# 주의사항

주어진 예제 코드를 수정하지 않고, 새 코드를 추가하기만 해야 함.

# 프로젝트 업로드

## 프로젝트명 D학번

예:) D201032054

프로젝트명이 다른 경우에 감점(-5)

## 패키지명과 파일명

주어진 문제와 package명과 파일명이 동일해야 함.

다른 경우에 감점(-5)

## 프로젝트 제출

프로젝트 폴더만 압축해서 제출하라 (예: D201032054 폴더)

제출할 파일명: C학번**-일련번호.zip** 예: D201032054-1.zip

프로젝트 폴더가 아닌 다른 폴더를 압축해서 제출한 경우 감점(-5)

**중간 제출:**

30분 마다, 프로젝트폴더를 zip으로 압축하여 제출하라.

파일명에 한글이 포함되면 업로드할 때 에러 발생함.

제출된 파일은 다운로드 할 수 없고, 삭제할 수도 없다.

만약 이미 업로드한 파일에 문제가 있어서, 다시 업로드해야 한다면,

일련번호를 더한 파일명으로 새로 업로드하라. (예: D201032054-2.zip, D201032054-3.zip)

**최종 결과물 제출:**

제출할 파일명은 C학번-일련번호.zip (예: D201032054-2.zip) (파일명에 한글이 포함되면 절대 안 됨)

만약 이미 업로드한 파일에 문제가 있어서, 다시 업로드해야 한다면,

일련번호를 더한 파일명으로 새로 업로드하라. (예: D201032054-5.zip)

일련번호가 가장 큰 파일이 최종 결과물로 인정된다.

## 무선 네트웍

교내 무선 네트웍에 연결해야 ftp 업로드 가능함

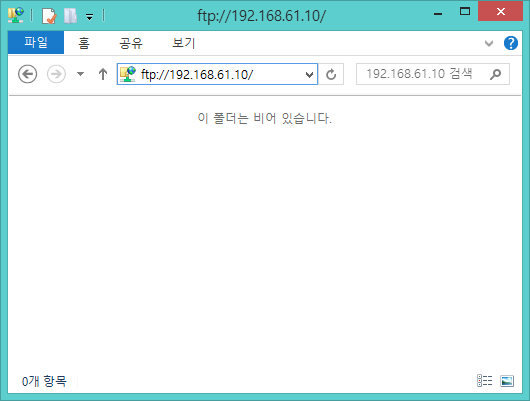
usb 메모리를 이용해서 강의실 PC에서 업로드 해도 됨.

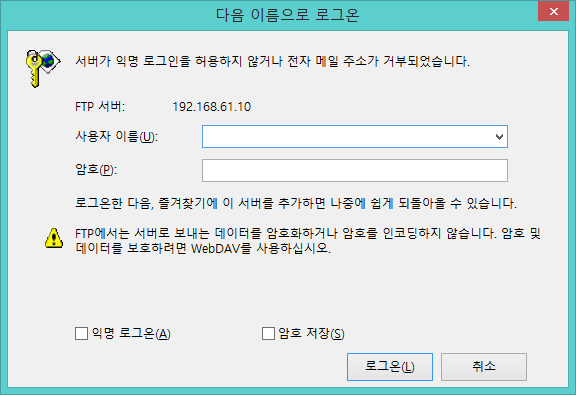
## ftp 업로드 방법

Windows 탐색기의 주소칸에

ftp://192.168.61.10

를 입력하고 엔터 키를 누른다.





사용자 이름: upload

암호: upload

### 업로드에 문제가 있으면

https://filezilla-project.org/

FileZilla FTP Client 프로그램을 다운로드하고 설치하여

이 프로그램을 이용해서 업로드하라.

## 업로드 잘되었는지 확인하기

Windows 탐색기를 종료한다.

Windows 탐색기를 다시 실행한다.

ftp 서버에 연결한다.

파일 목록에서 내가 제출한 파일명을 찾는다.

# 구현 문제

## Exam01.java

|  |
| --- |
| package e2018.exam2;  public class Exam01 {  public static double average(int... a) {  }  public static void main(String[] args) {  int[] a = { 2, 3, 4, 5 };  System.out.println(average(2, 3));  System.out.println(average(2, 3, 4));  System.out.println(average(a));  }  } |

출력

|  |
| --- |
| 2.5  3.0  3.5 |

파라미터로 주어진 정수들의 평균값을 리턴하는 average 메소드를 구현하라.

## Exam02.java

Person.java

|  |
| --- |
| package e2018.exam2;  import java.util.Objects;  public class Person {  String name;  int age;  public Person(String name,int age) {  this.name = name;  this.age = age;  }  } |

Exam02.java

|  |
| --- |
| package e2018.exam2;  import java.util.Arrays;  public class Exam02 {  static String toString(Person[] a) {  }  public static void main(String[] args) {  Person[] a = new Person[] { new Person("홍길동", 18), new Person("임꺽정", 21) };  System.out.println(toString(a));  }  } |

출력

|  |
| --- |
| [{"홍길동", 18}, {"임꺽정", 21}] |

실행 결과 출력이 위와 같도록

Exam02 클래스의 toString 메소드를 구현하라. (Arrays 클래스 메소드만 사용해서 구현해야 함)

Person 클래스도 수정해야함.

