class Data2{

int x;

}

public class Ex6\_7 {

public static void main(String[] args) {

Data2 d = new Data2(); //Create the instance with variable d

d.x = 10; //

System.out.println("main() x: " + d.x);

change(d);

System.out.println("After change(d)");

System.out.println("main() : x = " d.x)

}

static void change(Data2 d) { //참조형 매개 변수

d.x = 1000;

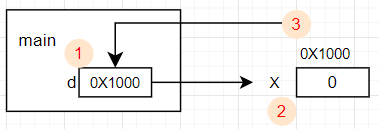
System.out.println("change : x " + d.x);

}

}

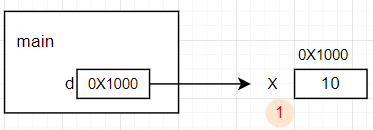
Data2 d = new Data2();

1. Main method호출 되면서 stack에 main method 올라간다
   1. 변수 d가 main method에 생성
   2. Data 2의 객체 생성
2. Data 2의 멤버가 x 하나이기 때문에 X하나만 생성
   1. Int 변수는 처음에 자동 초기화로 인하여 0의 값을 갖는다.
3. Referencing
   1. 0X100 이라는 객체의 주소가 main method의 변수 d에 저장된다.



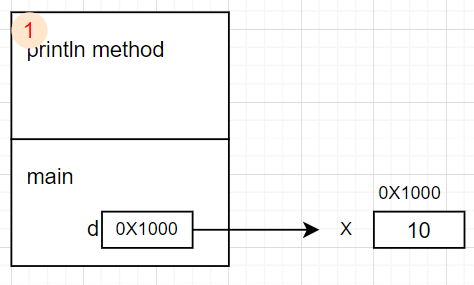
d.x = 10;

1. 참조 변수 d 가 가르키는 변수 X의 값에 10을 대입



System.out.println("main() x: " + d.x);

1. Println method 실행 이후 종료
2. 결과 값으로 main() : x = 1000 출력

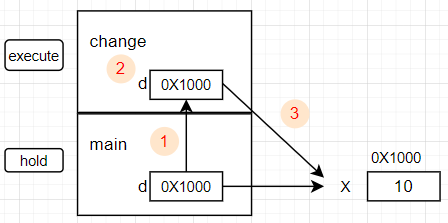


change(d);

static void change(Data2 d) { //참조형 매개 변수

}

1. Change method 호출 되면서
   1. Main method는 잠시대기
2. Change method 실행
   1. 참조형 매개변수 d로 main method의 d값인 주소값이 복사
3. Change method의 d 값도 주소 값을 가지고 있기때문에, X를 가르키게 된다.



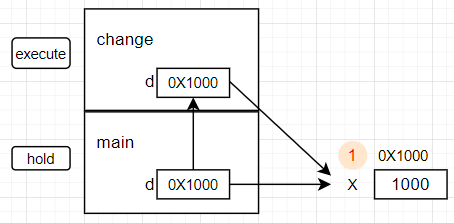
static void change(Data2 d) { //참조형 매개 변수

d.x = 1000;

System.out.println("change : x " + d.x);

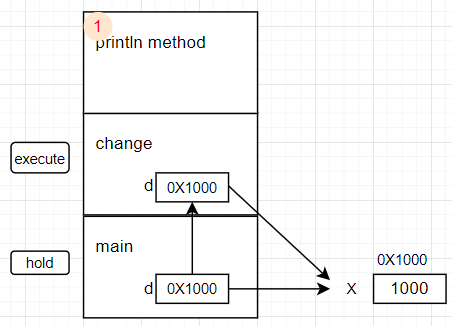
}

1. Change method 의 d가 가르키는 x의 값을 1000으로 변경



System.out.println("change : x " + d.x);

1. Println method 실행 이후 종료
2. 결과 값으로 change: x = 1000 출력



Change method 종료 이후 change method 호출 지점으로 되돌아 간다.

