



20191010 실습 4주차

박건호

KITRI BoB 8th

5422 RTDCS

devgunho.github.io

과제 Review



x64 / x86



```
4.1_x86x86.cpp  X
CP_Lecture_Code  (Global Scope)
1  #define _CRT_SECURE_NO_WARNINGS
2  #include <stdio.h>
3  #include <string.h>
4
5  struct dog
6  {
7      char name[10];
8      char breed[10];
9      int age;
10     char color[10];
11 };
12
13 int main()
14 {
15     printf("%d\n", sizeof(struct dog*)); // 뭐든 포인터 변수의 크기는 4byte
16     return 0;
17 }
```

Microsoft Visual Studio Debug Console

4

D:\Workspace\19_ComputerProgramming\CP_Lecture_Code\Debug\CP_Lecture_Code.exe: Press any key to close this window . . .

프로세스가 시작되었습니다..

> 8




x64 / x86


- x86은 32bit를 의미하고 x64는 64bit를 의미한다.
- x86으로 표시된 이유는 인텔에서 32비트 이하 계열 제품에 80-86이라는 숫자를 제품명에 붙였기 때문이다.

Windows 7 및 Windows Server 2008 R2

- Windows 7용 업데이트(KB2028551)


이 업데이트는 XPS(XML Paper Specification) 파일을
소가 잘리는 문제를 해결합니다. 이 문제는 XPS 파일에
을 때 발생합니다. 이 업데이트를 설치한 후에는 컴퓨터


 x86 패키지를 지금 다운로드하십시오.

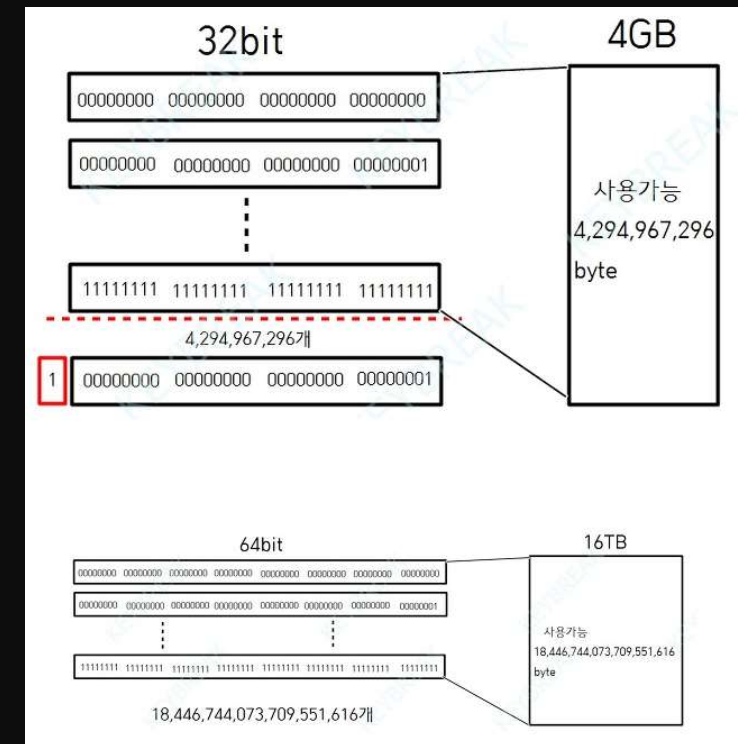
 x64 패키지를 지금 다운로드하십시오.

- Windows 7용 업데이트(KB2028560)

이 업데이트는 그래픽 플랫폼의 성능을 향상시켜줍니다.
다시 시작해야 할 수 있습니다.

 x86 패키지를 지금 다운로드하십시오.

 x64 패키지를 지금 다운로드하십시오.





