1. 상속

- 1. 상속은 이미 존재하는 클래스를 재사용하여 새로운 클래스를 만들어내는 작업.
- 2. 상속은 코드의 재사용성을 높여주고 쓸데없이 작성된 코드를 제거할 수 있다.
- 3. 상속을 해준 이미 존재하는 클래스를 부모클래스, 상속을 받아 새로 만들어진 클래스는 자식(하위)클래스라고 부른다.
- 4. 자식클래스에서는 부모클래스에 존재하는 멤버변수나 메소드를 사용할 수 있다.
- 5. 공통된 속성과 기능들을 부모클래스만 가지고 있고 자식클래스는 부모 클래스의 속성과 기능을 이용하기 때문에 소스코드의 재사용성이 높아 지고 자식클래스에서 공통된 코드가 사라지기 때문에 소스코드를 단축 할 수 있다.

```
public class Parent {
  int height;
  String haircolor;
  String skincolor;
  public void eat() {
  }
  public void sleep() {
  }
public class Child extends Parent {
  int weight;
  String speakType;
  int friendCnt;
  public void programming() {
  }
}
public static void main(String[] args) {
  Parent p = new Child();
}
```

- 6. 자식클래스에는 자식클래스만의 속성이나 기능을 추가하여 부모클래스 와는 차별된 객체를 만들 수 있다.
- 7. 자식클래스는 부모클래스를 상속받았기 때문에 부모클래스 형태도 가지고 있고 자식클래스의 형태도 가지게 된다. 부모클래스 타입으로 선언된 변수에 자식클래스로 만들어진 객체를 담을 수 있게 된다.
- 8. 상속 방법
- extends 키워드를 사용한다.
- public 자식클래스 extends 부모클래스 {

} 상속 extends 키워드를 부모클래스 { 자식클래스 extends 부모클래스 { 통한 상속 자식의 속성, 부모의 속성, 부모의 기능 자식의 기능 인스턴스화 자식객체 { 자식객체 부모의 속성, 부모의 멤버변수 자식의 속성, 자식의 멤버변수 부모의 기능, 자식의 기능 Heap 메모리 영역 부모의 메소드와 자식의 메소드 - Heap 메모리에 저장되는 참조 타입들은 동적 크기로 모두 static 영역에 저장. 할당 객체 자체와 멤버변수들은 heap 클래스마다 멤버변수의 개수 영역에 저장. 객체가 생성되면서 부모의 속성 타입들이 서로 다르기 때문에 바이트 수를 지정할 수 없기 (멤버변수)들도 함께 저장 때문에 동적으로 크기 지정 9.

2. 오버라이딩

- 1. 부모클래스의 메소드를 자식클래스에서 재정의하는 것.
- 2. 같은 메소드로 다양한 기능을 만들 수 있어 다형성의 기초가 된다.