change(d);

System.out.println("After change(d)");

System.out.println("main() : x = " d.x)

}

static void change(Data2 d) { //참조형 매개 변수

d.x = 1000;

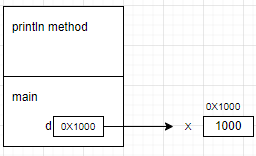
System.out.println("change : x " + d.x);

System.out.println("After change(d)");

1. After change(d) 출력

System.out.println("main() : x = " d.x);

1. main() : x =1000 출력



Stack은 종료되면서 프로그램 종료

class Data3{

int x;

}

public class Ex6\_8 {

public static void main(String[] args) {

Data3 d = new Data3();

d.x = 10;

Data3 d2 = copy(d);

System.out.println("d.x = " + d.x);

System.out.println("d2.x = " + d2.x);

}

static Data3 copy(Data3 d) {

Data3 tmp = new Data3();

tmp.x = d.x;

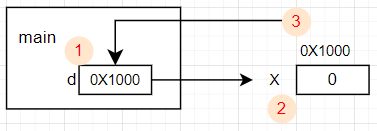
return tmp;

}

}

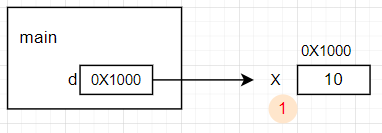
Data3 d = new Data3();

1. Main method호출 되면서 stack에 main method 올라간다
   1. 변수 d가 main method에 생성
   2. Data 2의 객체 생성
2. Data 2의 멤버가 x 하나이기 때문에 X하나만 생성
   1. Int 변수는 처음에 자동 초기화로 인하여 0의 값을 갖는다.
3. Referencing
   1. 0X100 이라는 객체의 주소가 main method의 변수 d에 저장된다.



d.x = 10;

1. 참조 변수 d 가 가르키는 변수 X의 값에 10을 대입



copy(d);

static Data3 copy(Data3 d) {

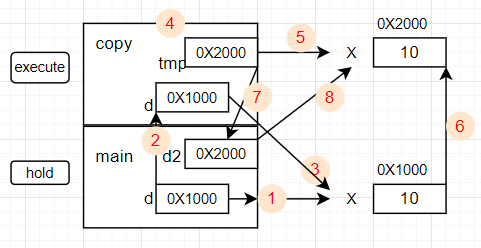
Data3 tmp = new Data3();

tmp.x = d.x;

return tmp;

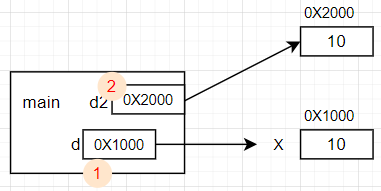
}

1. copy method 호출되면서 d2 변수 생성
2. main method의 d 값(주소 값)이 copy method의 매개변수 d로 복사된다
3. copy method의 변수 d가 가르키는 x의 값을 가리킨다
4. tmp 변수가 copy method에 생성되면서 data3 객체 생성
5. 객체 생성되면서 0x2000 tmp에 저장
6. 변수 d 가 가르키는 x의 값을 tmp변수가 가르키는 x에 저장
7. Tmp 값인 객체 주소 반환 후 copy method 호출한 곳으로 돌아간다
   1. 이때 return tmp의 statement로 인하여
   2. Tmp의 값인 주소값이 반환되면서 d2에 대입된다.
   3. D2의는 0X2000의 주소값을 가지면서 data3객체의 변수 X를 가르키게된다.



Data3 d2 = copy(d);

1. Copy method가 종료되고 나면 d2의 주소값이 대입



System.out.println("d.x = " + d.x);

System.out.println("d2.x = " + d2.x);

1. Stack에 print method가 올라가면서
   1. D.x가 가리키는 10을 출력
   2. D2.x가 가리키는 10을 출력

