

입문 1 ~ 14	14
초급 15 ~ 67	53
중급 68 ~ 145	78
활용 146 ~ 168	23
응용 169 ~ 180	11
실무 181 ~ 200	20

#### 대학교, 학원

입문, 초급, 중급 - 필수, 기본기 → 수업진도  
 활용 - 응용력 → 과제  
 응용, 실무 → 프로젝트

입문 1 ~ 14	14	필수 문법
초급 15 ~ 67	53	기본 문법, 기본 객체
중급 68 ~ 145	78	객체와 데이터 가공의 이해, 객체 지향 프로그래밍, 기본 문법 응용
활용 146 ~ 168	23	객체 지향과 로직의 만남
응용 169 ~ 180	11	실무를 위한 준비
실무 181 ~ 200	20	실무 맛보기

#### 실무를 위한 준비

##### 문제 해결을 위한 과정

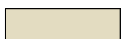
- 151 ~ 162 환율
- 163 ~ 166 거리 환산
- 167 ~ 168 온도 변환
- 169 ~ 177 산책한 거리 구하기
- 178 ~ 180 지진 정보

#### 실무 맛보기

##### 서비스 방법, 전송방법, 가공방법

- 181 ~ 191 2티어 JDBC를 이용 DB와 연동, 그럴듯한 화면
- 191 ~ 197 실무에서 많이 쓰는 Spring Boot를 이용하는 Restful 웹 서비스
- 198 ~ 200 네트워크, 소켓, IO, Thread 사용 3티어

#### javaFX를 사용하는 챕터



#### [표-챕터별 내용]

챕터	일부 소스	내용
1	public static void main(String[] args){}	메인 메서드

2	double latitude = 37.52;	double, int , char
3	Calendar cal=Calendar.getInstance();	java.util.Calendar java.util.Date
4	class JLocation{ }	사용자 정의 참조타입
5	double latitude=Double.parseDouble(slat.trim());	Wrapper java.lang.Double
6	Math.random() /* */	java.lang.Math 코멘트
7	(a+b+c)/3	정수연산
8	int [] grade={90, 80, 89};	배열
9	하이힐	실수연산
10	.length(), indexOf(a)	String
11	public static void showHelp() {	static 메서드
12	Scanner scann=new Scanner(System.in);	키보드입력
13	public static final int PHYSICAL = 23;	상수
14	double vals=(days % index) * 2 * Math.PI / index ;	상수와 연산
15	double phyval=100*Math.sin( (days % index) * 2 * Math.PI / index );	java.lang.Math 메서드, 상수
16	double phyval=getBioRhythm( days, PHYSICAL, 100);	static 메서드
17	BioCalendar bio=new BioCalendar(); double phyval=bio.getBiorhythm( days, PHYSICAL, 100);	non static 메서드
18	FRUIT pear=FRUIT.APPLE;	enum(상수 클래스)
19	double [] latlng1={latitude1,longitude1}; double [] latlng2={latitude2,longitude2}; double [][] latlng={ {latitude1,longitude1}, {latitude2,longitude2} };	double 1차원배열, 2차원배열
20	Geo geo2=new Geo(); geo2.latitude=latitude2; geo2.longitude=longitude2; Geo geo=geo1;	주소복사, 얇은복사, CBR
21	if(temp%2==1){ temp=temp*3+1; } else{ temp=temp/2; }	if ~ else
22	temp = (temp%2==1) ? temp*3+1 : temp/2;	삼항 연산자 ? :
23	switch( index ) { case 23 : result = "신체지수: " ;break; }	switch ~ case, int (fall through 조심)
24	switch( index ) { case PHYSICAL : result = "신체지수: " ;break; }	switch ~ case static final 상수
25	public enum PEI{ PHYSICAL(23), EMOTIONAL(28), INTELLECTUAL(33); ... switch( index ) { case PHYSICAL : result = "신체지수: " ;break; }	switch ~ case enum(값 대입)
26	switch(c){ case 'A': tot=1;break; case 'T': tot=10;break; }	switch ~ case char
27	switch(c){ case "USD": tot=1126.5;break;	switch ~ case String

	case "JPY": tot=110.6;break; }	
28	if( ((year%4==0)&&(year%100!=0))    (year%400==0)){ isS=true; }	논리연산자 &&
29	for (int i = 0; i < 101; i++) { double fahrenheit=9.0/5*i+32; }	for
30	while(temp!=1){ ... }	while
31	do{ ... }while(temp!=1);	do ~ while
32	for( ; temp!=1 ; ){ ... }	for( ; ; ) → while()
33	while(true){ a=(int)(range*Math.random()+start); ... }	while, 서로다른 세수 random()
34	while(a==b    b==c    c==a){ a=(int)(range*Math.random()+start); ... }	while, 서로 다른 세수 random()
35	while(n!=0){ tot+=n%10; // 3 -> 2-> 1 n/=10;     // 123-> 12-> 1->0 }	축약 +=, /=
36	for (int i = 2; i <=(int)Math.sqrt(n); i++) { if(n%i==0){ isS=false; break; } }	java.lang.Math.sqrt() 제 곱근, break 가장가까운 순환문 빠져 나오기, 에라토스테네스의 체
37	int aa=a%2; s=(aa>0)? aa+s : (-aa)+s; a>>=1;	쉬프트 >>, 단축연산자 >>=
38	s=(a & BITMASK)+s; a>>=1;	비트마스크, & 비트연산, 쉬프트 >>, 단축연산자 >>=
39	s.substring(s.indexOf('1'));	비트마스크, &, 쉬프트 >>, >>=, String substring(), indexOf()
40	for (int i = 1; i <= 100; i++) { if(i%2==1){	for { if() }, 합 구하기
41	for (int m : mm) { System.out.printf(m+"\t"); }	배열, 향상된 for
42	public static void print(int ... mm){ for (int m : mm) {	variable arguments ...
43	List<Integer> ilist=new ArrayList<>(); for (int i = 1; i <= 10; i++) {	java.util.ArrayList

	ilist.add(i);	
44	List<Integer> mmlists=Arrays.asList(1,6,16,22,23,33); mmlists.forEach( m -> { System.out.printf(m+"\t"); } );	List, lambda (m)->{ }
45	for(int m: ilist){ if(m%2==1){ s+= m*m;	List, 향상된 for
46	s=ilist.stream().filter(i-> i%2==1).reduce(0, Integer::sum);	List, stream, 홀수의 합
47	ilist=ilist.stream() .filter(i-> i%2==1) .map(i -> i*i) .collect(Collectors.toList()); ilist.forEach(i->{System.out.printf(i+"\t");});	List, Stream, filter, map, collect, lambda
48	value >>>=1;	shit 연산자 >>>
49	try{ int a=Integer.parseInt(sNum); System.out.println(a); }catch(NumberFormatException ee){	try{} catch(){}, 예외처리, NumberFormatException
50	try { int x = 5; int y = 20 / (5 - x); } catch (ArithmeticException e) {	try{} catch(){}, 예외처리, ArithmeticException

