Foreword by Grady Booch
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

桥接模式

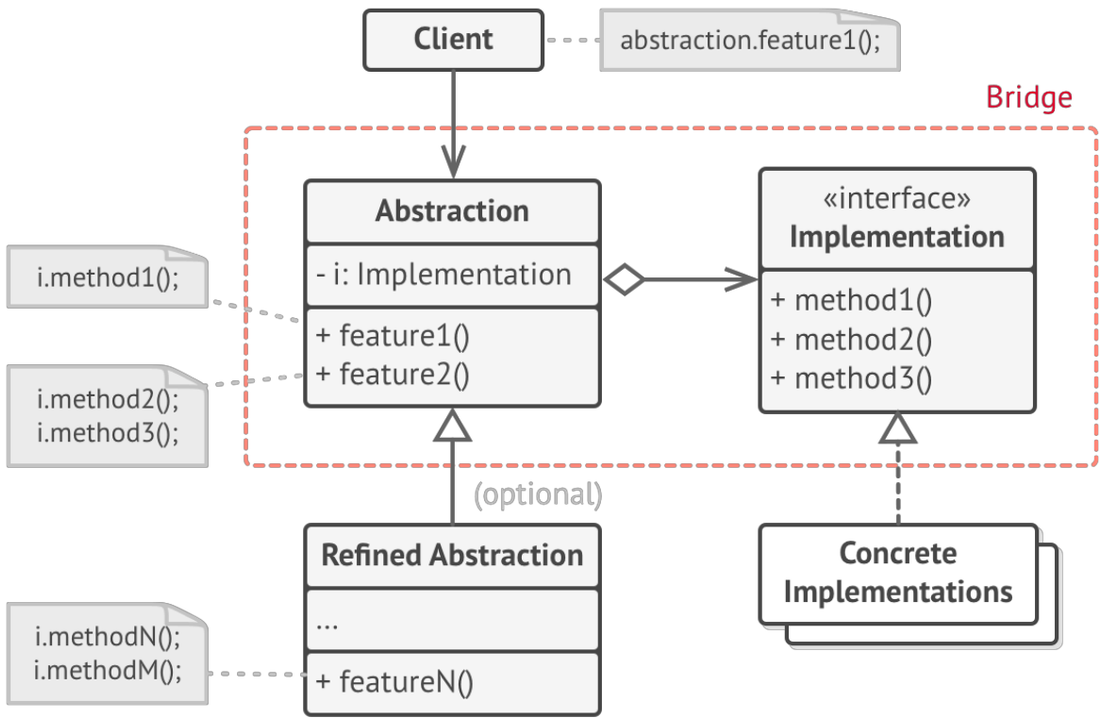

CHANGES
CHANGES
CHANGES
CHANGES
CHANGES
CHANGES
ABSTRACTION
BRIDGE
IMPLEMENTATION
CHANGES
CHANGES
CHANGES
CHANGES


Client
abstraction.feature1();
Bridge
Remote
- device: Device
+ togglePower()
+ volumeDown()
+ volumeUp()
+ channelDown()
+ channelUp()
«interface»
Device
+ isEnabled()
+ enable()
+ disable()
+ getVolume()
+ setVolume(percent)
+ getChannel()
+ setChannel(channel)
if (device.isEnabled())
device.disable();
else
device.enable();
old = device.getChannel();
device.setChannel(old+1);
AdvancedRemote
...
+ mute()
device.setVolume(0);
Radio
TV

```java
public class Client {
    public static void main(String[] args) {
        DriverManager manager = new DriverManager(new MysqlDriver());
        manager.getConnection();

        manager = new DriverManager(new OracleDriver());
        manager.getConnection();
    }
}

public abstract class Manager {
    private Driver driver;

    public  void getConnection(){
        driver.getConnection();
    }
    public void setDriver(Driver driver) {
        this.driver = driver;
    }
}

public class DriverManager extends Manager {
    public DriverManager(Driver driver){
        setDriver(driver);
    }
    public void getConnection() {
        super.getConnection();
    }
}
```