"\0" / '\0'

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>
#include <iostream>

int main()
{
    printf("%d : %s : %x\n", '\0', '\0', '\0');
    printf("%d : %s : %x\n", "\0", "\0", "\0");
    return 0;
}
```

Microsoft Visual Studio Debug Console

```
0 : (null) : 0
4684608 : : 477b40

D:\#_Workspace#19_ComputerProgramming#CP_Lectur
Press any key to close this window . . .
```



"₩0" / '₩0'

- Double quotes는 string literals를 생성하는 문법이다.
따라서 "₩0"은 문자 그대로 단일 문자 '₩0'을 사용하는 문자열이며,

10진수	16진수	문자	10진수	16진수	문자	10진수	16진수	문자	10진수	16진수	문자
0	0X00	NULL	16	0X10	DLE	32	0x20	SP	48	0x30	0
1	0X01	SOH	17	0X11	DC1	33	0x21	!	49	0x31	1
2	0X02	STX	18	0X12	SC2	34	0x22	"	50	0x32	2
3	0X03	ETX	19	0X13	SC3	35	0x23	#	51	0x33	3
4	0X04	EOT	20	0X14	SC4	36	0x24	\$	52	0x34	4
5	0X05	ENQ	21	0X15	NAK	37	0x25	%	53	0x35	5
6	0X06	ACK	22	0X16	SYN	38	0x26	&	54	0x36	6
7	0X07	BEL	23	0X17	ETB	39	0x27	'	55	0x37	7
8	0X08	BS	24	0X18	CAN	40	0x28	(56	0x38	8
9	0X09	HT	25	0x19	EM	41	0x29)	57	0x39	9

빈 문자열을 쓰고 싶다면 ""가 더 관용적인 방법이다.



malloc 앞에 (int*)의 의미

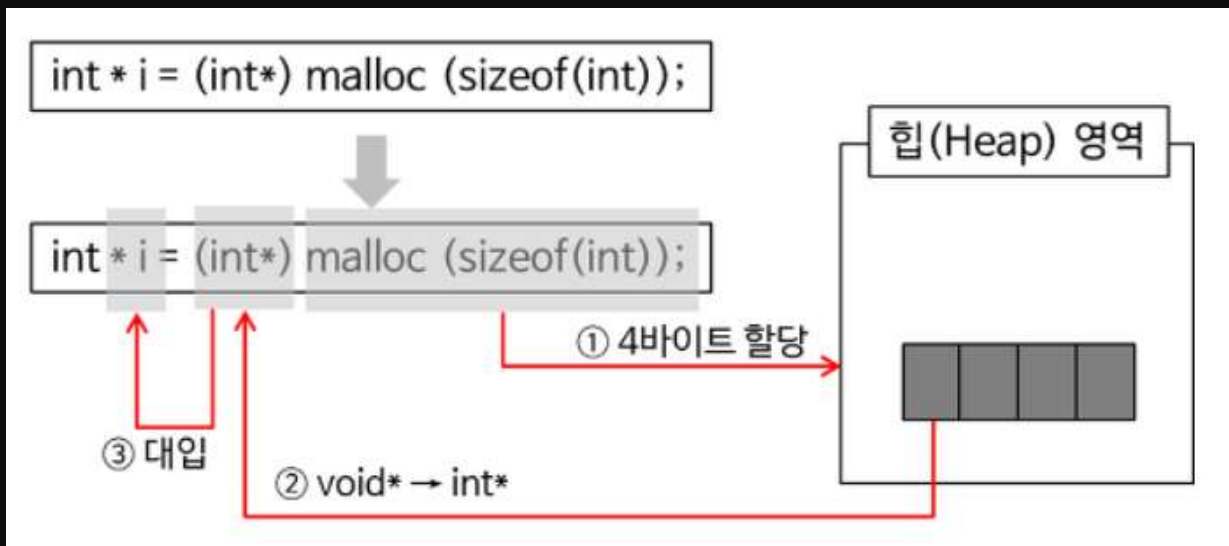
malloc의 함수의 원형은 stdlib.h에 다음과 같이 정의되어 있다.

void* malloc(size_t size)