챕터	일부 소스	내용
51	obj1.hashCode() obj1.equals(obj2) obj2.toString() obj1.getClass().getName()	Object, 4대 메서드
52	Class classes=obj1.getClass();	Class, reflection
53	System.out.println(city1==city2); System.out.println(city1.equals(city2));	String, ==, equals()
54	System.out.println(text.charAt(2)); System.out.println(text.concat("s")); System.out.println(text.contains("And"));	String 메서드
55	Class classes=Class.forName("java.lang.String"); Method[] mes =classes.getDeclaredMethods();	Class, reflection, String 의 메서드, 생성자
56	StringBuffer sb1=new StringBuffer(); sb1.append("안녕하세요.") .append("또 만나요.") .append("모두를 사랑해요. 기다려봐.");	StringBuffer, mutable String
57	StringBuilder sbu=new StringBuilder(); sbu.append("I") .append(" go") .append(" to shcool.");	StringBuilder, mutable String
58	Properties pro=System.getProperties(); System.out.println(pro.getProperty("java.vm.version")); System.out.println(pro.getProperty("file.separator"));	System 프로퍼티
59	Map<String, String> map=System.getenv(); // 환경변수	환경변수
60	System.out.println("3 : Math.abs(-5) 절댓값:"+Math.abs(-5));	Math abs(), ceil(),

	System.out.println("4 : Math.ceil(4.34) 올림:"+Math.ceil(4.34)); System.out.println("5 : Math.round(4.34) 반올림:"+Math.round(4.5));	round()
61	long millis=System.currentTimeMillis(); Date d=new Date(); Date dd=new Date(d.getTime()+24*60*60*1000); Calendar cal1970=Calendar.getInstance();	java.util.Date java.util.Calendar
62	TimeZone tz=TimeZone.getDefault(); Calendar cal=Calendar.getInstance(tz);	java.util.Calendar java.util.TimeZone
63	if(Character.isLetter(chr1.charValue())){ System.out.print("1 : "+chr1.charValue()); }	Wrapper Character, charValue(), isLetter()
64	Arrays.fill(e, -1); System.arraycopy(c,0,e,0,c.length); Arrays.sort(e);	java.util.Arrays fill(), sort(), 배열 깊은복사 arraycopy()
65	String[] myNum = new String[ ]{"2016-08-17", "2016-09-17", "2016-03-17"}; System.arraycopy(myNum, 0, tos, 0, myNum.length); for (String ss: tos) { System.out.printf(ss+"\t"); }	String 배열, 깊은복사 arraycopy(), 향상된 for
66	for(int i=0;i<c.length;i++){ System.arraycopy(c[i],0,d[i],0,d[i].length); }	2차원 배열의 깊은복사 (arraycopy() 이용)
67	Integer intWrap1=10; Long longWrap1=20L; Double doubleWrap1=35.98;	Wrapper, Boxing, Unboxing

챕터	일부 소스	내용
68	Data의 종류는 다양 기본 데이터(Raw data) -> 가공 -> 구조적 데이터 Data(txt, csv, json, xml)-> 가공 -> 정보(Information) 프로그래밍 data -> 정수, 실수, 문자열, 문자, boolean으로 구분 -> 기본데이터 묶음 -> 객체	DATA 종류 : JSON, XML, CSV -> Object
69	String sf=String.format("%d, %s, %d, %s, %s",rank, song,lastweek,imagesrc,artist);	int, String
70	public static void showAbout(Billbaord bb ){//객체를 전달하여 출력 String sf=String.format("%s, %s, %s, %s, %s", bb.getRank(), bb.getSong(),bb.getLastweek(),bb.getImagesrc(),bb.getArtist()); System.out.println(sf); }	객체 대입
71	public static void showAbout(Billbaord ... b ){ for(Billbaord bill : b){ showAbout(bill); } }	객체, variable arguments
72	ArrayList<Billbaord> bills=new ArrayList<>(); bills.add(new Billbaord(1,"Despacito",1, "q61808osztw.jpg","luis fonsi"));	List
73	File f=new File("billboard"); if(!f.exists()){ f.mkdirs(); }	File, PrintWriter, try ~ catch, AutoClosable

	<pre>try (PrintWriter pw=     new PrintWriter(new FileWriter("billboard\\bill.csv",false),true)){     for(Billbaord bb : bblist){         pw.println(bb);     } } catch (IOException e) {     System.out.println(e); }</pre>	try(){}
74	<pre>try(BufferedReader br=new BufferedReader(     new FileReader("billboard\\bill.csv"))) {</pre>	BufferedReader, try ~ catch, AutoClosable try(){}
75	<pre>System.out.println(new Date(ff.lastModified())); //수정일 try(BufferedReader br=new BufferedReader(     new FileReader(ff.getAbsolutePath())) {     ... } }</pre>	BufferedReader, try ~ catch, AutoClosable try(){}, File, java.util.Date
76	<pre>URL url= new URL(newUrls); BufferedReader reader = new BufferedReader(     new InputStreamReader(url.openStream(), "euc-kr"), 8);</pre>	java.net.URL 주소, BufferedReader, 한 줄씩 읽고 출력
77	<pre>ArrayList&lt;String&gt; htms=new ArrayList&lt;String&gt;(); ... while ((line = reader.readLine()) != null){     htms.add(line.trim()); }</pre>	java.net.URL 주소, BufferedReader, List, 한 줄씩 읽고 리스트에 저장
78	<pre>while ((line = reader.readLine()) != null){     if(!line.trim().equals("")){         if(line.trim().contains("&lt;time datetime=")){</pre>	BufferedReader, List, String equals(), contains(), 특정 문자열 찾기
79	77, 78, 읽어서 특정 문자열 찾기, 파싱	java.net.URL 주소, BufferedReader, List, String equals(), contains()
80	<pre>Billbaord board=new Billbaord(     toInt(rank), replace(song),toInt(__toStr(lastweek)),     imageUrl, replace(artisturl), replace(artist)); billbaords.add(board);</pre> <p>빌보드에서 한줄씩 읽어서 객체로 변환</p>	java.net.URL 주소, BufferedReader, List, 객체
81	<pre>public static void printToCSV(List&lt;Billbaord&gt; bblist, String fname){     try (PrintWriter pw=         new PrintWriter(new FileWriter(fname,false),true)){         for(Billbaord bb : bblist){             pw.println(bb);         }     } catch (IOException e) {     } }</pre> <p>[rank=1, song=Rockstar, lastweek=2, imagesrc=http://charts-static.billboard.com/img/2017/01/post-mal one.jpg, artist=Post Malone]</p>	PrintWriter, try ~ catch, AutoClosable try(){}, CSV
82	<pre>{"billboard": [ {"rank":"1","song":"Rockstar","lastweek":"2","imagesrc":"http://charts- static.billboard.com/img/2017/01/post-malone.jpg","artist":"Post</pre>	PrintWriter, try ~ catch, AutoClosable try(){}, JSON

