



# 상속

- super class : Car
  - String carName;
  - int start()
  - int changeSpeed(int speed)
  - int stop()
  - int changeGear(int num)
- sub class : SportsCar
  - int openCover()
  - int startBooster(int maxSpeed)
- Main
  - Car, SportsCar 객체 생성
    - start, changeSpeed 호출
  - SportsCar
    - openCover

## 상속

- super class : calculator
  - int sum(int x, int y)
  - int sub(int x, int y)
  - int mul(int x, int y)
  - int div(int x, int y)
- sub class : engineerCalculator
  - String showBinaryString(int dec)
  - String showHexString(int dec)

```
String hexString(int n )  
{  
    return Integer.toHexString(n);  
}  
String binaryString(int n )  
{  
    return Integer.toBinaryString(n);  
}
```

# 메소드 오버라이딩

- PrintArray
  - int arrayInt[]; -> 10개
    - for문을 이용 1-10까지 값 저장
  - print()
    - 인덱스 0부터 9까지 출력
- PrintReverseArray
  - print() : 메소드 오버라이딩
    - 인덱스 9부터 0까지 출력
- MainClass
  - 업캐스팅을 통해 메소드 오버라이딩된 print()호출

# 메소드 오버라이딩

- Animal
  - String name;
  - int age
  - int sound();
- Chicken
  - name 을 "park Geunhye"로 설정
  - int sound(); : 메소드 오버라이딩
    - "꽹꽹" 출력
- Mouse
  - name 을 "Lee Myeongbak"로 설정
  - int sound(); : 메소드 오버라이딩
    - "찍찍" 출력
- MainClass
  - chicken, mouse 객체의 name 출력
  - 업캐스팅을 통해 메소드 오버라이딩된 sound() 호출

# 메소드오버라이딩

- Printer
  - String company="HQ";
  - int modelNumber=45673;
  - int getModelNum(); -> 모델번호 리턴
  - boolean print(String content)
- InkJetPrinter
  - print() 메소드 오버라이딩
- LaserPrinter
  - print() 메소드 오버라이딩
- MainClass
  - 객체의 모델 번호 출력 및 업캐스팅을 통한 오버라이딩된 메소드 호출

# 메소드 오버라이딩

- Car
  - String company="Hia"
  - int carDriveType =2; -> 2륜구동
  - int checkEngine();
  - boolean startMusic(String musicName);
  - boolean stopMusic();
  - int run()
  - int stop()
- Car4WD
  - carDriveType 값을 4로 설정
    - 생성자의 매개변수를 통해 전달된 값으로 설정
  - run, stop() 를 메소드 오버라이딩
    - "4WD run", "4WD stop" 문자열만 출력되게
- CarApp
  - 업캐스팅을 통해 오버라이딩된 메소드들이 호출되도록 구현

- InviteFunction
  - int counter; -> 생성자에서 0으로 초기화
  - int increseCounter()
    - 호출 될때마다 counter ++;
  - int printCounter() : counter 값 출력
  - int resetCounter();
    - counter의 값을 0으로 리셋
  - void showGeetMsg();
  - int login(String id, String pw);
  - int logout();
- UpgradeInviteFunction
  - static totalCounter=0;
  - int increseCounter() : 메소드 오버라이딩
    - 호출 될때마다 counter ++, totalCounter++;
  - int printCounter() : 메소드 오버라이딩
    - counter, totalCounter 값 출력
- CounterApp
  - main()
    - InviteFunction 의 객체를 한개 생성
      - InviteFunction fi = new InviteFunction ();
    - InviteFunction 객체의 increseCounter()를 10번 호출되도록 구현
    - InviteFunction 객체의 printCounter() 호출
    - 업캐스팅의 방법으로 UpgradeInviteFunction의 객체를 두개 생성
      - uiif1, uif2
    - 반복문을 통해 uif1 객체의 오버라이딩된 increseCounter ()가 10번 호출되도록 구현, printCounter() 호출
    - 브릿지메소드 호출 : uif2 객체를 전달
  - 브릿지메소드
    - 메소드 오버라이딩된 메소드들 호출



- music
  - String songName
  - String size
  - String getMusicName();
    - songName 리턴
- Effect
  - String effName
  - int effKind;
  - String getEffect();
    - effKind리턴
  - boolean addEffect();
- SoundOutput
  - int volume;
  - String company;
  - int sendSound(music m);
  - int getVolume();
  - int setVolume();
  - int configequalizer();
- WooferOutout
  - Effect wooferEffect;
    - 생성자의 매개변수로 전달되며 저장되어야 함
  - int sendSound(music m); : 오버라이딩
  - int addWooferEffect();
    - Effect의 addEffect(); 호출
  - company 를 "oldi" 로 설정
- MusicApp
  - company 출력
  - 업캐스팅을 통해 오버라이딩된 메소드들이 호출되도록 구현

abstract

- Color
  - int Rgb\_r;
  - int Rgb\_g;
  - int Rgb\_b;
  - int getRGBValue()
  - int setRGBValue(int r, int g, int b)
- Shape
  - int setColor(Color col)
  - Color getColor()
  - abstract int draw(Color col);
- abstract 메소드 구현 클래스
  - TabletDraw
  - PCDraw

- Browser
  - String[] getBookmark()
  - void drawImage()
  - int printString(String s)
  - int getPointer(int x, int y)
  - abstract int initNetwork();
  - abstract int endNetwork();
  - abstract int sendData(int size, byte[] data);
  - abstract int receiveData(int size, byte[] data);
- 상속 클래스 : NetworkModule

- SoundOutput
  - int setMusic(Music music)
    - music 필드에 값 저장
  - Music music;
  - int volume;
  - String company;
  - int getVolume();
  - int setVolume();
  - int configequalizer();
  - int sendSound(music m);
  - abstract int addWooferEffect();
  - abstract int mixEffect(music m);

## ●music

```
String songName
String size
String getMusicName();
    songName 리턴
```

- Effect :sound 클래스의 abstract 메소드 구현
- MusicApp
  - 업캐스팅을 통해 구현된 abstract 메소드들이 호출되도록 구현

# 컬렉션