HW2 실습

template

- C++ 프로그래밍 언어의 한 기능으로 함수나 클래스가 개별적으로 다시 작성하지 않고도 각기 다른 수많은 자료형에서 동작할 수 있게 한다.

```
int max(int a, int b) {
    return a > b ? a : b;
}

double max(double a, double b) {
    return a > b ? a : b;
}

float ...

long ....
```

```
template <typename Type>
Type max(Type a, Type b) {
    return a > b ? a : b;
}
```

템플릿 함수(1)

- 클래스와 함수의 재사용에 기여

(1) 템플릿 함수를 사용하지 않는 경우

- (예) 정수 배열을 위한 선택 정렬 함수 SelectionSort(프로그램 1.6)
  - 부동 소수점 수의 배열을 정리하려면?
    - 첫째 행에서 int *a를 float *a로 변경
    - 둘째 행에서 “정수”를 “부동 소수점 수”로 변경
    - 11번째 행에서 int를 float로 변경

(2) 템플릿 / 인수화타입(parameterized type)을 이용한 해결

- 소프트웨어 개발 시간과 저장 공간을 상당히 절약

템플릿 함수(2)

- 템플릿 함수 SelectionSort

```
template <class T>
void SelectionSort(T *a, const int n)
{// sort a[0] to a[n-1] into nonincreasing order.
   for ( int i = 0; i < n; i++) {
      int j = i;
      // find smallest integer in a[i] to a[n-1]
      for (int k = i + 1; k < n; k++)
        if (a[k] < a[j]) j = k;
      swap(a[i], a[j]);
   }
}
```

- **템플릿 함수 SelectionSort 이용 예**

```
float farray[100];
int intarray[250];
...
// 이 시점에서 배열들이 초기화된다고 가정한다
SelectionSort(farray, 100);
SelectionSort(intarray, 250);
```

템플릿 함수(3)

- 템플릿 함수 **SelectionSort** 사용시 주의사항
  - <, = 및 복사 생성자
    - int와 float에 대해서는 자동적으로 정의
    - 사용자 정의 데이타 타입에 대해서는 별도로 정의하여야 함
  - (예) SelectionSort를 이용해 Rectangle 배열을 그 넓이 순으로 정렬
    - operator < 정의 필요
    - 지정 연산자, 복사 생성자는 컴파일러의 묵시적 구현으로 충분

- 1-dim array의 크기 변경하는 템플릿 함수 ChangeSize1D

```
template <class T>
void ChangeSize1D(T *& a, const int oldSize, const int newSize)
{
   if (newSize < 0) throw "New length must be >= 0";
   T* temp = new T[newSize];       // new array
   int number = min(oldSize, newSize);
   copy(a, a + number, temp);
   delete [] a;
   a = temp;
}
```

template

- 템플릿 특수화

  특정 타입이 들어왔을 때 다른 방식으로 처리하기 위한 방법

```cpp
#include <iostream>
using namespace std;
template <typename T>
T add(T x, T y) {
    return x + y;
}

template<>
char* add(char* s1, char* s2) {
    char* str = new char[100];
    strcpy(str, s1);
    strcat(str, s2);
    return str;
}

int main()
{
    char num1[] = "10", num2[] = "20";
    cout << add(num1, num2) << endl;

    return 0;
}
```

실습

1. hw2a.cpp 파일을 만든다. (vi hw2a.cpp)
2. 코드를 다음과 같이 작성하고 실행한다.
3. Template을 사용해 3개의 함수를 하나로 줄인다.
4. 실행결과를 확인한다.

```
[B611107@localhost ds_hw1]$ make hw1a
g++     -c -o hw1a.o hw1a.cpp
g++ -o hw1a hw1a.o
[B611107@localhost ds_hw1]$ hw1a
3
7.7
19.31
[B611107@localhost ds_hw1]$
```

```
#include <iostream>
using namespace std;


int add(int x, int y){
        return x + y;
}

double add(double x, double y){
        return x + y;
}

float add(float x, float y){
        return x + y;
}


int main() {
        int intX = 1, intY = 2;
        double dbX = 3.3, dbY = 4.4;
        float fX= 09.24, fY = 10.07;

        cout << add(intX, intY) << endl;
        cout << add(dbX, dbY) << endl;
        cout << add(fX, fY) << endl;


        return 0;
}
```

makefile

- 입력파일 변경 시 결과파일 자동 변경을 원할 때 지능적인 배치작업 수행
- 일일이 gcc 명령어를 안치고도 간단하면서 용이하게 컴파일을 진행할 수 있음

```
hwla:   hwla.o
        g++ -o hwla hwla.o
hwlb:   hwlb.o
        g++ -o hwlb hwlb.o
hwlc:   hwlc.o
        g++ -o hwlc hwlc.o

clean:
        rm -rf *.o
        rm -rf hwla
        rm -rf hwlb
        rm -rf hwlc
```

과제 설명

- 기술서에 적혀 있는 대로 hw2a.cpp, hw2b.cpp, hw2c.cpp 의 비어 있는 부분을 작성해서 제출하면 된다.
- 기술서의 조건 엄수.
- 컴파일 방법: make hw2_ (a, b, c)

Latex 내용

- template에 관한 설명
- 코드에 대한 설명, 풀이과정
- 어려웠던 점
- 등등..

제출 방법

- 1분반: submit pem_ta hw2a // 9월 30일 금요일 24시 까지
- 2분반: submit pem_ta hw2b // 9월 29일 목요일 24시 까지
- 3분반: submit pem_ta hw2c // 9월 30일 금요일 24시 까지
- Deadline: 과제 부여 후 2주 후 24시 까지
- 이메일 제출 절대!! 안 받습니다. 기간 준수해주세요

- 제출하는 파일: hw2a.cpp hw2b.cpp hw2c.cpp hw2.pdf hw2.tex
- hw2.tex에 포함된 사진이 있다면 함께 첨부
- 이외의 파일이 있다면 감점

주의사항

- 치팅 절대 금지.
- 기술서에 있는 출력 요구사항 어길시 감점!
- 이미 작성되어 있는 부분은 건들지 말아주세요

질문

- pemta81718@gmail.com
- 간단한 구글링으로 알 수 있는 내용은 답변하지 않습니다.