	Malone"},	
83	<pre>&lt;?xml version='1.0' encoding='utf-8'?&gt; &lt;billboards&gt; &lt;billboard&gt; &lt;rank&gt;1&lt;/rank&gt; &lt;song&gt;Rockstar&lt;/song&gt; &lt;lastweek&gt;2&lt;/lastweek&gt; &lt;imagesrc&gt;http://charts-static.billboard.com/img/2017/01/post-m alone.jpg&lt;/imagesrc&gt; &lt;artist&gt;Post Malone&lt;/artist&gt; &lt;/billboard&gt;</pre>	PrintWriter, try ~ catch, AutoClosable try() {}, XML
84	javaFX	ScenBuilder
85	<pre>Map&lt;String, Image&gt; images=Collections.synchronizedMap( new LinkedHashMap&lt;String, Image&gt;(10, 1.53f, true)); Image mimage= new Image(imagesrc,100,100,true,true); images.put(이름,mimage);</pre>	Map에 (key, Image) 저장
86	85	여러 이미지 보이기
87	<pre>ArrayList&lt;Billbaord&gt; bills=rfw.getBillbaords(); ObservableList&lt;BillbaordProperty&gt; obsbills= BillboardConvert.toObservBill(bills);</pre>	ArrayList<Billbaord> → ObservableList<BillbaordProperty>
88	87, TableView	TableView, Billboard → Youtube
89	87, Accordion	Accordion, Billboard → Youtube

챕터	일부 소스	내용
90	Card c3=new Card();	Card 객체 선언, 생성
91	<pre>Card c1=new Card("H4"); Card c2=new Card("H4"); System.out.println(c1.equals(c2)); //false</pre>	같은 문자열("H4")을 갖지만 다른 카드다→같은 객체는 없다.
92	<pre>Card c1=new Card("H4"); Card c2=new Card("H4"); System.out.println(c1.equals(c2)); // true -&gt; equals() overriding</pre>	같은 문자열("H4")을 갖을 때 같은 객체가 되려면 hashCode(), equals()를 오버라이딩한다.
93	<pre>while(count!=valu*suit){     Card c=new Card();    // 임의의 카드를 만든다     if(!cards.contains(c)){ // contains -&gt; equals()를 이용해 비교         cards.add(c);      // 같은 객체가 아니라면 저장         count++;     } }</pre>	List에 입력받은 카드와 같은 카드가 없다면 List에 대입한다. -> 서로 다른 카드의 묶음을 만든다.
94	<pre>public class CardComp implements Comparator&lt;Card&gt;{     public int compare(Card c1, Card c2) {...} } public void sort(){     cards.sort(new CardComp()); }</pre>	비교의 기준을 제공하는 객체 CardComp를 생성하여 대입한다 -> 정렬할 수 있다.
95	<pre>public void ramdbasort(){     cards.sort((c1, c2) -&gt;         {return c1.getCardVal().compareTo(c2.getCardVal());}); }</pre>	94에서 CardComp과 같은 비교 기준 객체를 제거하여 함수처럼 사용 → (c1,c2) →{... } 람다표현
96	<pre>public void ramdbasort3(){     Comparator&lt;Card&gt; cmp=Card::compareCard;     cards.sort(cmp); // static</pre>	static 메서드에서 ::를 이용하여 정렬을 할 수 있다.

	<pre> } public void rambdasort4(){     cards.sort(Card::compareCard); // static } </pre>	
97	<pre> if(is138(c1) &amp;&amp; is138(c2) ){ }else{ } </pre>	카드의 족보(셋다)를 만든다.
98	97, 셋다 게임 룰	족보를 이용하여 카드 값을 비교
99	96, 바카라 게임 룰	카드의 족보(바카라)를 만든다.
100	96, 바카라 게임 룰	족보를 이용하여 카드 값을 비교
101	<pre> if( x-1&lt;0 ){ // 윗벽     x=n-1; }else{     x--; // 정상 x=x-1; } </pre>	홀수 마방진, 2차원배열
102	<pre> private boolean isMagic( ){     boolean isM=true;     int[ ]m=new int[2*n+2];     ... } </pre>	홀수 마방진, 2차원배열, 마방진 체크로직, 2*n+2개를 비교한다.
103	<pre> if ((i &gt;= 0 &amp;&amp; i &lt; n / 4)    (i &gt;= n / 4 * 3 &amp;&amp; i &lt; n)) {     if (j &gt;= n / 4 &amp;&amp; j &lt; n / 4 * 3) {     }else{     } } </pre>	4배수 마방진, 2차원배열, n/4를 기준으로 좌표이용
104	<pre> OddMagicSquare odd=new OddMagicSquare(n/2); odd.make(); // 3 마방진을 만든다. int[ ][ ] mm=odd.getMagic(); for (int i = 0; i &lt; n/2; i++) {     for (int j = 0; j &lt; n/2 ; j++) {         magic[i][j]+=mm[i][j];         magic[i][j+n/2]+=mm[i][j];         magic[i+n/2][j]+=mm[i][j];         magic[i+n/2][j+n/2]+=mm[i][j];     } } </pre>	6배수 마방진, 2차원배열, 6마방진은 홀수 마방진을 이용(홀수 마방진을 의존: dependency)한다. 6마방 → 3마방
105	<pre> MagicSquare magic=null; if(n&gt;2 &amp;&amp; n%2==1){     magic=new OddMagicSquare(n); }else if(n&gt;2 &amp;&amp; n%4==0){     magic=new FourMagicSquare(n); }else if(n&gt;2 &amp;&amp; n%4==2){     magic=new SixMagicSquare(n); } </pre>	추상클래스를 상속한다. 입력받은 값에 따라 각 마방진을 생성한다. → 팩토리 패턴으로 발전, 다형성을 이용한다.
106	<pre> IMagicSquare magic=null; if(n&gt;2 &amp;&amp; n%2==1){     magic=new OddMagicSquare(n); }else if(n&gt;2 &amp;&amp; n%4==0){     magic=new FourMagicSquare(n); } </pre>	인터페이스를 구현한다. 입력받은 값에 따라 각 마방진을 생성한다. → 팩토리 패턴, 다형성을 이용한다.