## Exam03.java

|  |
| --- |
| package e2018.exam2;  import java.util.Arrays;  public class Exam03 {  static boolean compareIdentity(Person[] a1, Person[] a2) {    }  static boolean compareEquality(Person[] a1, Person[] a2) {    }  public static void main(String[] args) {  Person[] a1 = new Person[] { new Person("홍길동", 18), new Person("임꺽정", 21) };  Person[] a2 = new Person[] { new Person("홍길동", 18), new Person("임꺽정", 21) };  System.out.println(compareIdentity(a1, a2));  System.out.println(compareEquality(a1, a2));  }  } |

출력

|  |
| --- |
| false  true |

실행결과 출력이 위와 같도록 compareIdentity, compareEquality 메소드를 구현하라.

compareIdentity 메소드: identity 비교 결과를 리턴

compareEquality 메소드: equality 비교 결과를 리턴

## Exam04.java

|  |
| --- |
| package e2018.exam2;  import java.util.Arrays;  public class Exam04 {  static Person[] clone(Person[] a) {  }  static boolean compareIdentity(Person[] a1, Person[] a2) {  }  static boolean compareEquality(Person[] a1, Person[] a2) {    }  public static void main(String[] args) {  Person[] a1 = new Person[] { new Person("홍길동", 18), new Person("임꺽정", 21) };  Person[] a2 = clone(a1);  System.out.println(compareIdentity(a1, a2));  System.out.println(compareEquality(a1, a2));  }  } |

출력

|  |
| --- |
| false  true |

compareIdentity, compareEquality 메소드는 앞 문제와 동일하게 구현하라.

### clone 메소드 구현

파라미터로 주어진 Person[] 배열을 복제해서 리턴하도록 구현하라.

Arrays 클래스 메소드를 이용해서 구현하라.

## Exam05.java

|  |
| --- |
| package e2018.exam2;  import java.util.ArrayList;  import java.util.Collection;  import java.util.Iterator;  public class Exam05 {  static void removeEvenNumbers(Collection<Integer> c) {  }  public static void main(String[] args) {  Collection<Integer> c = new ArrayList<Integer>();  for (int i = 0; i < 10; ++i)  c.add(i);  removeEvenNumbers(c);  System.out.println(c.toString());  }  } |

출력

|  |
| --- |
| [1, 3, 5, 7, 9] |

### removeEvenNumbers 메소드 구현

파라미터 Collection 객체에서 짝수를 모두 제거하라.

## Exam06.java

|  |
| --- |
| package e2018.exam2;  import java.util.ArrayList;  import java.util.Arrays;  import java.util.Collection;  public class Exam06 {  static void intersection(Collection<String> c1, Collection<String> c2) {  }  public static void main(String[] args) {  Collection<String> c1 = new ArrayList<String>();  Collection<String> c2 = new ArrayList<String>();  c1.addAll(Arrays.asList("a", "b", "c", "d"));  c2.addAll(Arrays.asList("b", "d", "e", "f"));  intersection(c1, c2);  System.out.println(c1.toString());  }  } |

출력

|  |
| --- |
| [b, d] |

### intersection 메소드 구현

c1과 c2에 모두 들어있는 값만, c1에 남아있어야 한다.

addAll, removeall, retailAll 메소드를 사용해서 구현하라.

## Exam07.java

|  |
| --- |
| package e2018.exam2;  import java.util.ArrayList;  import java.util.Arrays;  import java.util.Collection;  public class Exam07 {  static Collection<String> intersection(Collection<String> c1, Collection<String> c2) {    }  public static void main(String[] args) {  Collection<String> c1 = Arrays.asList("a", "b", "c", "d");  Collection<String> c2 = Arrays.asList("b", "d", "e", "f");  Collection<String> c3 = intersection(c1, c2);  System.out.println(c1.toString());  System.out.println(c2.toString());  System.out.println(c3.toString());  }  } |

출력

|  |
| --- |
| [a, b, c, d]  [b, d, e, f]  [b, d] |

### intersection 메소드 구현

새 ArrayList 객체를 생성하여 리턴하라.

리턴되는 객체에는, c1과 c2에 모두 들어있는 값만 들어있어야 한다.

c1, c2는 변경되지 말아야 한다.

addAll, removeall, retailAll 메소드를 사용해서 구현하라.

## Exam08.java

|  |
| --- |
| package e2018.exam2;  import java.util.ArrayList;  import java.util.Arrays;  import java.util.Collection;  public class Exam08 {  static Collection<String> intersection(Collection<String> c1, Collection<String> c2) {  }  public static void main(String[] args) {  Collection<String> c1 = Arrays.asList("a", "b", "c", "d");  Collection<String> c2 = Arrays.asList("b", "d", "e", "f");  Collection<String> c3 = intersection(c1, c2);  System.out.println(c1.toString());  System.out.println(c2.toString());  System.out.println(c3.toString());  }  } |

출력

|  |
| --- |
| [a, b, c, d]  [b, d, e, f]  [b, d] |

### intersection 메소드 구현

새 ArrayList 객체를 생성하여 리턴하라.

