

# 창의적 소프트웨어 프로그래밍

## 실습 과제\_05

## 3-2. 간단한 댓글 관리 프로그램 (Simple Reply Administrator)2

- 2-4에서 만든 댓글 관리 프로그램의 기능이 포함된 ReplyAdmin class 작성
- private 멤버 변수
  - string\* chats : 댓글 목록
- 생성자, 소멸자
  - ReplyAdmin() : chats를 NUM\_OF\_CHAT만큼 초기화
  - ~ReplyAdmin() : chats를 delete

## 3-2. 간단한 댓글 관리 프로그램 (Simple Reply Administrator)2

- 멤버 함수
  - `int getChatCount()` : hw02의 skeleton code 참조
  - `bool addChat(string _chat)` : `_chat`을 `chats`에 추가(추가 실패시 `false` 리턴)
  - `bool removeChat(int _index)` : `_index`에 있는 chat 삭제(`_index`가 없을시 `false` 리턴)
  - `bool removeChat(int *_indices, int _count)` : `_count` 크기의 `_indices` 배열 안에 있는 `index`에 해당되는 chat을 모두 삭제(하나라도 삭제 성공했을 시 `true` 리턴, `index`가 없을 시 무시)
  - `bool removeChat(int _start, int _end)` : `_start`부터 `_end`까지 chat을 모두 삭제 (하나라도 삭제 성공했을 시 `true` 리턴, `start`가 음수거나 `end`가 `chats`보다 클 경우 해당되는 부분만 삭제)
- 자세한 기능은 2-4와 동일하다.

## 3-2. 간단한 댓글 관리 프로그램 (Simple Reply Administrator)2

- 파일명 : reply\_admin  
(reply\_admin.h reply\_admin.cpp reply\_admin\_main.cpp)
- **입력** : 단순 문자열 or 명령어가 포함된 문자열 (#remove, #quit)
- **출력** : 명령어 입력 후 변화된 댓글 목록을 출력

## 3-2. 간단한 댓글 관리 프로그램 (Simple Reply Administrator)2

```
$/reply_admin
Hello, World!
0 Hello, Reply Administrator!
1 I will be a good programmer.
2 This class is awesome.
3 Professor Lim is wise.
4 Two TAs are kind and helpful.
5 I think male TA looks cool.
6 Hello, World!
#remove 5
0 Hello, Reply Administrator!
1 I will be a good programmer.
2 This class is awesome.
3 Professor Lim is wise.
4 Two TAs are kind and helpful.
5 Hello, World!
#remove 0,5
0 I will be a good programmer.
1 This class is awesome.
2 Professor Lim is wise.
3 Two TAs are kind and helpful.
C++ is so hard
0 I will be a good programmer.
1 This class is awesome.
2 Professor Lim is wise.
3 Two TAs are kind and helpful.
4 C++ is so hard
Yesterday was too bad.
0 I will be a good programmer.
1 This class is awesome.
2 Professor Lim is wise.
```

## 3-3. 정수 집합 연산(Integer set operations)

- 입력 받은 정수 집합에 대한 합집합, 교집합, 차집합을 계산하는 SimpleIntSet class 작성
  - 주어진 SimpleIntSet(simple\_int\_set.h) 클래스를 구현하고 이를 이용한다
  - 메인 함수와 멤버 함수의 인터페이스는 변경할 수 없다
  - 집합의 크기는 MAX\_SIZE를 넘지 않는다고 가정한다
  - 주어진 모든 public 멤버 함수를 구현한다
  - 집합 내의 숫자를 오름차순으로 정렬하여 관리한다  
(hw01에서 구현한 정렬 재사용 가능)
  - 한 집합 내에 동일한 원소는 존재할 수 없다  
(e.g. {1 1 2 3} 입력 시 {1 2 3}으로 처리)
  - 입력에 오류가 있으면 종료한다

## 3-3. 정수 집합 연산(Integer set operations)

- 파일명 : simple\_int\_set  
(simple\_int\_set.h simple\_int\_set.cpp simple\_int\_set\_main.cpp)
- **입력** : { num1 num2 ... numk1 } OP { num1 num2 ... numk2 }
- OP = +, \*, -
- **출력** : 연산 결과 집합을 같은 형식으로 출력

```
./simple_int_set
{ 1 2 3 } + { 3 4 5 }
{ 1 2 3 4 5 }
{ -1 5 3 2 } - { 1 2 3 }
{ -1 5 }
{ -1 5 3 2 } * { 1 2 3 }
{ 2 3 }
0
$
```

## 3-4. 이진 탐색(Binary search)

- 앞의 SimpleIntSet 내의 원소를 빠르게 찾을 수 있는 이진 탐색 함수 작성.
  - 이진 탐색 알고리즘은 아래 링크를 참조.
    - [http://en.wikipedia.org/wiki/Binary\\_search\\_algorithm](http://en.wikipedia.org/wiki/Binary_search_algorithm)
  - 주어진 헤더파일에 있는 BinarySearch 클래스를 구현한다.
  - 해당 원소의 위치를 리턴한다. 주어진 값이 없으면 -1 을 리턴한다.
  - 찾을 값이 -999 가 입력되면 종료.



## 3-4. 이진 탐색(Binary search)

- 파일명 : binary\_search  
(binary\_search.h binary\_search.cpp binary\_search\_main.cpp)
- 입력 : SimpleIntSet 입력 이후 탐색하고자 하는 원소 입력
- 출력 : 해당 탐색 값의 위치

```
$/binary_search
{ 2 3 5 7 9 }
2
0
5
2
4
-1
-3
-1
100
-1
9
4
-999
$
```

# 제출 양식 및 제출 기한

- 제출 양식

- 학번\_파일명.cpp파일을 압축해서 학번.zip으로 제출  
(ex. 2018120511.zip)

- 각각 파일명

- 2018120511\_reply\_admin.h
    - 2018120511\_reply\_admin.cpp
    - 2018120511\_reply\_admin\_main.cpp
    - 2018120511\_simple\_int\_set.h
    - 2018120511\_simple\_int\_set.cpp
    - 2018120511\_simple\_int\_set\_main.cpp
    - 2018120511\_binary\_search.h
    - 2018120511\_binary\_search.cpp
    - 2018120511\_binary\_search\_main.cpp

- 제출 기한

- 7/1 23:59까지