	<pre> }else if(n&gt;2 &amp;&amp; n%4==2){     magic=new SixMagicSquare(n); } </pre>	
107	<pre> try(FactoryMagic magicFactory=FactoryMagic.getInstence()) {     IMagicSquare magic = magicFactory.getMagicSquare(n);     MagicPrint.print(magic); } catch (MagicException e) { } </pre>	AutoClosable의 close() 구현 → try( ) { }에서 {}을 끝내면 자동으로 close()호출, 팩토리 패턴, 싱글톤 패턴

챕터	일부 소스	내용
108	<pre> for(int i=0; i&lt;n;i++){     for(int j=0; j&lt;i;j++){ </pre>	이중 for
109	<pre> for (int i = 0; i &lt; n ; i++) {     for (int j = 0; j &lt;Math.abs(i-n/2) ; j++) { </pre>	이중 for, Math.abs() 절대값
110	<pre> while(n!=1){     if(n%a==0){ </pre>	소인수 분해, 10=2x5와 같이 2부터 10을 나눠보고 나누어 떨어지는 수를 찾는다
111	<pre> a=a*(1+r); tot+=a; </pre>	s=a1+a2+a3..., 연속적인 합은 s=s+a 공식사용, a의 규칙만 찾는다.
112	<pre> if( c==s.charAt(i) ){     count++; }else {     t=t+c+count;     c=s.charAt(i);     count=1; } </pre>	같은 문자가 몇 번 들어가는가를 찾는다. String concatenation, 1121은 1이 2번 2가 1번 1이 1번이므로 "122111"이된다.
113	<pre> if(tod.get(Calendar.DAY_OF_WEEK)==Calendar.SATURDAY        tod.get(Calendar.DAY_OF_WEEK)==Calendar.SUNDAY ){     isRest=true; } </pre>	토요일, 일요일은 휴일이다. java.util.Calendar, s62
114	<pre> if(!Java200Math.isPrime(i) &amp;&amp;     Java200Math.sumEach(i)==Java200Math.sumSmith(i)){ </pre>	&,   , 숫수, 스미스넘버, 각자리합과 소인수 분해한 숫자의 합이 같다
115	<pre> cal.set( year,month,13 ); if(cal.get(Calendar.DAY_OF_WEEK)==Calendar.FRIDAY){ </pre>	13일이 금요일인가?, java.util.Calendar
116	<pre> if( ((year%4==0)&amp;&amp;(year%100!=0))    (year%400==0)){ </pre>	윤년을 이용하여 달력 만들기
117	<pre> Calendar cal=Calendar.getInstance(); cal.set(year, month-1,1);//0~11 int dayOfWeek=cal.get(Calendar.DAY_OF_WEEK);// 요일 1~7 </pre>	1일의 요일을 이용, java.util.Calendar API이용
118	<pre> System.out.println(f.getAbsolutePath()); System.out.println(f.getName()); System.out.println(RestDay.toStrDate(new Date(f.lastModified()))); </pre>	java.io.File을 이용, 파일경로, 수정일, 파일명 찾기
119	<pre> if(!newFile.exists()){ //디렉토리가 존재하지 않는가?     boolean iss=newFile.mkdirs(); //존재하지 않으면 만들자     if(iss){         copyUsingFiles(fff, toFile);//파일 복사     } } } </pre>	java.io.File, FileUploadUtil2에 파일 복사법 3개 소개

	<pre> copyUsingFiles(fff, toFile);//파일복사 } // 파일 복사는 3가지 많이 사용 , Files Files.copy(source.toPath(), target.toPath()); // 채널 FileInputStream input = new FileInputStream(source).getChannel(); FileOutputStream output = new FileOutputStream(target).getChannel(); output.transferFrom(input, 0, input.size()); // Stream 이용 InputStream input = new FileInputStream(source); OutputStream output = new FileOutputStream(target); while ((bytesRead = input.read(buf)) &gt; 0) {     output.write(buf, 0, bytesRead); } </pre>	
120	<pre> while(count!=ballNum){     int temp=rand(); //1~45     if(! contain(temp)){         lots[count++]=temp;     } } //Lotto6NumMain2 Lotto6Num lot=new Lotto6Num(1000,100); </pre>	Lotto, 1~45사이의 서로 다른 6개의 수, Lotto6NumMain2는 1~1000사이의 서로다른 100의 수 만들기
121	<pre> for(int i=0; i&lt;n.length-1; i++){     for(int j=0; j&lt;n.length-1-i; j++){  // 소요시간 BubbleSort &gt; SelectSort &gt; InsertSort &gt; QuickSort </pre>	큰 수를 오른쪽으로 이동하고 다시 왼쪽부터 시작, JSortMain2~JSortMain4 에 3개의 소트가 더 있다.
122	<pre> if(i==divide(i)){ // 친화수(2개 수, 5개 수) </pre>	완전수 : 약수의 합(단 자신제외)이 자신, 6=1+2+3
123	서로 다른 두 숫수를 이용, 정수의 특징을 이용	최대공약수(gcd), 숫수
124	<pre> moveHanoi(num-1,ringA,ringC,ringB); System.out.println(ringA+" bar쪽에 있던 링이 "+ringB+" bar쪽으로 이동"); moveHanoi(num-1,ringC,ringB,ringA); </pre>	재귀(자신이 자신을 호출) -반드시 끝나는 조건 필요
125	<pre> moveHanoi(a,num-1,ringA,ringC,ringB); showHanoi(a,ringA,ringB); moveHanoi(a,num-1,ringC,ringB,ringA); //이동과정을 배열로 보임, 124 </pre>	재귀, 2차원 배열
126	<pre> if(d==0){ // 오른쪽으로 이동     if(y+1&lt;n &amp;&amp; loc[x][y+1]!=0){         y++;     }else{ // 오른쪽 벽이나 숫자         d=1; // 아래         x++; // 0.5(y++)=&gt;0.6(x++)-&gt;1,6     } } else if(d==1){ //아래로 이동 </pre>	2차원 배열, 방향 d=0, 1, 2, 3이용, 시계방향
127	<pre> if(d==0){ //2) 오른쪽이동 if(y+1&lt;n &amp;&amp; loc[x][y+1]!=0){     y++; // 오른쪽 }else{ // 오른쪽 벽또는 숫자 </pre>	2차원 배열, 방향 d=0, 1, 2, 3이용, 반시계방향

