Practice 7

실습 시 유의 사항

- 실습 과제 제출 시 패키지 형태로 압축된 소스코드와 보고서를 함께 이클레스에 제출하셔야 합니다.
- 각 문제마다 필수 적인 요소들은 빨간색으로 강조되어 있습니다. 참조하시면 되겠습니다.
- 이번 실습 과제는 중간고사 관계로 5월 4일 오후 9시 까지 이클레스에 제출하시면 됩니다.
- 연장 제출은 제출 기한 종료 이후 7일 이내로만 받습니다. 필요시 강의 계획서의 조교 메일로 보내시면 됩니다.

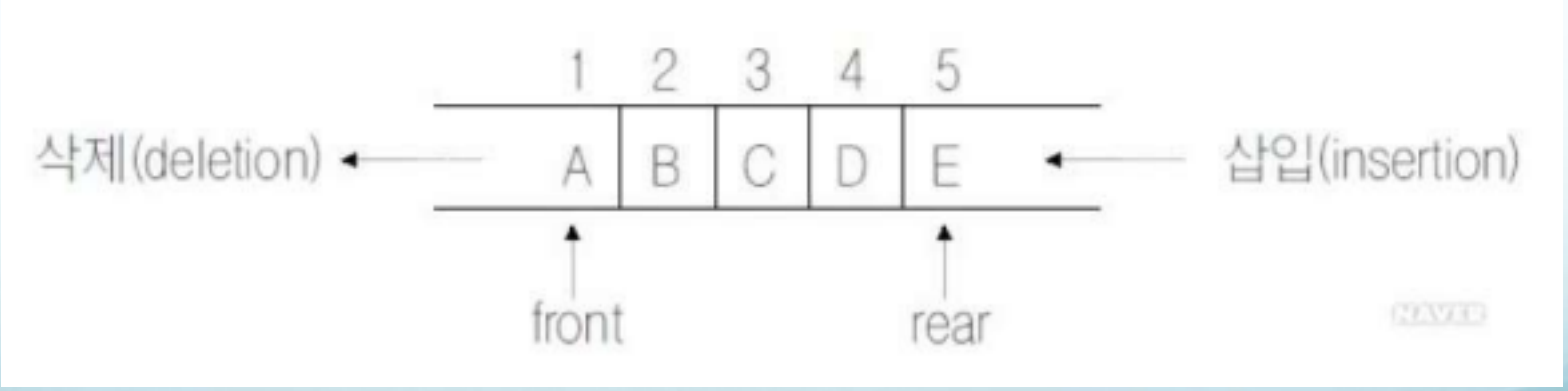
문제 1

- 다음 코드에서 오류를 찾고 이를 수정하라. (보고서 기입)

```
public class Bike
{       protected
        private int gear;
        public int speed;
}
                                extends
public class MountainBike expands Bike
{
        public int seatHeight;
        public void MountainBike(int g)
        {
                super();
                gear=g;
        }
}
```

문제 2

- 먼저 들어온 데이터가 먼저 나가는 자료구조를 큐라고 한다. 다음 인터페이스와 클래스를 이용하고 주어진 코드의 빈칸을 완성해 큐 클래스에서 삽입, 삭제, 검색, 큐 원소 개수 반환이 가능하도록 클래스를 구현하고 주어진 테스트 클래스를 이용하여 결과를 출력하시오.



문제 2

- 인터페이스 Collection

```
interface Collection
{
        void add(Object obj);          // 큐에 obj 데이터 추가

        void delete();                 // 큐에서 데이터 삭제

        void find(Object obj);         // 큐에서 obj 데이터가 있는지 찾기

        int currentCount();            // 큐에서 현재 몇 개의 데이터가 있는지 찾기
}
```

문제 2

- 클래스 Link

```
class Link
{
        Object data;            // 실제 데이터
        Link next;              // 다음 데이터를 읽기 위한 연결고리


        Link(Object d, Link n)
        {
                data = d;       // 실제 데이터 값 설정
                next = n;       // 연결 고리 설정
        }
}
```

문제 2

- 빈칸을 채워 큐 클래스의 코드를 완성하시오.

```
class queue implements collection {
    private Link head = null;
    private Link tail = null;
    private Link ptr = null;
    private int count = 0;

    // 빈칸 채워 넣기
    {
    }
```

문제 2

- 빈칸을 채워 큐 클래스의 코드를 완성하시오.

```
class queue implements collection {
    private Link head = null;
    private Link tail = null;
    private Link ptr = null;
    private int count = 0;

    // 빈칸 채워 넣기
    {
    }
```

