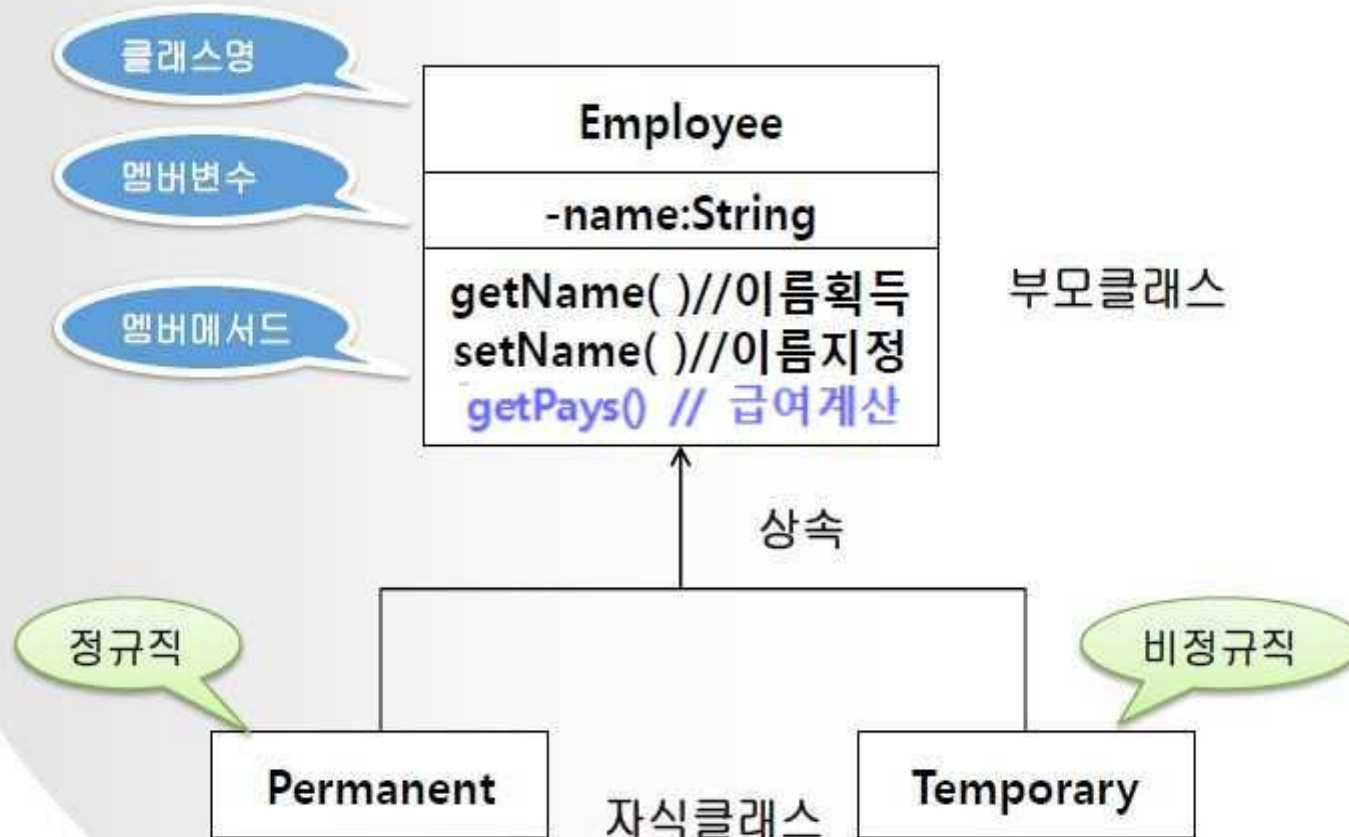




급여관리 시스템

- 클래스 다이어그램





Employee 클래스 설계

- 직원의 고용형태
 - 정규직, 임시직
 - 급여 계산 방식의 차이
 - 정규직(Permanent) : 기본급 + 보너스
 - 임시직(Temporary) : 작업시간 * 시간당 급여
- Employee클래스
 - 정규직과 임시직이 공통적으로 지니고 있어야 하는 멤버들을 모아서 Employee 클래스를 만든다.



정규직 클래스 설계

- 정규직 클래스 (Permanent) – Employee클래스를 상속받음
 - 직원의 이름과 급여정보를 저장하기 위한 클래스
 - (단, 정규직 급여는 입사 당시 정해졌다고 가정하고 급여인상은 없음)
 - 멤버변수 : 기본급(pay), 보너스(bonus)
 - 생성자 : 기본급, 보너스 초기화
 - 상속받은 Setter/Getter : 이름 지정 및 획득
 - 상속받은 메서드 재정의 : 급여계산(기본급 + 보너스)



임시적 클래스 설계

- 임시적 클래스 정의(Temporary) -Employee클래스를 상속받음
 - 멤버변수 : 작업시간(time), 시간당 급여/pay)
 - 생성자 : 작업시간, 시간당 급여 초기화
 - 상속받은 Setter/Getter : 이름 지정 및 획득
 - 상속받은 메서드 재정의 : 급여계산(time*pay)



실행 화면

<<정규직 실행화면 >>

고용형태-정규직<P>, 임시직<T>를 입력하세요.

P

이름, 기본급, 보너스를 입력하세요.

홍길동

2000000

50000

=====

고용형태 : 정규직

이름 : 홍길동

급여 : **2,500,000원**

<<임시직 실행화면>>

고용형태-정규직<P>, 임시직<T>를 입력하세요.

T

이름, 작업시간, 시간당 급여를 입력하세요.

유관순

45

5000

=====

고용형태 : 임시직

이름 : 유관순

급여 : **225,000원**



급여관리 클래스 정리

● 급여관리 시스템 관련 클래스 구성

- class Employee
 - 멤버변수 : 이름(name)
 - 생성자, getter/setter
 - 메서드 : 급여계산 getPays(){ return 0; }
- class Permanent - 정규직
 - 멤버변수 : 기본급여(pay), 보너스(bonus)
 - 생성자, getter/setter
 - 메서드 : 급여계산 getPays() 재정의 => pay + bonus
- Class Temporary - 임시직
 - 멤버변수 : 시간당 급여(pay), 작업시간(time)
 - 생성자, getter/setter
 - 메서드 : 급여계산 getPays() 재정의 => pay * time
- main()
 - 결과 화면 출력 시 getter 이용하여 출력