	<pre> d=3; x--; // 위로     } } else if(d==1){ </pre>	
128	<pre> int[][] pt=new int[n][]; //zigzag for (int i = 0; i &lt; n; i++) {     pt[i]=new int[i+1];    pt[i][0]=1;     for (int j = 1; j &lt;= i; j++) {         pt[i][j]=pt[i-1][j-1]+pt[i-1][j];     }     pt[i][i]=1; } </pre>	<p>지그재그 배열, nCr=n-1Cr-1 + n-1Cr</p>

챕터	일부 소스	내용
129	<pre> if(start&lt;right){     QuickSort(nn,start,right); } if(left&lt;end){     QuickSort(nn,left,end); } </pre>	<p>퀵소트, 재귀, 중앙에 있는 수보다 큰 수는 오른쪽, 작은 수는 왼쪽으로 2등분</p>
130	<pre> if (data[i][ j] &gt; data[i][ k] + data[k][ j]){     p[i][ j] = k;     data[i][ j] = data[i][ k] + data[k][ j]; } </pre>	<p>플로이드 k지점을 지날 때 가중치가 작은가를 판별한다. p2-&gt;p3시작과 끝이라면 2, 3, 사이에 k를 넣어 p2-&gt;pk-&gt;p3를 만들고 이 보다 더 작은 가중치를 만들 수 있는가 확인한다.</p>
131	<pre> count = 1 + findApart(region, x, y - 1, step, depth + 1) + findApart(region, x + 1, y, step, depth + 1) + findApart(region, x, y + 1, step, depth + 1) + findApart(region, x - 1, y, step, depth + 1); </pre>	<p>(x, y)를 기준으로 4방향으로 *가 있는가 확인한다. 침수지(SubmergeLands) 구하기와 같다.</p>
132	<pre> public class Matrix {     private double [][] matrix;     public Matrix(int m, int n) {         matrix=new double[m][n];     } } </pre>	<p>깊은 복사, 복사생성자, 행렬을 구할 때는 이차원 배열을 멤버로 사용하는 객체를 만든다.</p>
133	<pre> c[i][j]=a.getMatrix()[i][j]+b.getMatrix()[i][j]; c[i][j]=a.getMatrix()[i][j]-b.getMatrix()[i][j]; c[i][j]+=a.getMatrix()[i][k]*b.getMatrix()[k][j]; </pre>	<p>행렬의 덧셈, 뺄셈, 곱셈</p>
134	$A^T \cdot X = Y \rightarrow X = A^{-1} \cdot Y$	<p>역행렬 <math>\rightarrow</math> 가우스 소거법</p>
135	<pre> Matrix msdo=MatrixUtil.toTwoOne(new Matrix(basic)); Collections.shuffle(list); msdo=MatrixUtil.toOneTwo(new Matrix(mms),n,n); </pre>	<p>셔플 <math>\rightarrow</math> java.util.Collections.shuffle() 1차원배열 <math>\rightarrow</math> 2차원배열 2차원배열 <math>\rightarrow</math> 1차원배열</p>
136	<pre> Matrix s0=new Matrix(basic); Matrix s1=MatrixUtil.mulMatrix( new Matrix(x2), s0); // 자바에서 연산자의 오버로딩은 없다. static 메서드를 사용 </pre>	<p>수도쿠 static 메서드를 사용, 깊은 복사, 백트래킹으로도 구할</p>

		수 있다.
137	<pre> if(i!=location &amp;&amp; j!=location &amp;&amp; sh[i]&gt;sh[j] ){     ivn++; } // SongPuzzleMain은 2~5 게임 가능 // kr.co.infopub.chapter.s137.ref.TilePanelMain을 실행시켜 참고 // s137은 게임을 만드는 것이고 컴퓨터가 그림을 맞추게 하려면 // 깊이우선, 넓이우선, 맨하탄외에도 많은 방법이 있다. </pre>	뒤집힌 수(큰 수가 앞쪽에 있는 경우 수)를 구한다.
138	<pre> 투수, 타자, 심판등 기능에 따라 객체를 나눈다. // kr.co.infopub.chapter.s138.ref.BaseBallJFrameMain로 참고 </pre>	여러 객체로 나누기, 불과 스트라이크 로직
139	<pre> public static boolean isStraightFlush(Card[] cd){     boolean isSF=false;     if(isAllStraight(cd) &amp;&amp; isAllFlush(cd)){         isSF=true;     }     return isSF; } </pre>	5카드 족보 스트레이트플러쉬= 스트레이트 && 플러쉬
140	<pre> int tot=100000; tot+=CardUtil.toVal(ts.charAt(0))*200; // H1H2H3H4H5 → 기본 100000 + 1*200 // H2H3H4H5H6 → 기본 100000 + 2*200 </pre>	족보마다 기본값 + 세부 차이값

챕터	일부 소스	내용
141	<pre> MyRun mr1=new MyRun(); Thread t1=new Thread(mr1); MyThread t2=new MyThread(); new Thread( ()-&gt; {     for(int i=0;i&lt;500;i++){System.out.print("U");} }).start(); t1.start(); t2.start(); </pre>	쓰레드-동시작업 흉내 쓰레드 구현 2가지 implements Runnable, extends Thread  구현할 메서드가 1개뿐인 인터페이스 → 람다표현
142	<pre> MyRuns mr1=new MyRuns(); Thread t1=new Thread(mr1,"a"); Thread t2=new Thread(mr1,"b"); </pre>	여러 개의 쓰레드가 1개의 자원을 공유
143	<pre> // 멤버가 없는 공유 - A모두, B모두, C모두 찍기 public synchronized void show(){     for(int i=0;i&lt;500;i++){ // 멤버 공유 - A만 찍고 끝 private int i=0; //멤버 자원공유 public synchronized void show(){     for( ;i&lt;500;i++){ </pre>	여러 개의 쓰레드가 1개의 자원을 공유할 때 동기화 (synchronized), 멤버를 공 유할 때와 하지 않을 때 차 이점
144	<pre> public synchronized void makeBread(){     if(breadCount&gt;=10){         try{             System.out.println("빵이 남는다.");             wait();         }catch(InterruptedException ire){}     }     breadCount++; // 빵이 10개가 안되면 더 만들자.     System.out.println("1개 더 만들: "+breadCount+"개");     this.notifyAll(); } </pre>	여러 개의 쓰레드가 1개의 자원을 공유할 때 동기화 (synchronized), 공유접시 에 10개를 모두 채웠을 때 와, 빵이 없을 때 락이 걸 릴 수 있다. 이때 wait()와 notifyAll()로 대화를 통해 락없이 먹고 만들