문제 2

- 테스트 케이스 및 출력 결과

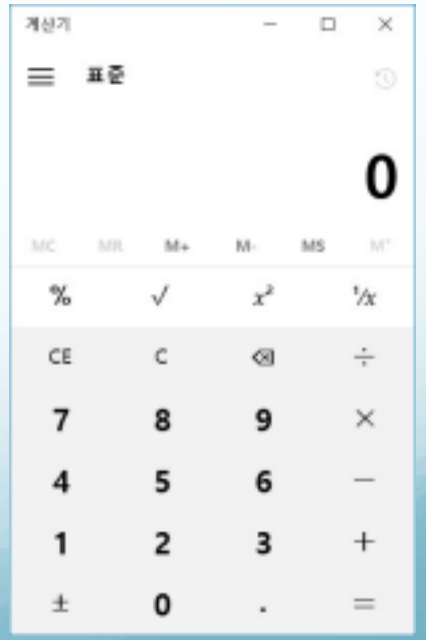
```
class executes
{
    public static void main(String args[])
    {
        System.out.println("Queue simulator start.");

        queue q = new queue();

        Integer a = new Integer(1);

        q.add(new Integer(1));
        q.add(a);
        q.add(new Integer(5));

        q.currentCount();
        q.find(new Integer(5));
        q.find(a);

        q.delete();
        q.delete();
        q.delete();

        q.currentCount();

        System.out.println("Queue simulator end.");
    }
}
```

```
Queue simulator start.
1 is added. code : 366712642
1 is added. code : 1829164700
5 is added. code : 2018699554
3 Objects are found.
There is no 5 in queue.
1 is 2th queue's data.
1 is deleted.
1 is deleted.
5 is deleted.
Queue is empty
Queue simulator end.
```

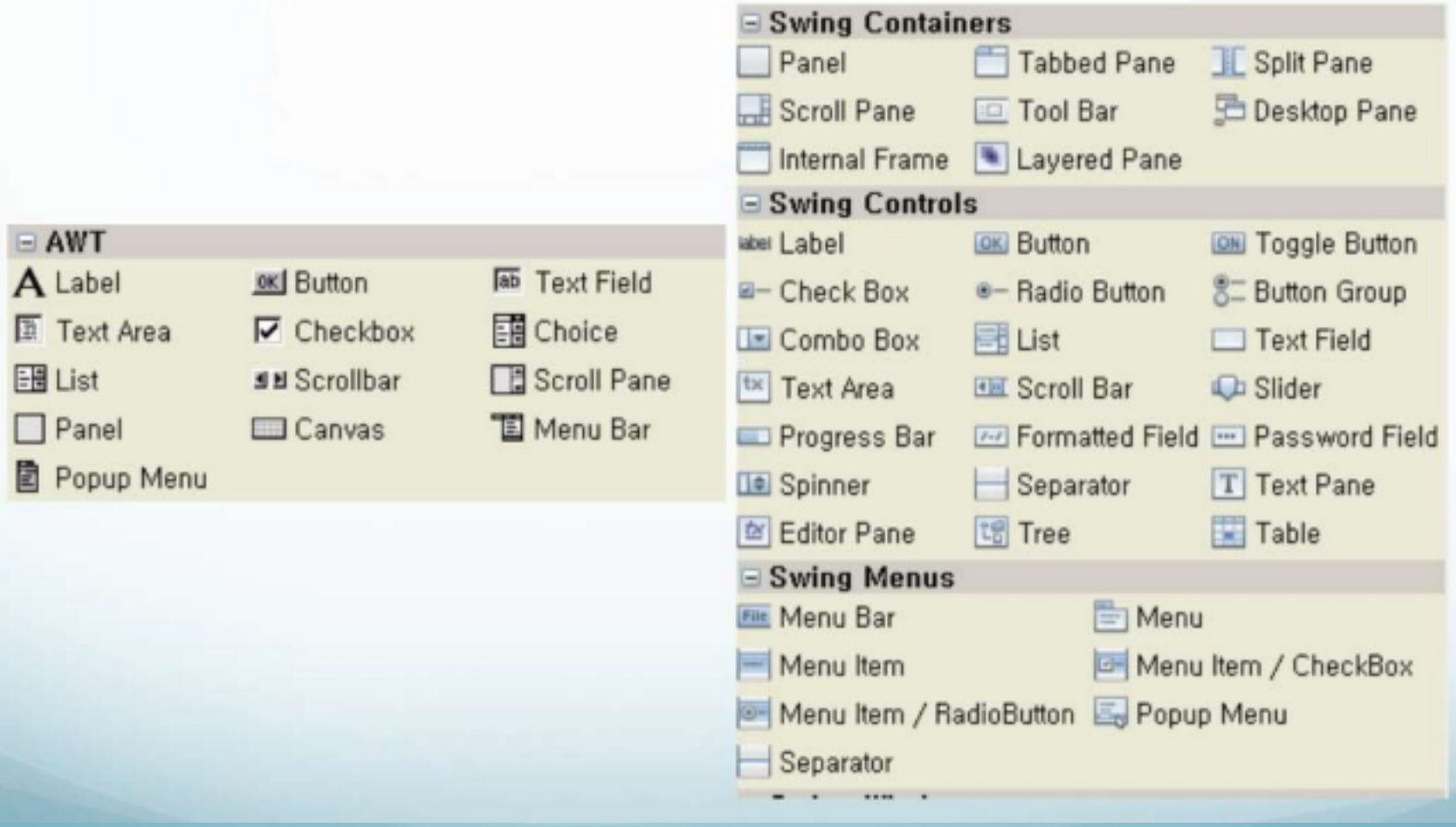
Code는 출력하지 않아도 됨
(Hash code 값)