JDBC

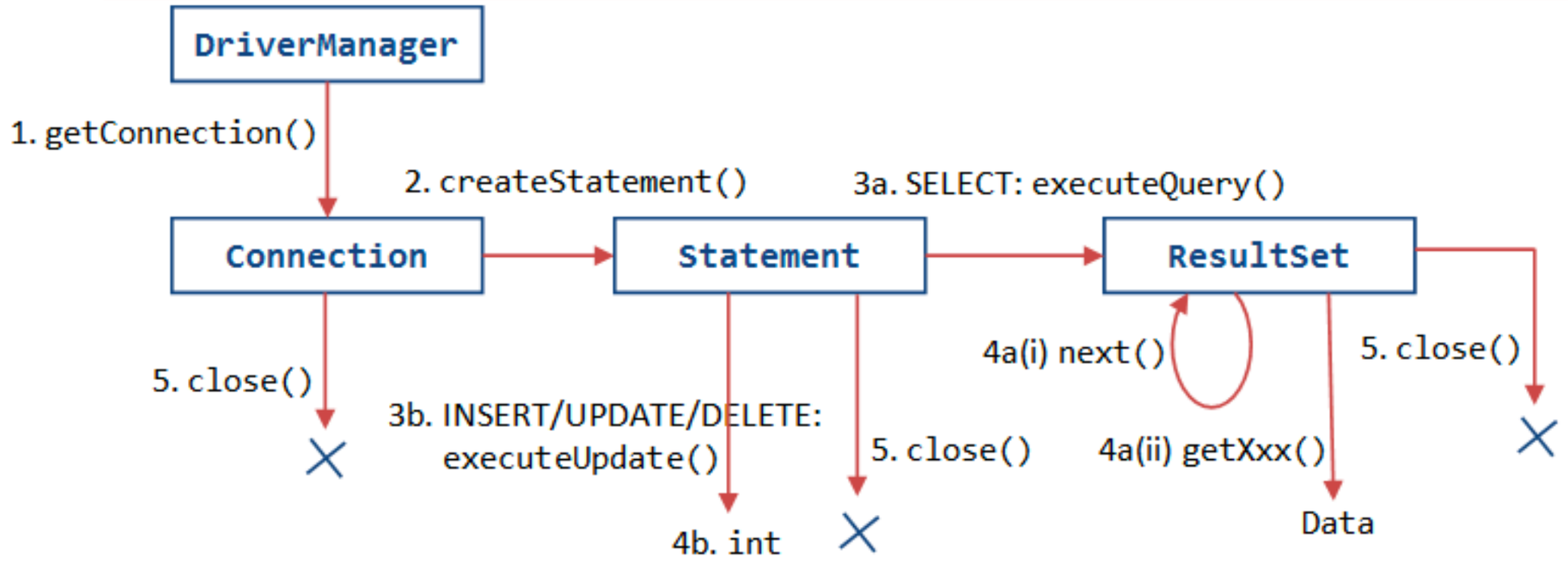



```java
public interface Driver {
    public void getConnection();

}

public class MysqlDriver implements Driver{
    @Override
    public void getConnection() {
        System.out.println("mysql 数据库连接");
    }
}

public class OracleDriver implements Driver {
    @Override
    public void getConnection() {
        System.out.println("oracle数据库连接");
    }
}
```

Java DataBase Connectivity

```java
public class Driver extends NonRegisteringDriver implements java.sql.Driver {
    //
    // 注册mysql驱动本身到驱动管理器
    //
    static {
        try {
            java.sql.DriverManager.registerDriver(new Driver());
        } catch (SQLException E) {
            throw new RuntimeException("Can't register driver!");
        }
    }

    public Driver() throws SQLException {
        // Required for Class.forName().newInstance()
    }
}
```

0. Load database driver `Class.forName("com.mysql.jdbc.Driver");`



```java
public class MySQLDemo {
    // JDBC 驱动名及数据库 URL
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
    static final String DB_URL = "jdbc:mysql://localhost:3306/RUNOOB";

    // 数据库的用户名与密码，需要根据自己的设置
    static final String USER = "root";
    static final String PASS = "123456";

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
        try{
            // 注册 JDBC 驱动
            Class.forName("com.mysql.jdbc.Driver");
            // 打开链接
            System.out.println("连接数据库...");
            conn = DriverManager.getConnection(DB_URL,USER,PASS);

            // 执行查询
            System.out.println(" 实例化Statement对象...");
            stmt = conn.createStatement();
            String sql;
            sql = "SELECT id, name, url FROM websites";
            ResultSet rs = stmt.executeQuery(sql);
            // 展开结果集数据库
            while(rs.next()){
                // 通过字段检索
                int id  = rs.getInt("id");
                String name = rs.getString("name");
                String url = rs.getString("url");

                // 输出数据
                System.out.print("ID: " + id);
                System.out.print(", 站点名称: " + name);
                System.out.print(", 站点 URL: " + url);
                System.out.print("\n");
            }
            // 完成后关闭
            rs.close();
            stmt.close();
            conn.close();
        }catch(SQLException se){
            // 处理 JDBC 错误
            se.printStackTrace();
        }catch(Exception e){
            // 处理 Class.forName 错误
            e.printStackTrace();
        }finally{
            // 关闭资源
            try{
                if(stmt!=null) stmt.close();
            }catch(SQLException se2){
            }// 什么都不做
            try{
                if(conn!=null) conn.close();
            }catch(SQLException se){
                se.printStackTrace();
            }
        }
        System.out.println("Goodbye!");
    }
}
```