12. 간단한 콘서트 예약 시스템을 만들어보자. 다수의 클래스를 다루고 객체의 배열을 다루기에는 아직 자바 프로그램 개발이 익숙하지 않은 초보자에게 다소 무리가 있을 것이다. 그러나 반드시 넘어야 할 산이다. 이 도전을 통해 산을 넘어갈 수 있는 체력을 키워보자. 예약 시스템의 기능은 다음과 같다. 난이도 9

- 공연은 하루에 한 번 있다.
- 좌석은 S석, A석, B석으로 나뉘며, 각각 10개의 좌석이 있다.
- 예약 시스템의 메뉴는 "예약", "조회", "취소", "끝내기"가 있다.
- 예약은 한 자리만 가능하고, 좌석 타입, 예약자 이름, 좌석 번호를 순서대로 입력받아 예약한다.
- 조회는 모든 좌석을 출력한다.
- 취소는 예약자의 이름을 입력받아 취소한다.
- 없는 이름, 없는 번호, 없는 메뉴, 잘못된 취소 등에 대해서 오류 메시지를 출력하고 사용자가 다시 시도하도록 한다.

핵심 여러 개의 클래스와 여러 개의 객체를 다루는 종합 응용

```
명품콘서트홀 예약 시스템입니다.
예약:1, 조회:2, 취소:3, 끝내기:4>>1
좌석구분 S(1), A(2), B(3)>>1
S>> --- --- --- --- --- --- --- --- --- ---        ← 현재 S석 상태
이름>>황기태
번호>>1
예약:1, 조회:2, 취소:3, 끝내기:4>>1
좌석구분 S(1), A(2), B(3)>>2
A>> --- --- --- --- --- --- --- --- --- ---
이름>>김효수
번호>>5
예약:1, 조회:2, 취소:3, 끝내기:4>>2
S>> 황기태 --- --- --- --- --- --- --- --- ---
A>> --- --- --- --- 김효수 --- --- --- --- ---     ← 예약된 모든 좌석 조회
B>> --- --- --- --- --- --- --- --- --- ---
<<<조회를 완료하였습니다.>>>
예약:1, 조회:2, 취소:3, 끝내기:4>>3
좌석 S:1, A:2, B:3>>2
A>> --- --- --- --- 김효수 --- --- --- --- ---
이름>>김효수
예약:1, 조회:2, 취소:3, 끝내기:4>>2
S>> 황기태 --- --- --- --- --- --- --- --- ---
A>> --- --- --- --- --- --- --- --- --- ---        ← 김효수가 삭제된 좌석 상황
B>> --- --- --- --- --- --- --- --- --- ---
<<<조회를 완료하였습니다.>>>
예약:1, 조회:2, 취소:3, 끝내기:4>>4
```

9. 다음 Stack 인터페이스를 상속받아 실수를 저장하는 StringStack 클래스를 구현하라.

핵심 인터페이스에 대한 이해 및 클래스 구현 활용

```
interface Stack {
  int length(); // 현재 스택에 저장된 개수 리턴
  int capacity(); // 스택의 전체 저장 가능한 개수 리턴
  String pop(); // 스택의 톱(top)에 실수 저장
  boolean push(String val); // 스택의 톱(top)에 저장된 실수 리턴
}
```

그리고 다음 실행 사례와 같이 작동하도록 StackApp 클래스에 main() 메소드를 작성하라. 난이도 6

```
총 스택 저장 공간의 크기 입력 >> 3
문자열 입력 >> hello
문자열 입력 >> sunny
문자열 입력 >> smile
문자열 입력 >> happy
스택이 꽉 차서 푸시 불가!
문자열 입력 >> 그만
스택에 저장된 모든 문자열 팝 : smile sunny hello
```

"그만"을 입력하면 프로그램 종료

10. 다음은 키와 값을 하나의 아이템으로 저장하고 검색 수정이 가능한 추상 클래스가 있다.

핵심 추상 클래스의 구현과 활용 연습

```
abstract class PairMap {
  protected String keyArray []; // key 들을 저장하는 배열
  protected String valueArray []; // value 들을 저장하는 배열
  abstract String get(String key); // key 값을 가진 value 리턴. 없으면 null 리턴
  abstract void put(String key, String value); // key와 value를 쌍으로 저장. 기존에
                                               key가 있으면, 값을 value로 수정
  abstract String delete(String key); // key 값을 가진 아이템(value와 함께) 삭제.
                                      삭제된 value 값 리턴
  abstract int length(); // 현재 저장된 아이템의 개수 리턴
}
```