145	<pre> FileOutputStream fos=new FileOutputStream(fname); ObjectOutputStream oos=new ObjectOutputStream(fos); oos.writeObject(v.get(i));  FileInputStream fis = new FileInputStream(fname); ObjectInputStream ois=new ObjectInputStream(fis); System.out.println(ois.readObject()); </pre>	객체 직렬화 (java.io.Serializable), 객체를 파일로 저장, 파일에서 객체 읽기
146	<pre> for (String pstr : proarr) {     Product p=plist.splistP(pstr);     System.out.println(p); } </pre>	String을 객체로 변환시키기
147	<pre> public class ProductComp implements Comparator&lt;Product&gt;{     @Override     public int compare(Product sg1, Product sg2) {         int pMoney1=sg1.getPMoney(); </pre>	객체는 대소를 판별 할 수 없다 → 정렬하기 위해 대소 기준 객체 만들기 (Comparator 구현)
148	<pre> IntStream.range(0, gradepros.size()).forEach(     i -&gt;{ System.out.printf("%d%s\n", (i+1),         gradepros.get(i).toString());}); gradepros.sort(Product::compareProduct); // 람다 gradepros.sort((c1, c2) -&gt;{return c1.compareTo(c2);}); </pre>	List Stream, range(), :: → static 메서드로 정렬 람다표현으로 정렬(s1482)
149	<pre> Calendar bf15day=RestDay.dateAfOrBf(todayCal,i); // 살아온날 double value=BioCalendar.getPhysical(live+i,100); // 신체지수 값 // 날짜에 대하여 신체지수값을 XY로 점을 찍고 연결 series.getData().add(new XYChart.Data&lt;String, Number&gt;(     RestDay.toStrDate(bf15day), value)); </pre>	javaFX, RestDay 날짜에 관련된 각종 메서드, 날짜 변환 메서드
150	<pre> public void emotionalChartData(XYChart.Series&lt;String, Number&gt; series, long live ) {} public void intellectualChartData(XYChart.Series&lt;String, Number&gt; series, long live ) {} public void compositeChartData(XYChart.Series&lt;String, Number&gt; series, long live ) {} </pre>	s149, 신체, 감정, 지능 등 각 메서드로 만들, javaFX, RestDay 날짜에 관련된 각종 메서드, 날짜 변환 메서드
151	<pre> double result=krw/USD; // 연산 1000000.00/1113.00 </pre>	환율을 알고 환전
152	<pre> public static double calculate(double krwMoney, double exchangeRatio){     return krwMoney/exchangeRatio; } </pre>	메서드로 만들
153	<pre> if(choice==1){     result=calculate(krw,USD); } </pre>	if를 이용하여 원하는 나라의 화폐 USD, JPY, CNY로 환전
154	<pre> switch (choice) {     case 1:         result=calculate(krw,USD); } </pre>	switch를 이용하여 원하는 나라의 화폐 USD, JPY, CNY로 환전
155	<pre> FixerRate finance =new FixerRate(); finance.base="USD"; finance.jpy=110.22; finance.cny=6.7978; </pre>	환율정보를 객체에 저장
156	<pre> http://api.fixer.io/2017-10-22?base=USD {"base":"USD","date":"2017-10-20","rates":{"AUD":1.2739,"BGN":1.6549,"BRL":3.1747,"CAD":1.2494,"CHF":0.9813,"CNY":6.6185,"CZK":21.738,"DKK":6.2987,"GBP":0.75836,"HKD":7.803,"HRK":6.3518,"HUF":260.73,"IDR </pre>	java.io.BufferedReader를 이용하여 문자열을 읽은 후 JSON 객체로 변환, 원하는 정보를 얻는다.
157	<pre> JSONObject jObject = new JSONObject(data); </pre>	JSON 객체로 변환, 원하는

	JSONObject rates=JSONObject.getJSONObject("rates"); FixerRate fixerrate=new FixerRate(); fixerrate.date=JSONObject.getString("date"); fixerrate.base=JSONObject.getString("base");	정보를 얻고 객체에 저장
158	FixerRate finance=FixerConvert.convert(date, you);	환율정보를 저장한 객체를 얻어 환전
159	환율과 환전화면을 javafx로 만든다.	ScenBuilder 사용
160	현재 환율을 인터넷에스 스크래핑, 환율을 얻는다.	s157, s158
161	String todays=RestDay.toWantedDay(date,i); if(! RestDay.isRest(todays)){ FixerRate fixerrate2=FixerConvert.convert(todays,base);	휴일이 아니면 날짜별 환율을 얻는다.
162	날짜별 1년치 환율변화를 그래프로 그린다	javaFX LineChart
163	String [] names={"Cm","M","Inch","Feet","Yard","Mile"};	String 배열
164	double []distance={1,0.01,0.3937007874015748,0.03280839895013123, 0.010936132983377079,0.00000621371192237334}; .. System.out.printf("%19.12f\t",distance[j]/distance[i]);	하나의 단위로 다른 단위를 나누면 비율을 알수 있다. double 배열
165	public static double [][] convert(){ mm[i][j]=distance[j]/distance[i];	메서드로 단위별 비율을 구한다.
166	ObservableList<String> options = FXCollections.observableArrayList(); options.addAll( Conversion.names ); cbdistance.setItems(options);	비어있는 ObservableList<String> 에 문자열 배열을 넣어 ObservableList<String>로 만든다.
167	public static double toCelsius(double fahrenheit){ return 5./9 *(fahrenheit-32); }	화씨를 섭씨로 변환하기 메서드를 만든다
168	fdegree.valueProperty().addListener(new ChangeListener<Number>() { public void changed(ObservableValue<? extends Number> ov, Number old_val, Number new_val) { double fval=new_val.doubleValue(); double cval=TemperatureConv.toCelsius(fval);	섭씨 Slider에 이벤트를 등록하여 변동된 섭씨를 얻어 화씨로 변환하고 화씨 Slider에 반영한다.
169	Haversine	java.lang.Math의 삼각함수를 이용
170	위도, 경도를 객체에 저장, Haversine	java.lang.Math의 삼각함수를 이용
171	double distance=HaversineDistance.distance( latlng[i][0], latlng[i][1], latlng[i+1][0], latlng[i+1][1]); tot+=distance;	배열을 이용, 지점사이의 거리를 더하여 총거리를 구한다
172	BufferedReader br=new BufferedReader(new FileReader("distance.txt"));	CSV 파일을 읽어 파싱, 배열로 만든다.
173	ArrayList<Geo> geos=hfyw.getGeos(); double tot=0.0; //산책한 총 거리 for (int i = 0; i < geos.size()-1; i++) { double distance=HaversineDistance.distance( geos.get(i), geos.get(i+1)); tot+=distance;	List에 위도와 경도를 저장한 객체를 저장하고, 저장된 객체에서 위도와 경도를 얻고 거리들의 총합을 구한다.
174	BufferedReader br= new BufferedReader( new FileReader("loc.json")); JSONObject jobject = new JSONObject(json); JSONArray loc = jobject.getJSONArray("loc");	문자열을 붙여서 JSON 객체로 변환 후 위도와 경도를 얻고 거리를 구한다.
175	try (BufferedReader reader =	좀더 편한 SimpleJSON을

