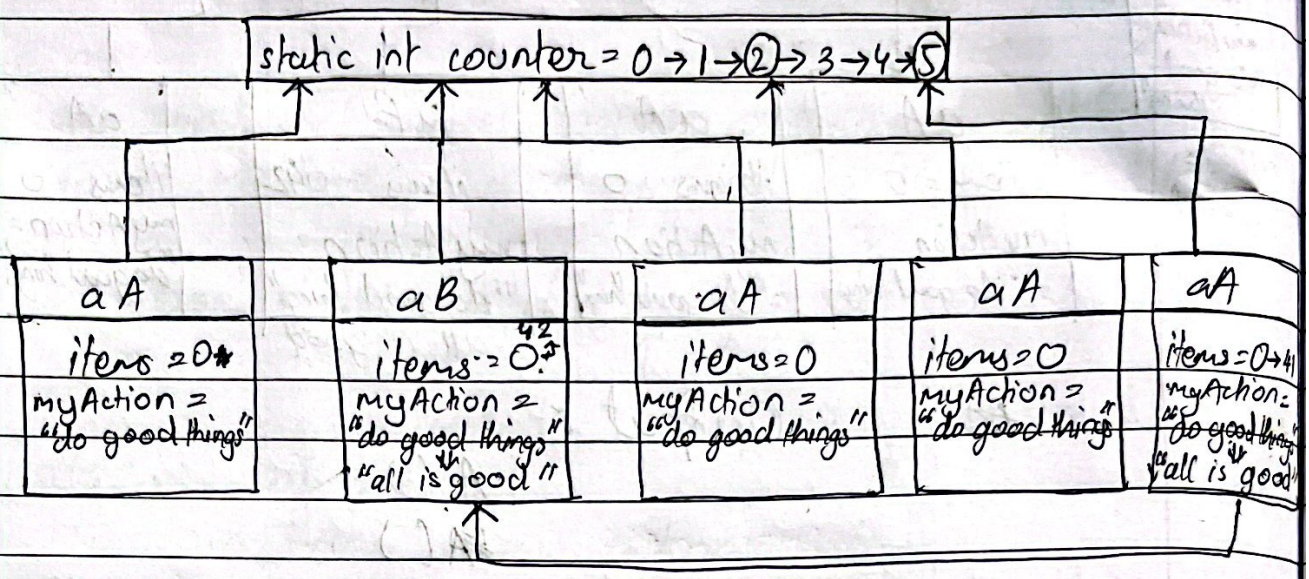
Srivenkatesh Nair (sn6711)
Abhinav Menon (am6176)

Homework 6.1



Output:

A()
A()
B()
a do good things/0/2/b do good things/0/2
A()
A(int x)
A()
B(int x)
A()
A(int x)
a all is good/41/5/b do good things/0/5
b all is good/42/5/b all is good/42/5
b all is good/42/5/b all is good/42/5

Explanation:

In Main function:

- Line 4: A aA = new A();

→ In this line we create an object 'aA' using the default constructor of the parent class A.

→ Object aA has the instance variables items = 0 and myAction = "do good things" and it also points to the static variable counter which is initially equal to 0.

→ Within the default constructor of class A we first print "A()" and then increment counter by 1.

- line 5: B aB = new B()

→ In this line we create an object 'aB' using the default constructors of parent class A and child class B.

→ Object aB has the instance variables items = 0 and myAction = "do good things" of class B and it also points to the static variable counter which is equal to 1.

→ Within the default constructor of class A we first print "A()" and then increment counter by 1.

→ Within the default constructor of class B we just print "B()"

- line 6: System.out.println(aA + "/" + aB);

→ To print "aA" we go to the toString() method within class A which returns a do good things/0/2

→ To print "aB" we go to the toString() method within class B which returns b do good things/0/2.

- line 7: aA = new A(1);

→ In this line we create another object 'aA' using the ~~default constructor of~~ parameterised constructor of class A

→ Within the parameterised constructor, this() calls the default constructor of class A and

then prints "A(int x)".

- Line 8: aA = new B(2);
→ This line calls the parameterised constructor of class B to create an object "aA".
→ Within the parameterised constructor, super() keyword calls the default constructor of parent class A. and then prints "B(int x)".

- Line 9: aA = new A(3);
→ The line call the parameterised constructor of class A to create another object "aA".
→ Within the parameterised constructor, this() keyword calls the default constructor of class A and then prints "A(int x)

- Line 10: aA.setIt("all is good");
→ This line calls the setIt() on the most recent object "aA". It calls the setIt() method of class A.
→ within the setIt() method, we change the string literal of the String myAction to "all is good".

- Line 11: aA.setSoManyItems(41);
→ This line calls the setSoManyItems() method on the most recent object "aA". It calls the setSoManyItems() method of class A.
→ Within the setSoManyItems() method, we change the value of items from 0 to 41.

- Line 12: System.out.println(aA + "/" + aB);
→ Similar to line 6, we go to the toString()

method of the respective objects to print "a all is good/41/5/6 do good things/0/5.

- Line 13: aB.setIt("all is good");
→ This line calls setIt method on the object "aB". It calls the setIt() method of class B.
→ Within the setIt() method, we change the string literal of the String myAction to "all is good".

- line 14: aB.setSoManyItems(42);
→ This line calls the setSoManyItems() method on the object "aB". It calls the setSoManyItems() method of class B.
→ Within the setSoManyItems() methods, we change the value of items from 0 to 42.

- Line 15: aA = aB
→ In this line, we are using the assignment operator "=" to point object "aA" to object "aB"
→ Using this line, we basically mean that both the objects will pointing to the same values of the instance variables.

- line 16: System.out.println("aA + "/" + aB);
→ Similar to line 12, 4 & 6, we go to the toString() method of the ~~respective~~ objects aA & aB to print "b all is good/42/5/6 all is good/42/5.

- Line 17: aA = (A)aB
→ In this line, we are using the assignment operator "=" to point object "aA" to object "aB".

using class casting.

→ The line 15 also does the same thing.

• Line 18: System.out.println(aA + "/" + aB).

→ Similar to line 16, we go to String() method of class B to print "b all is good/42/5/b all is good/42/5".

* "aA = aB" is legal because the compiler recognizes that it is a case of class-casting and functions like "aA = (A) aB".