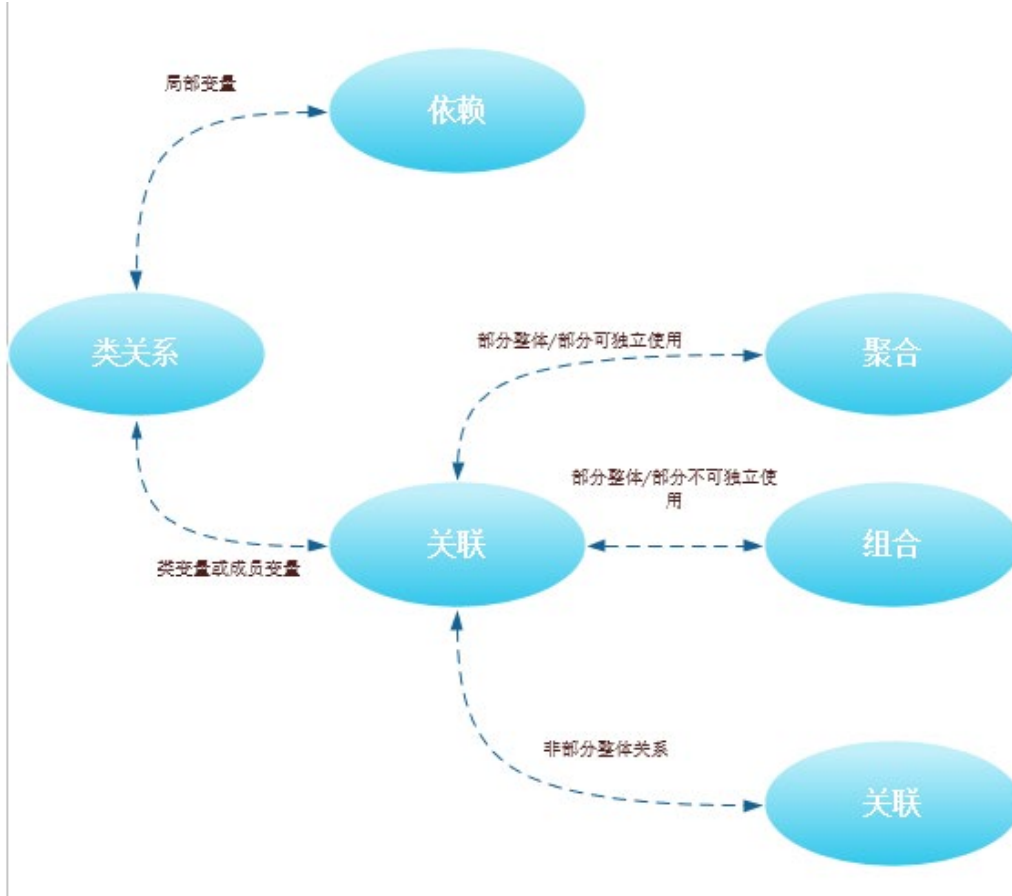


UML 时序图

一、类与类之间的四种关系（详情见 CSDN）。

1. 类与类之间的关系有：依赖、关联、组合、聚合、继承等



2. 所谓依赖关系：A 类中有某个或某几个方法用类 B 做参数或局部变量，我们就叫 A 依赖于 B，依赖关系最重要的特点是在 A 类加载时不会加载 B，只有在方法调用的时候才会加载 B 类，即 B 为局部变量

```

public class A{

    public void function1(B b){

    }

    public B function2(){

    }

}

public class B{

}

```

A依赖B

https://blog.csdn.net/weixin_41160066

- 3.
4. 依赖关系的例子有很多，比如：LocalDateTime 的 now 方法根据时区 ID 创建 LocalDateTime，这里说明 LocalDateTime 依赖 ZoneId。

```

public static LocalDateTime now(ZoneId zone) {
    return now(Clock.system(zone));
}

```

5. 关联关系：特点是：B 作为 A 的类变量或成员变量存在，而不是局部变量存在，我们称 A 关联 B，当然可以循环依赖

```

// @Autowired
public class UserController {

    //static final Logger LOGGER = LogManager.g
    static final Logger LOGGER = LoggerFactory.
    @Autowired
    private UserService userService;
}

```

关联关系

6. 聚合关系：在关联关系的基础上，比如 B、C 为 A 的类变量或成员变量，并且组成了部分和整体的关系，且部分可以独立于整体单独使用，称之为聚合关系

```

public static void main(String[] args) {
    Student student = new Student();
    Teacher teacher = new Teacher();
}

```

并且他们可以单独使用

```

3 import lombok.Data;
4
5 @Data
6 public class Grade {
7     //老师、学生和班级是聚合关系
8     private Student student;
9
10    private Teacher teacher;
11
12 }
13

```

老师和学生组成了班级

1. 组合关系：在关联关系的基础上，比如 B、C 为 A 的类变量或成员变量，并且组成了部分和整体的关系，且部分不可以独立于整体单独使用，称之为组合关系
2. 比如对于攀岩运动来讲，四肢和躯干组成的身体来完成攀岩，单独的一部分对攀岩运动来讲是没有意义的，

```

public static void panyan(){
    Leg leg = new Leg();
    Arm arm = new Arm();
}

```

单独的手臂和腿相对于攀岩是无意义

3. 组合和聚合相对来讲，聚合很常见。
4. 继承：继承有很明显的标志 extends

```

@Data
public class Grade extends String{
|
}

```