	<pre> new BufferedReader(new FileReader("loc.json"))){     Object obj = JSONValue.parse(reader);     JSONObject = (JSONObject) obj; </pre>	이용, Reader를 이용 JSON 객체로 변환 후 위도와 경도를 얻고 거리를 구한다.
176	javaFX WebView에서 JS로 JAVA 메서드를 호출할 수 있다.	WebView
177	javaFX WebView에서 JS로 JAVA 메서드를 호출할 수 있다.	WebView
178	<pre> BufferedReader reader = new BufferedReader( new InputStreamReader(inputStream, "euc-kr"), 8); </pre>	지진정보를 스크래핑, 원하는 정보를 얻는다.
179	KMA에서 얻은 정보를 JSON으로 변환 후 JS를 이용 WebView에 반영	웹뷰를 이용하면 Java <-> JS 상호 호출이 가능하다.
180	<p>org.geo에서 제공하는 JSON을 파싱하여 List&lt;Earthquake&gt;로 제공하는 EarthRequest, JSON 형식의 문자열을 제공받아 지진의 정보를 웹뷰에 보이는 HowFarDistanceFx2는 이미 만들어져 있는 소스. 두 소소를 연결하여 geo 지진정보를 웹뷰에 보임</p> <pre> // http://api.geonames.org/earthquakesJSON { "earthquakes": [     { "datetime": "2017-03-01 07:08:30", "depth": 130.99,       "lng": 130.0288,"src": "us","eqid": "us100085ws",       "magnitude": 5.5, "lat": -6.325    },     → BurfferedReader → String → JSONObject → List&lt;Earthquake&gt;     → String ss=ss.concat("{\"loc\": [");     → {"loc": [       {"marknum":"7.0","markerX":"14.9","markerY":"-92.0"},       {"marknum":"6.6","markerX":"-1.27","markerY":"120.48"},     → JSONObject → WebView </pre>	java.io.BurfferedReader, JSONObject , List<Earthquake>,' WebView

챕터	일부 소스	내용
181	HRM을 위한 화면 준비, log	java.util.Date, java.util.Calendar java.sql.Sql 데이트 변환
182	HRM 기본 프레임	SceneBuilder Alert, ActionEvent, javaFX @XML 이벤트처리
183	<p>HRM DB관련 인사관리를 위한 부서용 DAO 만들기</p> <pre> public class EmployeeDAO extends DataBase{     public int getEmployeesTotal () throws SQLException {}     public List&lt;DepartmentDto&gt; findAllDepartments () throws SQLException {}     public List&lt;DepartmentDto&gt; findAllDepartments2 () throws SQLException {}     public List&lt;DepCountDto&gt; findAllDepCounts () throws SQLException {} } </pre>	DAO, 상속, JDBC java.sql.Connection java.sql.Statement java.sql.ResultSet 예외처리 java.sql.SQLException
184	<p>HRM DB관련 인사관리를 위한 사원관리용 DAO 만들기</p> <pre> public class EmployeeDAO extends DataBase{     List&lt;EmployeeDto&gt; findAllEmployees() throws SQLException {}     List&lt;EmployeeDto&gt; findTreeManagerInEmployee() throws </pre>	DAO, 상속, JDBC java.sql.Connection java.sql.Statement java.sql.ResultSet 예외처리

	<pre> SQLException {}     int getTreeMaxLevel() throws SQLException {}     List&lt;EmployeeDto&gt; findEmployeesByManagerId(empid) throws SQLException {}     List&lt;String&gt; findAllJobs() throws SQLException {} } </pre>	java.sql.SQLException
185	<p>HRM 화면 처리용(ObservableList 테이블뷰, 트리뷰에 적합한 프로퍼티 리스트), 부서 트리뷰</p> <pre> List&lt;DepartmentDto&gt; blist = employeeDAO.findAllDepartments (); // JavaFX TreeItem을 위한 리스트로 변환 List&lt;Department&gt; dlists=DepConvert.toObservProFromDto(blist); // 부서 아이템 TreeItem&lt;String&gt; root = new TreeItem&lt;String&gt;("부서"); // 부서 아이템에 부서들 붙이기 makeDepTree(root,dlists); // 트리뷰에 트리아이템 붙이기 TreeView&lt;String&gt; treeView = new TreeView&lt;String&gt;(root); </pre>	<p>javaFx 트리뷰, 테이블뷰용 프로퍼티(get/set), List → ObservableList → TreeItem → TreeView 부서별 정보 s183, s814 DAO 사용</p>
186	<p>HRM 화면 처리용(ObservableList 테이블뷰, 트리뷰에 적합한 프로퍼티 리스트), 관리자/사원 계층 트리뷰</p> <pre> // 가장 큰 레벨을 얻는다. int max = employeeDAO.getTreeMaxLevel(); // 관리자/부서원을 트리관계를 얻는다. 계층구조다. List&lt;EmployeeDto&gt; blists = employeeDAO.findTreeManagerInEmployeee(); // JavaFX 트리뷰용 리스트로 변환한다. List&lt;Employee&gt; dlists = EmpConvert.toObservProFromDto(blists); // "매니저와 직원" 트리아이템을 만든다. TreeItem&lt;String&gt; root = new TreeItem&lt;String&gt;("매니저와 직원"); // "매니저와 직원" 트리아이템에 계층 구조 트리 아이템을 붙인다. makeEmpTree(root,dlists,"",1,max); // 트리뷰에 "매니저와 직원" 트리아이템을 붙인다. TreeView&lt;String&gt; treeView = new TreeView&lt;String&gt;(root); </pre>	<p>javaFx 트리뷰, 테이블뷰용 프로퍼티(get/set), List → ObservableList → TreeItem → TreeView 관리자/사원 정보, 계층 트리(재귀) s183, s814 DAO 사용</p>
187	<p>HRM 화면 처리용(ObservableList 테이블뷰, 트리뷰에 적합한 프로퍼티 리스트), 사원 테이블뷰</p> <pre> @FXML private TableView&lt;Employee&gt; employeeTable;  // 모든 사원 가져오기 List&lt;EmployeeDto&gt; emplists=employeeDAO.findAllEmployees(); // JavaFX 테이블뷰용 리스트로 변환 ObservableList&lt;Employee&gt;     empData = EmpConvert.toObservProFromDto(emplists); // 테이블에 보여주기 employeeTable.setItems(empData); </pre>	<p>javaFx 트리뷰, 테이블뷰용 프로퍼티(get/set) 테이블뷰 List → ObservableList → TableView s183, s814 DAO 사용</p>
188	<p>HRM 화면 처리용(ObservableList 테이블뷰, 트리뷰에 적합한 프로퍼티 리스트), 파이차트에 파이(부서별 사원인원)</p>	<p>javaFx 트리뷰, 테이블뷰용 프로퍼티(get/set) 파이차트</p>