PairMap을 상속받는 Dictionary 클래스를 구현하고, 이를 다음과 같이 활용하는 main() 메소드를 가진 클래스 DictionaryApp도 작성하라. 난이도 7

```
public static void main(String[] args) {
    Dictionary dic = new Dictionary(10);
    dic.put("황기태", "자바");
    dic.put("이재문", "파이선");
    dic.put("이재문", "C++"); // 이재문의 값을 C++로 수정
    System.out.println("이재문의 값은 " + dic.get("이재문"));
    System.out.println("황기태의 값은 " + dic.get("황기태"));
    dic.delete("황기태"); // 황기태 아이템 삭제
    System.out.println("황기태의 값은 " + dic.get("황기태")); // 삭제된 아이템 접근
}
```

이재문 아이템 출력

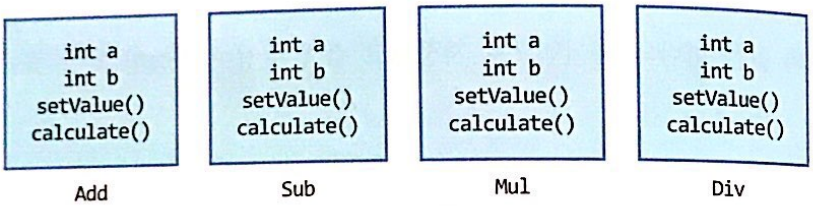
황기태 아이템 출력

```
이재문의 값은 C++
황기태의 값은 자바
황기태의 값은 null
```

●● 추상 클래스, 오버라이딩, 동적바인딩

11. 철수 학생은 다음 3개의 필드와 메소드를 가진 4개의 클래스 Add, Sub, Mul, Div를 작성하려고 한다(4장 실습문제 11 참고).

- int 타입의 a, b 필드: 2개의 피연산자
- void setValue(int a, int b): 피연산자 값을 객체 내에 저장한다.
- int calculate(): 클래스의 목적에 맞는 연산을 실행하고 결과를 리턴한다.



곰곰 생각해보니, Add, Sub, Mul, Div 클래스에 공통된 필드와 메소드가 존재하므로 새로운 추상 클래스 Calc를 작성하고 Calc를 상속받아 만들면 되겠다고 생각했다. 그리고 main() 메소드에서 다음 실행 사례와 같이 2개의 정수와 연산자를 입력받은 후, Add, Sub, Mul, Div 중에서 이 연산을 처리할 수 있는 객체를 생성하고 setValue()와 calculate()를 호출하여 그 결과 값을 화면에 출력하면 된다고 생각하였다. 철수처럼 프로그램을 작성하라. 난이도 5

```
두 정수와 연산자를 입력하시오>>5 7 +
12
```

12. 텍스트로 입출력하는 간단한 그래픽 편집기를 만들어보자. 본문 5.6절과 5.7절에서 사례로 든 추상 클래스 Shape과 Line, Rect, Circle 클래스 코드를 잘 완성하고 이를 활용하여 아래 시행 예시처럼 "삽입", "삭제", "모두 보기", "종료"의 4가지 그래픽 편집 기능을 가진 클래스 GraphicEditor을 작성하라. 난이도 7

상속을 이용한 응용 만들기

```
그래픽 에디터 beauty을 실행합니다.
삽입(1), 삭제(2), 모두 보기(3), 종료(4)>>1
Line(1), Rect(2), Circle(3)>>2
삽입(1), 삭제(2), 모두 보기(3), 종료(4)>>1
Line(1), Rect(2), Circle(3)>>3
삽입(1), 삭제(2), 모두 보기(3), 종료(4)>>3
Rect
Circle
삽입(1), 삭제(2), 모두 보기(3), 종료(4)>>2
삭제할 도형의 위치>>3
삭제할 수 없습니다.
삽입(1), 삭제(2), 모두 보기(3), 종료(4)>>4
beauty을 종료합니다.
```

힌트

Shape을 추상 클래스로 작성한 사례는 다음과 같다.

```
public abstract class Shape {
  private Shape next;
  public Shape() { next = null; }
  public void setNext(Shape obj) { next = obj; } // 링크 연결
  public Shape getNext() { return next; }
  public abstract void draw(); // 추상 메소드
}
```

13. 다음은 도형의 구성을 묘사하는 인터페이스이다. 난이도 6

인터페이스에 대한 이해와 클래스 구현

```
interface Shape {
  final double PI = 3.14;          // 상수
  void draw();                     // 도형을 그리는 추상 메소드
  double getArea();                // 도형의 면적을 리턴하는 추상 메소드
  default public void redraw() {   // 디폴트 메소드
    System.out.print("--- 다시 그립니다. ");
    draw();
  }
}
```