



## 자바1 수업차시

1.	<p>오전 : 자바설치, 자바로 코드 작성법</p> <p>순서도 작성법 : 변수의 역할 , 약속된 기호로 절차표현 (수 맞추기게임, 1~10까지 합 - 체계적으로 풀어보기 )</p>	<p>문제풀이 - 체계적인 문제 풀이 , 순서도 작성하기</p> <pre>public class Hello{      public static void main(String[] args ) {         System.out.println( "hello java");     }  }</pre>
2.	체계적인 문제 풀이 방법 - 자료명세 , 처리과정, 순서도 작성하기 ( 하루 반 정도 문제 풀이 시간을 갖는다)	
3.	오전 : 문제풀이 발표	<p>변수, 연산자, 제어구조 학습</p> <p>문제풀이를 코드로 구현해보자</p>
4.	문제풀이 코딩 (풀이한 문제를 코딩해보기)	
5.	배열	<p>배열의 사용하는 이유</p> <p>배열관련문제 풀이</p> <p>내가 좋아하는 숫자 6개 배열로 저장하자 !</p> <pre>int[] number =new int[3]; int[] number = new int[] { 7,8,9} ;</pre>
6.	<p>배열2</p> <p>-new의미 (동적메모리 , 정적메모리)</p>	<p>C:메모리관리를 직접한다</p> <p>java: 메모리관리를 직접하지 않는다</p> <p>1차원배열</p> <p>2차원배열</p> <p>3차원배열</p>
7.	함수	<p>함수:</p> <p>명령의 집합 (코드 덩어리)</p> <p>함수보내기</p> <p>-청소년 판단 함수 만들기</p> <p>-1~10까지합 함수 만들기</p> <p>-오모라이스만드는 함수 만들기</p>
8.	함수2	라이브러리 만들기
9.	<p>클래스</p> <p>-type으로써의 클래스</p> <p>-절차지향과 객체지향 코드</p>	<p>고객정보를 저장할 구조화된 변수가 필요해 !!</p> <p>구조화된 데이터를 처리하는 변수가 없다면?</p> 
10.	<p>클래스 , 캡슐화 -생성자, this, 패키지 ,접근제어자</p> <p>( 생성자오버로딩)</p>	<p>객체지향코드 작성하기</p> <p>-this 파헤치기</p>



		Comparable 인터페이스 Comparator 인터페이스
16차시	예외처리하기 파일다루기	예외를 처리한다는 것은? - checked Exception - unchecked RuntimeException
프로젝트	4일 , 발표 하루	
21차시	멀티쓰레드와 네트워크 프로그램	- 채팅프로그램 만들어보기

자바2 수업차시

1일차	라이브러리를 만들 때 인터페이스 활용	세가지 매서드 라이브러리 만들기 - 별을 3번 출력하는 매서드 만들기 - 원하는 문자를 3번 출력하는 매서드 만들기 - <b>원하는 코드를 3번 실행해 주는 매서드 만들기</b> (팁! 인터페이스가 필요함)
	제네릭 클래스 복습	- ArrayList, GList, OList
	제네릭 매서드 만들기	
	매서드인자에서 와일드카드 사용하기	- ? => 같은 의미 <? extends Object> - <? extends T> T와 T를 상속받은 클래스 의미 - <? super T> T와 T이상 (부모들)
	인터페이스를 활용한 요리사 찾기프로그램	 중식이가능한,일식이가능한, 한식이가능한
2일차	함수형인터페이스란	
	함수형인터페이스를 구현하는 3가지 방법	- 이름있는 클래스로 구현 - 익명 클래스로 구현 - 랴다식으로 구현
	- Runnable : void run() - Consumer<T> : void accept( T t) - Supplier<T> : T get() - Predicate<T> : Boolean test( T t) - Function<T,R> : R apply( T t)	함수형인터페이스는 주로 매서드의 매개변수로 사용된다 .  ArrayList<String> list = new ArrayList<>(); list.add("장미"); list.add("후리지아");  list.forEach( item -> System.out.println( item) );
3일차	Collection - List, Set, Map	- List 인터페이스를 구현한 ArrayList (배열처럼 쓰고싶을 때, 순서있음, 중복허용) - Set 인터페이스를 구현한 HashSet (중복을 허용하지 않고 저장할 때) - equals 와 hash 를 오버라이딩해야함 (같은객체로 인식되어야 하나만 저장됨)  - Map 인터페이스를 구현한 HashMap ( key, value)로 값을 저장할 때 사용
	스트림 (데이터의 종류에 상관없이 같은방식으로 데이터를 다룰 수 있게 하는 것 !)	String[] stringArray = {"딸기", "포도", "사과"}; Stream<String> stream2 = Stream.of(stringArray); //정렬하고 출력하기 !! stream2.sorted().forEach( s -> System.out.println( s)); ArrayList<String> stringList = new ArrayList<>(); stringList.add("딸기"); stringList.add("포도"); stringList.add("사과"); //정렬하고 출력하기 !! Stream<String> stream = stringList.stream(); stream.sorted().forEach(s -> System.out.println(s));

4일차	<p>변경에 유리한 코드 작성 (인터페이스와 reflect가 사용됨)</p> <p>-인터페이스사용으로 느슨한 결합 -reflect로 객체를 동적으로 생성</p>	<p>-reflect (클래스로더를 통해 읽어온 클래스 정보를 사용하는 기술) 런타임에 클래스, 인터페이스, 매서드 ,필드 분석 및 조작</p> <pre> Class  clazz = Class.forName("com.acorn.Acorn"); Acorn  obj=     (Acorn)clazz.getDeclaredConstructor().newInstance(); </pre>
	<p>객체를 생성하는법</p> <ul style="list-style-type: none"> <li>- new</li> <li>- 클래스정보를 얻어서 객체를 생성할 수 있다</li> </ul>	
	<p>DI개념 (의존성 주입)</p>	<p>-dependency (의존성) -Injection (주입)</p>
5일차	<p>애너테이션 유지범위 사용자애너테이션 만들기 enum 작성하기 재귀매서드 작성하기</p>	<p>재귀매서드 공부법 - 작은범위의 예제로 만들고 재귀과정을 그리면서 이해한다</p> <ul style="list-style-type: none"> <li>- 별5개 출력 재귀매서드</li> <li>- 1~10까지 합 재귀매서드</li> <li>- 배열의 최대값 구하기를 재귀매서드</li> <li>- 하노이탑</li> <li>- 퀵소트 작성하기</li> </ul>