문제 3

- 간단한 계산기를 작성하여 보자. 계산 기능은 나중에 추가하기로 하자. 여기서는 외관만 구현하면 된다. 계산기의 모양은 각자 컴퓨터의 계산기나 평소 사용하는 계산기 모양을 각자 참고하면 된다. oracle을 참고해서 GUI 구현의 다양한 시도를 해보자.
- 이는 이후 과제에 사용 될 예정입니다.
- 반드시 추가해야 할 버튼 목록
  - 입력 숫자 0 ~ 9
  - 덧셈(+), 뺄셈(-), 나눗셈(/), 곱셈(*), 등호(=)
  - . (소숫점)
  - 계산 결과 출력화면
  - 타이틀에 자신의 학번 넣기

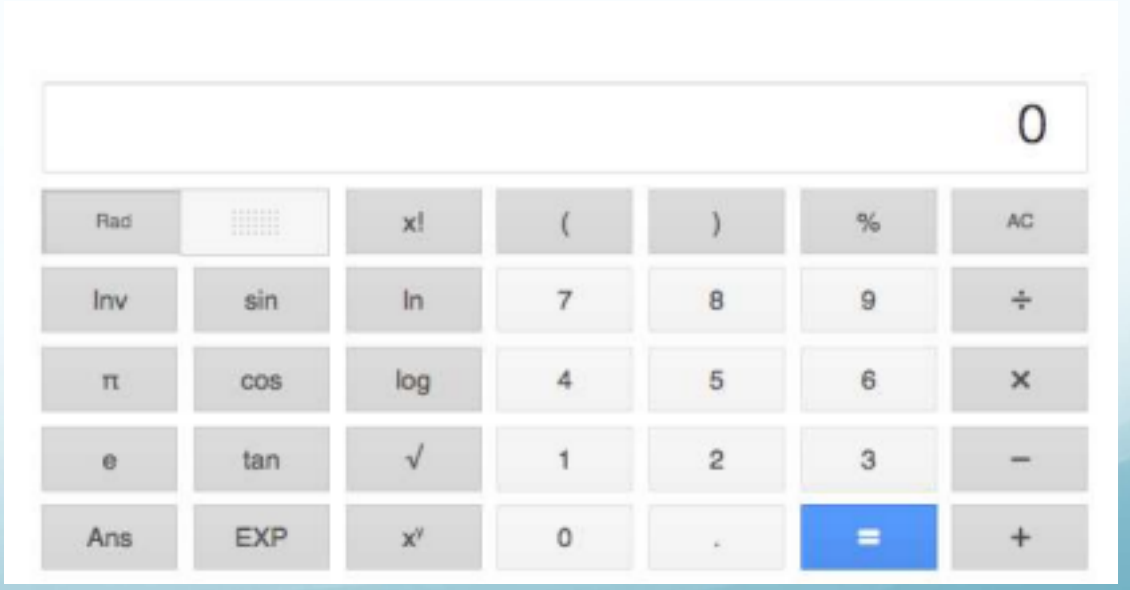


문제 3


AWT
Label
Button
Text Field
Text Area
Checkbox
Choice
List
Scrollbar
Scroll Pane
Panel
Canvas
Menu Bar
Popup Menu
Swing Containers
Panel
Tabbed Pane
Split Pane
Scroll Pane
Tool Bar
Desktop Pane
Internal Frame
Layered Pane
Swing Controls
Label
Button
Toggle Button
Check Box
Radio Button
Button Group
Combo Box
List
Text Field
Text Area
Scroll Bar
Slider
Progress Bar
Formatted Field
Password Field
Spinner
Separator
Text Pane
Editor Pane
Tree
Table
Swing Menus
Menu Bar
Menu
Menu Item
Menu Item / CheckBox
Menu Item / RadioButton
Popup Menu
Separator

문제 3


0
AC
+/-
%
÷
7
8
9
×
4
5
6
−
1
2
3
+
0
.
=


0
Rad
x!
(
)
%
AC
Inv
sin
ln
7
8
9
÷
π
cos
log
4
5
6
×
e
tan
√
1
2
3
−
Ans
EXP
x^y
0
.
=
+

문제 3

- 출력 예시


setTitle을 이용하여 학번 기입!
2345111234, 홍길동's 계산기
calculator
←
CE
C
7
8
9
/
sqrt
4
5
6
x
%
1
2
3
-
1/x
+/-
0
.
+
=