	<pre>List&lt;DepCountDto&gt; bdlists=employeeDAO.findAllDepCounts(); List&lt;DepCount&gt; dlists =DepCountConvert.toObservProFromDto(bdlists);  PieChart pieChart = new PieChart(); // 부서별 부서원 수로 파이 넓이를 차지한다. pieChart.setData(getChartData(dlists));</pre>	<p>List → ObservableList → 파이(부서별 인원) s183, s814 DAO 사용</p>
189	HRM 화면(부서별 트리뷰 + 부서별 사원정보 테이블뷰)	<p>HRM 기본 프레임 s182 프레임, s181 log, s183, s814 DAO 사용 → s186완성</p>
190	HRM 화면(관리자/사원 계층 트리뷰 + 관리자/사원 사원정보 테이블뷰)	<p>HRM 기본 프레임 s182 프레임, s181 log, s183, s814 DAO 사용 → s187완성</p>
191	<p>HRM 2tier 어플리케이션</p> <p>HRM 화면(부서별 인원 파이차트+ 부서정보) View : JavaFx 테이블뷰+ 트리뷰+ 파이차트 Model : Service(Convert+DAO) 싱글톤 Controller : View+Model 흐름제어</p>	<p>HRM 기본 프레임 s182 프레임, s181 log, s183, s814 DAO 사용 → s188, 189, 190완성 DAO+Convert(ObserableList -&gt; List) → Service로 통합</p>
192	<p>HRM Restful 웹 서비스 어플리케이션</p> <p>Spring Boot, Restful, Maven, Mybatis, DAO, interface, JUnit Maven 프로젝트 → POM.xml → 환경변수(application.properties, application.yml, logback.xml, mybatis-config.xml) → DAO → DAO Test → Service → Service Test</p>	<p>HRM Restful 웹 서비스 서버 준비 1 hrm192server project</p>
193	<p>Service Test → Controller → Swagger config → Controller Test → API 확인(서버쪽 Restful 웹 서비스 API 테스트) → Post Man(클라이언트쪽 Restful 웹 서비스 API 테스트) → HRM Restful 웹 서비스 서버 실행</p>	<p>HRM Restful 웹 서비스 서버 준비 2 hrm192server project</p>
194	<p>Swagger로 만든 API 확인 및 테스트</p> <p>RestTemplate을 이용하여 Restful 웹서비스의 이름을 호출, JSON 데이터를 객체로 변환 → List&lt;DTO&gt; (전송용) → ObservableList&lt;Property&gt;로 변환(화면용)</p>	<p>HRM Restful 웹 서비스 어플리케이션 만들기 1 hrmclient project JsonDaoTest~JsonDaoTest16 Restful 웹 서비스 테스트</p>
195	<p>s194에서 ObservableList&lt;Property&gt;로 변환시킨 후 부서별 트리뷰, 관리자/사원 계층 트리뷰, 사원 정보 테이블 뷰, 부서별 사원정보 파이차트를 화면으로 테스트</p>	<p>HRM Restful 웹 서비스 어플리케이션 만들기 2 hrmclient project javaFX</p>
196	HRM Restful 웹 서비스 개시	HRM Restful 웹 서버 실행
197	HRM Restful 웹 서비스를 사용하여 HRM 어플리케이션 만들기	HRM Restful 어플리케이션
198	<p>ServerSocket, Socket이용하여 TCP/IP 채팅 프로그래밍</p> <p>서버는 ServerSocket을 실행 클라이언트의 Socket 접속을 기다림 클라이언트가 서버소켓에 접속하는 순간 Socket 생성 생성된 소켓을 Thread에 대입, 여러 쓰레드가 실행되도록 함 동기화된 리스트와 동기화된 브로드캐스팅을 이용하여 한곳에서 온 메시지를 모든 쓰레드에 보냄 클라이언트는 소켓을 이용하여 서버에 접속, IO를 이용하여 소켓에서</p>	<p>java.net.ServerSocket java.net.Socket java.lang.Thread synchronized synchronized List</p>

	메시지를 보내고 받음 , 동시에 쓰고/받기를 위해 쓰레드	
199	<p>서버와 클라이언트와의 대화를 위해 약속 프로토콜을 정수로 약속함</p> <p>메서드별로 프로토콜 번호를 붙이고 Request 객체를 통해 요청을 서버로 보냄</p> <p>서버는 요청(Request) 객체의 프로토콜을 받고 요청한 내용을 DB에서 가져와 Response 객체에 담아서 클라이언트로 되돌려 보냄</p> <p>서버는 Request로 전달된 프로토콜 번호를 전담해서 요청을 처리하는 핸들러 쓰레드가 필요</p>	<p>java.net.ServerSocket</p> <p>java.net.Socket</p> <p>java.lang.Thread</p> <p>synchronized</p> <p>synchronized List</p> <p>객체 직렬화/역 직렬화</p>
200	클라이언트는 Response에서 결과를 얻고 화면에 반영함	<p>java.net.ServerSocket</p> <p>java.net.Socket</p> <p>java.lang.Thread</p> <p>synchronized</p> <p>synchronized List</p> <p>객체 직렬화/역 직렬화</p> <p>JavaFX</p>