

2차강 Class의 기본개념

7/13

1. 절차 지향 프로그래밍

- (1) 프로그램을 개발할 때 프로그램이 동작하는 순대로 프로그래밍을 하는 기법
- (2) 프로그램이 수행되고 종료될 때까지 모든 기능이 직렬순서대로 기술되어진다.
- (3) 유지보수 ↓, 프로그램을 각 기능별로 구분하여 작성하기 힘들.
- (4) 코드 재사용 ↓

(5) OOP (Object Oriented Programming) 객체지향 프로그래밍

- 프로그램을 작성할 때, 먼저 프로그램에서 사용할 기능들을 정하고 정해진 기능을 가진 작은 부품들을 만들어서, 이렇게 만든 부품들을 사용하여 프로그램을 작성하는 기법.

- ① 프로그램에서 사용되는 작은 부품들 = 객체
- ② 객체를 만들 수 있는 것 = 클래스
- ③ 개발자 → 클래스 → 객체

프로그램 코드

- ④ 유지보수 ↑

2. 클래스 설계 방법

- (1) 나에게 필요한 기능이 무엇인가?
- (2) 필요한 기능들을 구체별로 구분할 수 있는가?
- (3) 클래스 간의 관계성을 최한으로 하였는가?
- (4) 이미 존재하는 클래스와 공유되는 부분이 있는가?
- (5) 사용하기 쉽게 만들었는가?
- (6) 꼭 필요한 기능만을 가진 클래스인가?

Page

1. 플러그 설치방법

(1) 클러스터에 대한 설명

① $\frac{1}{2}$ 이 $\frac{1}{2}$ 이상

→ class 클래스 이름

② { } 와 { } 사이에 클래스의 내용은 라벨

→ class는 멤버변과 멤버함수의 묶음을 가진다.

(2) 과제의 생성 및 검토 classmate 생성된 경우

① new 클래스 이름 ();

② new를 통해 객체 생성 → 객체의 메모리의 주소를 넘겨줌.

→ 이 주파수를 변수에 지정했다가, 이후에 플롯할 때 주파수를 이용하여
표.

→ 이 수평선 상에 $\vec{v}_1 = \vec{v}_2$

③ 참조변수를 통해, 멤버 접근시,

"참고문헌 없음" 라고 기록하여 생략

ex) class → public class class 1 {
// 클래스
}

```
int va;  
// initialize  
public void setVa(int -va) {  
    va = -va;  
}  
  
public int getVa() {  
    return va;  
}  
}
```

class 이용 → // 객체 생성
class1 c1 = new class1()
// 멤버의 접근


```

cl. va = 100;
System.out.println(cl.getVa());
cl.setVa(1000);
System.out.println(cl.getVa());

```

29강 Class Member Method

1. 클래스의 멤버 메서드

(1) 멤버 메서드

- ① 클래스 내에서 개발자가 만든 코드의 단위
- ① 클래스가 가지고 있는 기능들을 정의할 때 사용
- ② 프로그래밍을 할 때 사용되는 코드는 메서드 내부에 작성해야함
- ③ 메서드는 클래스에서의 작업의 단위.

(2) 메서드 작성방법

→ [접근제한자] [리턴타입] 메서드 이름 ([매개변수]) {
 코드
 }

① 접근제한자

- ① 같은 클래스 내부가 아닌 외부에서 접근하고자 할 때 제한을 두는 부분.

② 리턴타입

- ① 메서드 코드를 실행한 후, 메서드를 사용한 부분이 값을 전달할 때, 전달되는 값의 타입을 적어준다.
 (전달할 값이 無 → void 작성)

③ 매개변수

- ① 메서드를 사용하는 부분에서 이 메서드에 값을 전달해주고자 할 때 매개변수를 통해 값을 사용할 수 있다.
 메서드를 사용하는 곳에서 값을 전달해주면 매개변수에 값이 설정되고 메서드 내부에서는 매개변수를 통해 받은 값을 사용할 수 있다.

2. 메서드의 overloading

(1) Overloading

- ① 클래스 내부의 모든 메서드들은 그 이름이 모두 같아야 한다.
- ② 객체지향프로그래밍 언어에서는 개발자가 메서드 사용을 좀 더 편하게 할 수 있도록 여러개의 메서드를 만들어 사용하는 것을 지원한다.
- ③ 원래, 메서드라 메서드의 구분은 메서드의 이름을 가지고 하지만, 메서드의 이름이 같은 경우 내부의 매개변수를 가지고 구분하게 된다.
- ④ 이러한 개념을 메서드의 overloading 이라 한다.

30강 클래스의 멤버변수

1. 클래스의 멤버변수

(1) 멤버변수

- ① class에서 선언되어 같은 class 내부라면 자유롭게 접근하여 사용할 수 있는 변수
- ② 상수인 이름으로 선언할 수 없다.
- ③ 다른 클래스에서 접근하기 위해서는 ". " 연산자를 이용한다.

(2) 지역변수

- ① 메서드 내부에서 선언된 변수. 선언된 메서드 내부에서만 사용가능.
- ② 클래스의 멤버변수라 지역변수의 이름이 동일
→ 지역변수가 우선, 멤버변수 접근을 위해선 this를 이용.