@Autowired 단일 타입 주입

```java
package com.goorm.springmissionsplayground.mission02_spring_core_basic.task04_auto_injection.service;

import com.goorm.springmissionsplayground.mission02_spring_core_basic.task04_auto_injection.dto.InjectionCaseResult;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class AutowiredOnlyService {

    private final AmountFormatter amountFormatter;

    @Autowired
    public AutowiredOnlyService(AmountFormatter amountFormatter) {
        this.amountFormatter = amountFormatter;
    }

    public InjectionCaseResult compare(int amount) {
        return new InjectionCaseResult(
                "@Autowired",
                "amountFormatter",
                "포맷 결과: " + amountFormatter.format(amount),
                "동일 타입 빈이 1개라서 타입 기반 자동 주입"
        );
    }
}
```

@Qualifier 명시 주입

```java
package com.goorm.springmissionsplayground.mission02_spring_core_basic.task04_auto_injection.service;

import com.goorm.springmissionsplayground.mission02_spring_core_basic.task04_auto_injection.dto.InjectionCaseResult;
import com.goorm.springmissionsplayground.mission02_spring_core_basic.task04_auto_injection.policy.DiscountPolicy;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.stereotype.Service;

@Service
public class QualifierInjectionService {

    private final DiscountPolicy discountPolicy;
    private final AmountFormatter amountFormatter;

    @Autowired
    public QualifierInjectionService(
            @Qualifier("fixedDiscountPolicy") DiscountPolicy discountPolicy,
            AmountFormatter amountFormatter
    ) {
        this.discountPolicy = discountPolicy;
        this.amountFormatter = amountFormatter;
    }

    public InjectionCaseResult compare(int amount) {
        int discounted = discountPolicy.discount(amount);
        return new InjectionCaseResult(
                "@Autowired + @Qualifier",
                discountPolicy.beanName(),
                "할인 금액: " + amountFormatter.format(discounted),
                "동일 타입 빈이 여러 개일 때 이름으로 주입 대상을 지정"
        );
    }
}
```

@Primary 우선 주입

```java
package com.goorm.springmissionsplayground.mission02_spring_core_basic.task04_auto_injection.policy;

import org.springframework.context.annotation.Primary;
import org.springframework.stereotype.Component;

@Primary
@Component("rateDiscountPolicy")
public class RateDiscountPolicy implements DiscountPolicy {

    private static final int DISCOUNT_RATE = 10;

    @Override
    public int discount(int amount) {
        return amount * DISCOUNT_RATE / 100;
    }

    @Override
    public String beanName() {
        return "rateDiscountPolicy";
    }
}
```

의존관계 자동 주입 방식 비교표

```
주입 방식 비교표

| 방식 | 적용 위치 | 동일 타입 빈 다수일 때 동작 | 선택된 빈(이번 실습) | 장점 |
| --- | --- | --- | --- | --- |
| @Autowired | 생성자 파라미터 | 타입 기준 자동 선택, 후보 2개 이상이면 충돌 | amountFormatter (단일 타입) | 기본 전략으로 가장 간단 |
| @Autowired + @Qualifier | 생성자 파라미터 + 한정자 | 지정한 빈 이름으로 정확히 선택 | fixedDiscountPolicy | 의도한 구현체를 명시적으로 선택 |
| @Autowired + @Primary | 구현체 클래스(@Primary) + 생성자 | @Primary 빈을 기본 후보로 우선 선택 | rateDiscountPolicy | 기본 구현체를 전역 기본값으로 지정 |
```