리턴되는 객체에는, c1과 c2에 모두 들어있는 값만 들어있어야 한다.

c1, c2는 변경되지 말아야 한다.

add, remove, contains 메소드를 사용해서 구현하라.

## Exam09.java

|  |
| --- |
| package e2018.exam2;  interface Parent {  void method1();  }  class Child1 implements Parent{  @Override  public void method1() {  System.out.println("Child1");  }  }  class Child2 implements Parent{  @Override  public void method1() {  System.out.println("Child2");  }  }  public class Exam09 {  static void doSomething(Child1 child) {  child.method1();  }  static void doSomething(Child2 child) {  child.method1();  }  public static void main(String[] args) {  Child1 child1 = new Child1();  Child2 child2 = new Child2();  doSomething(child1); // 다형성 호출이 아님  doSomething(child2); // 다형성 호출이 아님  }  } |

출력

|  |
| --- |
| Child1  Child2 |

다형성 호출 구조가 되도록 수정하라.

불필요한 소스 코드를 삭제해도 됨.

## Exam10.java

|  |
| --- |
| package e2018.exam2;  public class Exam10 {  static int sum(int[][] a) {    }  public static void main(String[] args) {  int[][] a = new int[][] {  { 1, 2 },  { 3, 4, 5 },  { 6, 7, 8, 9 }  };  System.out.println(sum(a));  }  } |

출력

|  |
| --- |
| 45 |

### sum 메소드 구현

2차원 배열의 합계를 계산해서 리턴하라.

## Exam11.java

|  |
| --- |
| package e2018.exam2;  import java.util.ArrayList;  import java.util.Collection;  import java.util.Objects;  public class Exam11 {  static class Data {  String value;  public Data(String value) {  this.value = value.toUpperCase();  }  }  public static void main(String[] args) {  Collection<Data> a = new ArrayList<Data>();  a.add(new Data("a"));  a.add(new Data("b"));  a.add(new Data("c"));  a.remove(new Data("B"));  System.out.println(a.toString());  }  } |

출력

|  |
| --- |
| [Data("A"), Data("C")] |

실행결과 출력이 위와 같도록 Data 클래스를 수정하라.

## Exam12.java

|  |
| --- |
| package e2018.exam2;  class Hello {  World world;  public Hello(World world) {  this.world = world;  }  public void say() {  System.out.println(world.message);  }  }  class World {  String message = "안녕하세요";  public void doSomething() {  Hello hello = new Hello(this);  hello.say();  }  }  public class Exam12 {  public static void main(String[] args) {  World world = new World();  world.doSomething();  }  } |

inner class 문법을 활용하여 Hello 클래스를 다시 구현하라.

## Exam13.java

|  |
| --- |
| package e2018.exam2;  interface OnCreateListener {  void onCreate();  }  class MyOnCreateListener implements OnCreateListener {  @Override  public void onCreate() {  System.out.println("MyOnCreateListener onCreate");  }  }  class MessageBox {  OnCreateListener listener;  public void setButton(OnCreateListener listener) {  this.listener = listener;  }  public void create() {  System.out.println("MessageBox create");  listener.onCreate();  }  }  public class Exam13 {  public static void main(String[] args) {  MessageBox messageBox = new MessageBox();  MyOnCreateListener listener = new MyOnCreateListener();  messageBox.setButton(listener);  messageBox.create();  }  } |

annonymous inner class 문법을 적용하여 소스코드를 수정하라.

## Exam14.java

|  |
| --- |
| package e2018.exam2;  import java.util.Arrays;  public class Exam14 {  static void sort(int[] a) {  }  static boolean contains(int[] a, int value) {  }  public static void main(String[] args) {  int[] a = { 3, 8, 1, 2, 4, 5, 7 };  sort(a);  System.out.println(Arrays.toString(a));  System.out.println(contains(a, 2));  System.out.println(contains(a, 6));  }  } |

출력

|  |
| --- |
| [1, 2, 3, 4, 5, 7, 8]  true  false |

sort 메소드와 contains 메소드를 구현하라.

Arrays 클래스 메소드를 사용해서 구현하라.

## Exam15.java

|  |
| --- |
| package e2018.exam2;  import java.util.Arrays;  public class Exam15 {  static void sort(int[] a) {  }  static int[] insert(int[] a, int value) {    }  public static void main(String[] args) {  int[] a = { 3, 8, 1, 2, 4, 5, 7 };  sort(a);  System.out.println(Arrays.toString(a));  a = insert(a, 2);  a = insert(a, 6);  System.out.println(Arrays.toString(a));  }  } |

출력

|  |
| --- |
| [1, 2, 3, 4, 5, 7, 8]  [1, 2, 3, 4, 5, 6, 7, 8] |

sort 메소드와 insert 메소드를 구현하라.

구현 방법에 제한 없음.