창의적 소프트웨어 설계

10 주차 실습 – Template

노인우 , inwoo13@hanyang.ac.kr
한중수 , soohan@hanyang.ac.kr

Overview

목표

- Polymorphism
- Function Templates
- Generic Programming
- C++ Template
- Template Class
- Inline Function

Polymorphism

- Subtype Polymorphism
  - Runtime Polymorphism
- Parametric Polymorphism
  - C++ Template
  - Compile time Polymorphism
- Ad-hoc Polymorphism
  - Overloading
- Coercion Polymorphism
  - (implicit or explicit) casting

Subtype Polymorphism

- Class Cat

```
class Felid {
public:
virtual void meow() = 0;
};

class Cat : public Felid {
public:
void meow() { std::cout << "Meowing like a regular cat! meow!\n"; }
};

class Tiger : public Felid {
public:
void meow() { std::cout << "Meowing like a tiger! MREOWWW!\n"; }
};

class Ocelot : public Felid {
public:
void meow() { std::cout << "Meowing like an ocelot! mews!\n"; }
};
```

Subtype Polymorphism

- Main

```
#include <iostream>
#include "cat.h"

void do_meowing(Felid *cat) {
	cat->meow();
}

int main()
{
	Cat cat;
	Tiger tiger;
	Ocelot ocelot;

	do_meowing(&cat);
	do_meowing(&tiger);
	do_meowing(&ocelot);

	return 0;
}
```

Parametic Polymorphism

- Main

```
#include <iostream>
#include <string>

template <class T>

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


int main() {
	std::cout << ::max(9, 5) << std::endl;	// 9

	std::string foo("foo"), bar("bar");
	std::cout << ::max(foo, bar) << std::endl; // "foo"
}
```

Function Templates

Template as Parameter

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

template <class T> // Template Prefix
T GetMax(T a, T b) {
	T result;

	result = (a>b) ? a : b;

	return (result);
}

int main() {
	int i = 5, j = 6, k;
	long l = 10, m = 5, n;

	k = GetMax<int>(i, j);
	n = GetMax<long>(l, m);
	cout << k << endl;
	cout << n << endl;

	return 0;
}
```

General Programming

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

template<typename T>
void SelectionSort(T* array, int size) {
	for (int i = 0; i < size; ++i) {
		int min_idx = i;

		for (int j = i + 1; j < size; ++j) {
			if (array[min_idx] > array[j]) {
				min_idx = j;
			}
		}

		T tmp = array[i];
		array[i] = array[min_idx]; // set min value to the front
		array[min_idx] = tmp;
	}
}
```

General Programming

```
int main() {
	int array[] = { 2, 5, 3, 1, 4 };
	const int size = sizeof(array) / sizeof(int);

	// You may use SelectionSort(array, size);
	SelectionSort<int>(array, size);

	for (int i = 0; i < size; ++i) {
		cout << " " << array[i];
	}
	cout << endl;

	return 0;
}
```

C++ Template

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

template<class First, class Second>
struct Pair{
	First first;
	Second second;
	Pair(const First& f, const Second& s) : first(f), second(s) {}
};

template<class First, class Second>
Pair<First, Second> MakePair(const First& first, const Second& second) {
	return Pair<First, Second>(first, second);
}

int main() {
	// Equivalently MakePair<int, int>(10, 10);
	Pair<int, int> p = MakePair(10, 10);
	Pair<int, int> q = Pair<int, int>(20, 20);

	return 0;
}
```

Template Class

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

template <typename T>
class Data
{
	T data;
	public:
	Data(T d);
	int SetData(T d);
	T GetData();
};

template <typename T>
Data<T>::Data(T d) {
	data = d;
}

template <typename T>
int Data<T>::SetData(T d) {
	data = d;
	return 0;
}
```

```
int main(void) {
	Data<int> d1(0);
	d1.SetData(10);

	Data<char> d2('a');

	cout << d1.GetData() << endl;
	cout << d2.GetData() << endl;

	return 0;
}
```

Inline Function

- C 의 매크로 함수와 비교
- inline 선언을 해도 컴파일러가 인라인화를 거부 가능

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

inline int add(int a, int b);
int add(int a, int b) {
	return a + b;
}

int add_stack(int a, int b) {
	return a + b;
}

int main(void) {
	int c = add(5, 3);
	int d = add_stack(5, 3);

	cout << c << " , " << d << endl;

	return 0;
}
```

```
int main(void) {
	00162790  push	ebp
	00162791  mov	ebp,esp
	00162793  sub	esp,0D8h
	00162799  push	ebx
	0016279A  push	esi
	0016279B  push	edi
	0016279C  lea	edi,[ebp-0D8h]
	001627A2  mov	ecx,36h
	001627A7  mov	eax,0CCCCCCCCh
	001627AC  rep stos	dword ptr es:[edi]
int c = add(5, 3);
	001627AE  push	3
	001627B0  push	5
	001627B2  call	add (01616EAh)
	001627B7  add	esp,8
	001627BA  mov	dword ptr [c],eax
```

참고자료

1. **class vs typename,** https://stackoverflow.com/questions/213121/use-class-or-typename-for-template-parameters

2. **typename, Stan Lippman,** https://web.archive.org/web/20060619131004/http://blogs.msdn.com/slippman/archive/2004/08/11/212768.aspx


Q&A