

Day10

Day9 5/25 리뷰&실습

클래스 선언과 사용

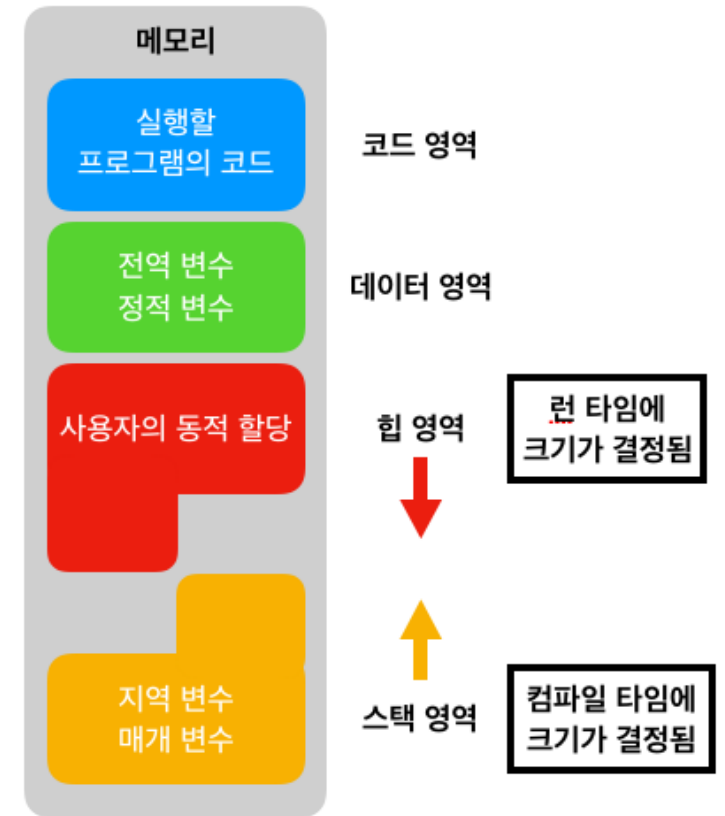
- 자바 클래스 생성
- Main class 와 일반 class
 - Main.java, Student.java
- 패키지가 다른 경우
 - import
 - full package name
 - Chapter5Main.java

클래스 메서드

- 1. 클래스 안에서만 존재.
- 2. 반환형식이 있다.
- 3. 반환이 없을 수 있다: void
- 4. 매개변수를 사용.
- return 키워드
 - 반환
 - Return void
- Main class 의 static main() 메서드와 일반 메서드 차이

객체지향 프로그래밍

- 클래스 작성(설계)
 - 현실 세계의 요구사항 -> 객체로 분석
 - 객체: 속성, 기능(메서드)
 - 객체 -> 코드화
 - 클래스로 정의
 - 멤버 필드
 - 멤버 메서드
 - 클래스를 인스턴스로 생성해 사용
 - new 연산자
 - 식별자 -> 객체
- 참조형
 - 기본형 제외한 나머지 (스트링 리터럴, 배열, 클래스 객체)



<https://jinshine.github.io/2018/05/17컴퓨터%20기초/메모리구조/>

생성자

- 특별한 메소드: () 연산자, 매개변수 사용
 - class 이름과 같다.
 - 매개변수 갖는다
- return 타입 없다
- 생성자 오버로딩
 - 메서드 오버로딩

실습: 소수 계산

- 연습) 숫자 n 값을 입력 받아 1부터 n 까지 소수를 구하는 프로그램을 작성
 - 1에서 n 까지 소수 출력.

1보다 큰 자연수 중 1과 자기 자신만을 약수로 가지는 수다.

예) 5는 1×5 또는 5×1 로 수를 곱한 결과를 적는 유일한 방법이 그 수 자신을 포함하기 때문에 5는 소수이다.

그러나 6은 자신보다 작은 두 숫자(2×3)의 곱이므로 소수가 아니다

1과 그 수 자신 이외의 자연수로는 나눌 수 없는 자연수로 정의하기도 한다.

실습: 소수 계산

- 소수 를 계산하는 클래스를 선언하고 대상 수까지의 소수를 계산해 보자.
 - 생성자 입력된 t 값까지 소수 계산

Prime
Prime() Prime(int t)
number: int // 타겟 수 t primeNumbers[:]: int
print() primeNumber() setNumber(t)

T까지 소수 배열

0 부터 t 까지 소수는 [1, 2, 3, 5, ...]
이다.

T 까지 소수를 찾는...

Day10 과정

Day10

- 리뷰 실습: Prime number class
- This p170
- 유효범위 p191
- 정보는닉 p162
 - 접근 제한자 (한정자, Modifier)
 - <https://wikidocs.net/232>
- 상속
 - <https://opentutorials.org/course/1223/6060>
 - <https://wikidocs.net/280>
- 객체지향 언어 특성
 - 추상화 <https://codevang.tistory.com/78>
 - 캡슐화 <https://codevang.tistory.com/79>
 - 상속 <https://codevang.tistory.com/80>
 - 다형성 <https://codevang.tistory.com/81>

실습: 클래스 생성자, this, this() 이용

- 사람을 표현하는 Person 클래스
 - this, this() 생성자 사용.

Person
Person() // name=이름없음 Person(name, age) Person(name, age, gender, blood, job)
name: String job: String age: int gender: String blood: String
play();



의사: 진료한다
골퍼: 라운딩 갔다.
교수: 강의한다.

Constructor, this, this() 정리

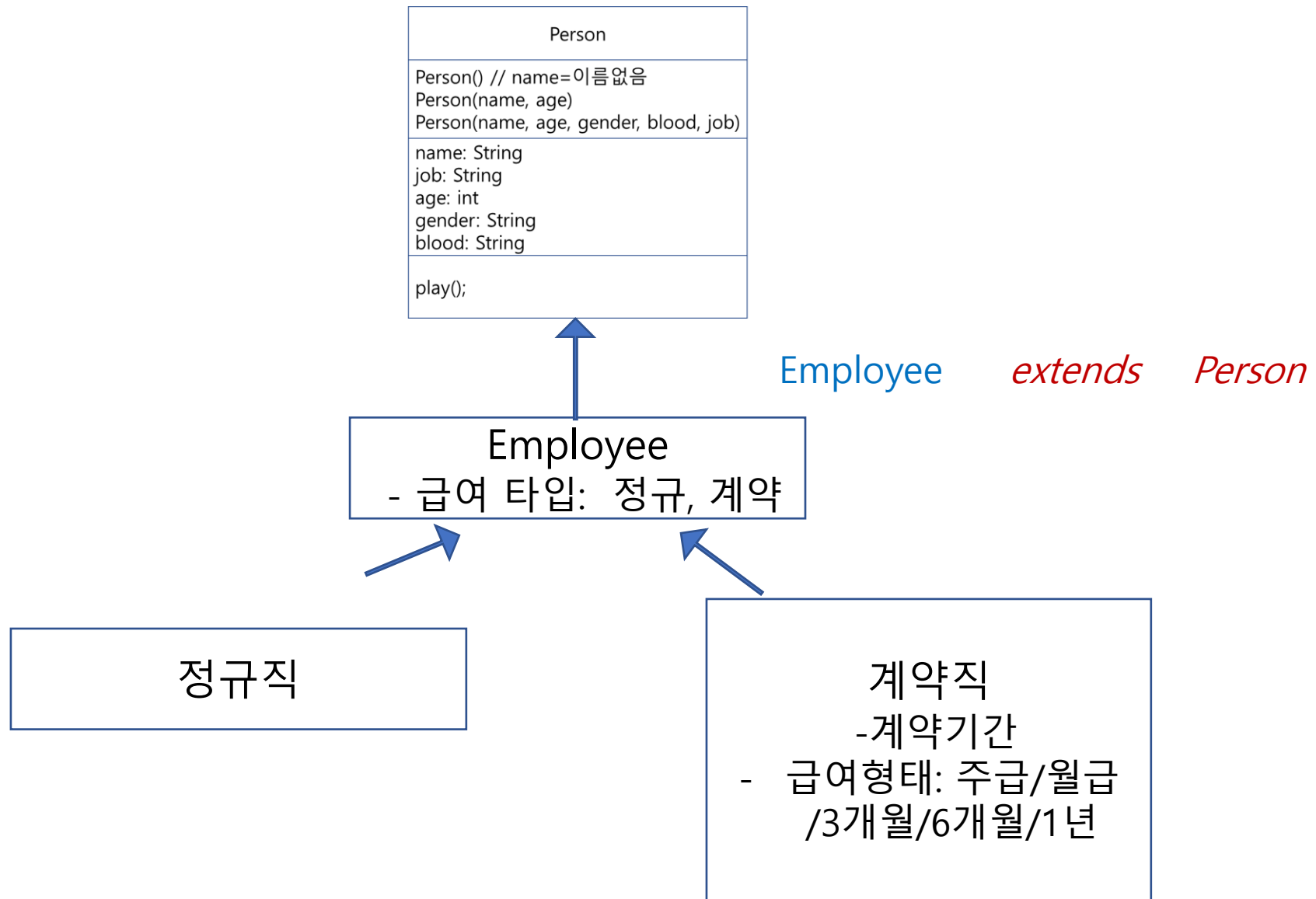
- 생성자의 특징
 - 생성자는 **메소드**
 - 생성자 이름은 클래스 이름과 반드시 동일
 - **생성자 오버로딩 == 메서드 오버로딩**
 - **생성자는 리턴 타입을 지정할 수 없음**
 - 생성자의 목적은 객체 초기화
 - 생성자는 객체가 생성될 때 반드시 호출됨.
 - 생성자 없으면 컴파일러가 자동으로 기본 생성자 삽입
 - 생성자는 new를 통해 객체를 생성할 때, 객체당 한 번 호출
- this의 필요성
 - 객체의 멤버 변수와 메소드 변수의 이름이 같은 경우
 - 다른 메소드 호출 시 객체 자신의 레퍼런스를 전달할 때
 - 메소드가 객체 자신의 레퍼런스를 반환할 때
- this()
 - 클래스 내의 다른 생성자 호출
 - 생성자 내에서만 사용 가능
 - 반드시 생성자 코드의 제일 처음에 수행

실습: 접근제한자

- Main 클래스에서 Bus 클래스의 인스턴스를 만들고 name, number를 출력하도록 수정 혹은 사용해 보자.

```
3 public class Bus {  
4     private String name;  
5     private int number;  
6  
7     public Bus(String name, int number) {  
8         this.name = name;  
9         this.number = number;  
10    }  
11 }
```

Bus 필드 name, number 접근을 위해
1. 직접 사용하게 하거나
2. 메서드를 사용해 보세요.



실습: 자동차

- 자동차는 run() 으로 달린다
- 구급차는 사이렌을 울리고
 - 응급키트가 배치되어 있다.

자동차: Car
- 이름
- 배기량: cc
- run()

구급차: Ambulance
- 응급키트: kit
- siren()

자동차 이름, 배기량 - 매개변수

실습: 상속을 통한 클래스 구현

- 자동차
 - 달린다
- 경찰차
 - 사이렌
- 덤프트럭
 - 짐칸 내린다
 - 짐칸 올린다.

실습: 상속을 통한 클래스 구현

- Calculator 를 상속한 SubCalculator 를 구현
 - sub() 는 op1, op2를 뺀다.