// malloc 함수의 원형

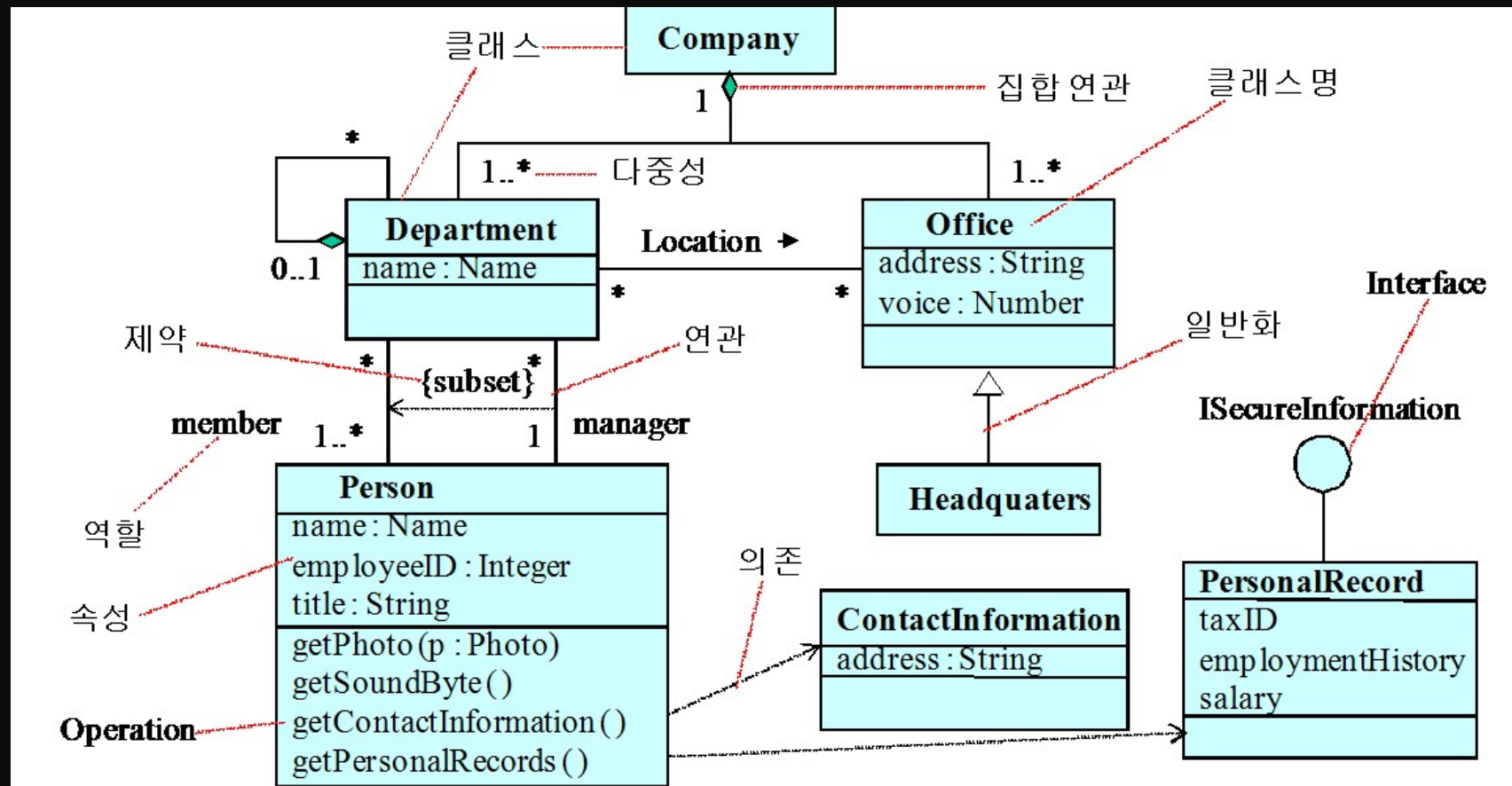
다시 말해 malloc 이라는 함수를 사용하고 Return 되는 값은 void* 형태인데, 우리가 원하는 형태의 데이터를 저장하기 위해서는 원하는 형태로 Type Casting 을 해주어야 한다.

예를 들어 **int형 데이터를 저장하기 위해서는** Return 되는 void* 를 int* 로 변환해야 한다.

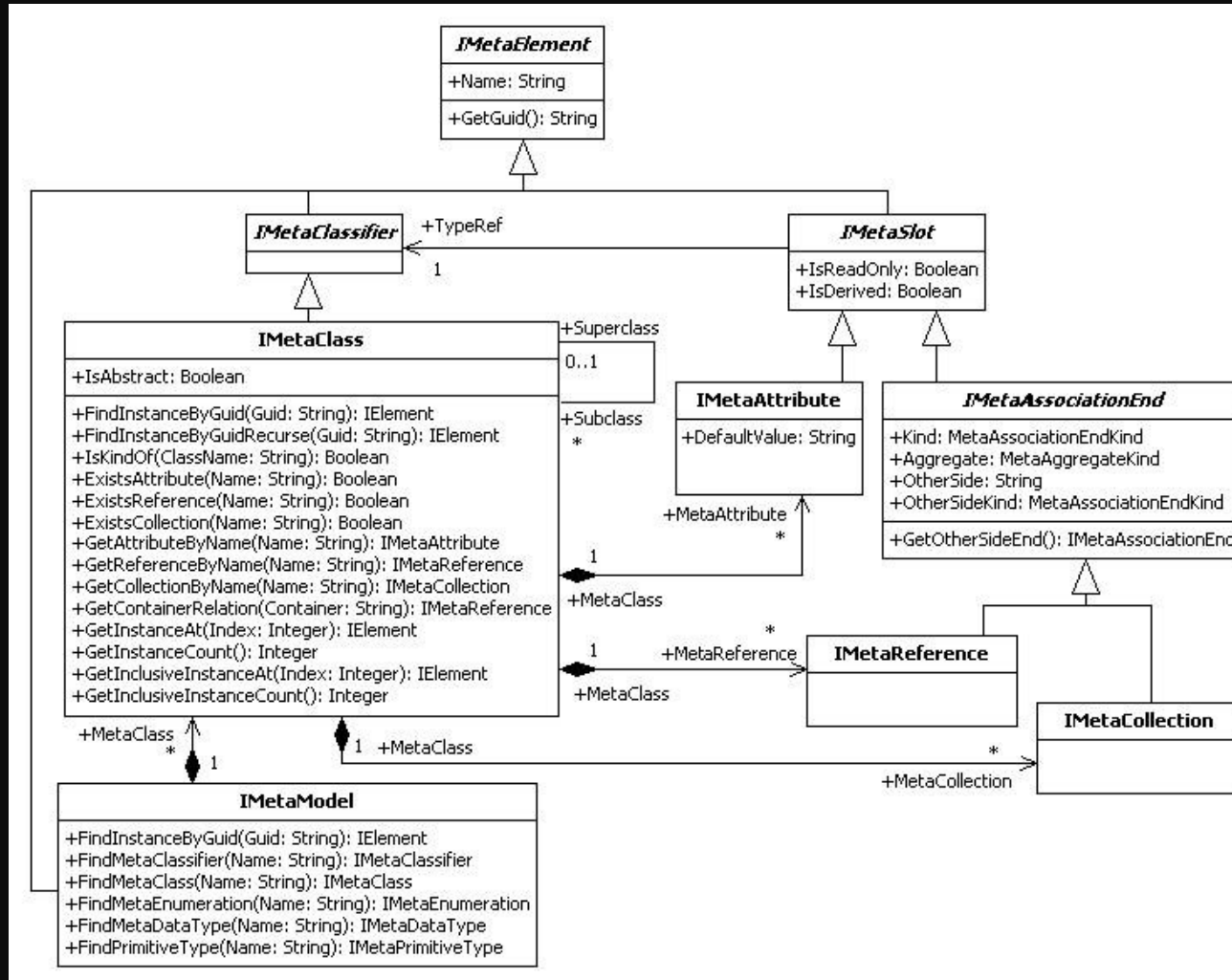


포인터 변수 i 에 4바이트를 할당

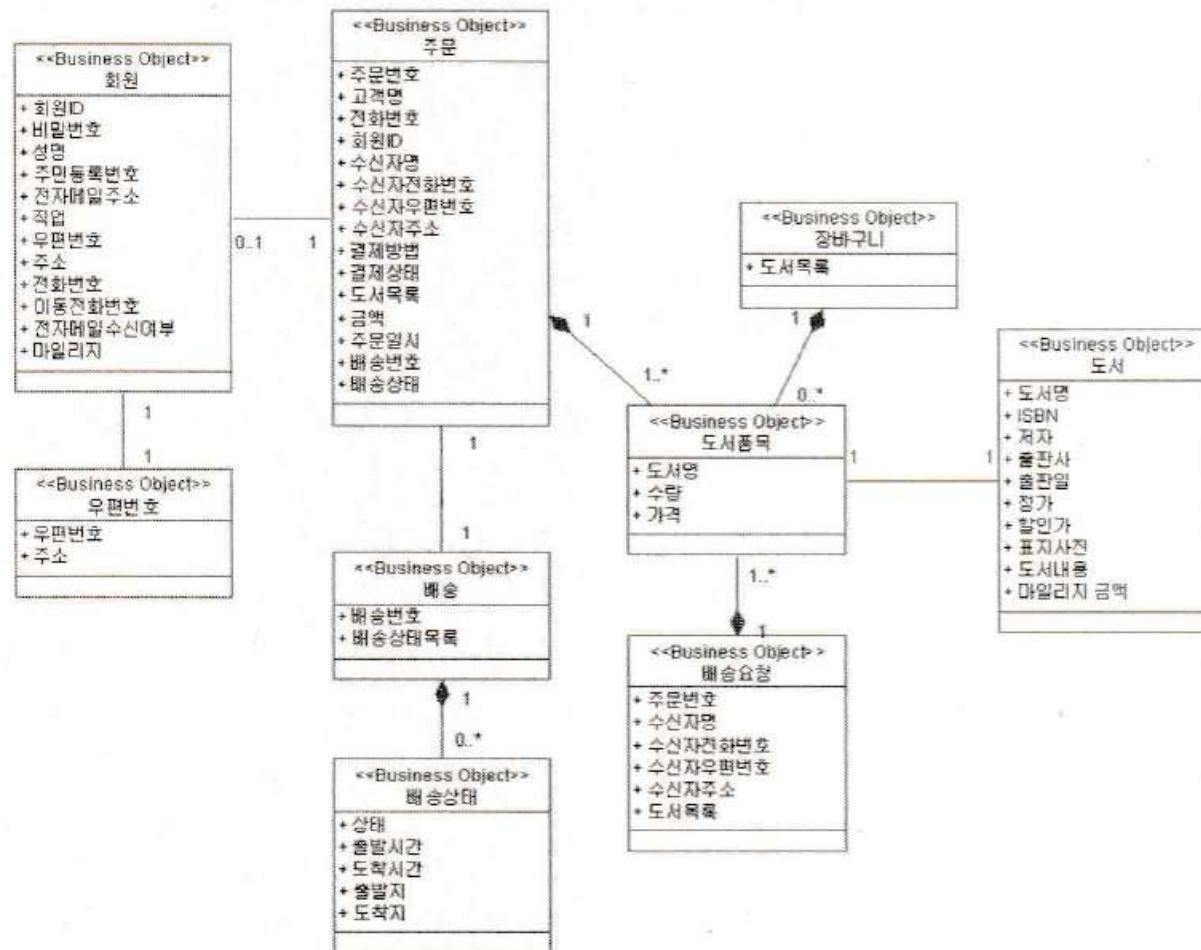
UML



UML



그리고 유스케이스 실현에서의 메시지 흐름을 참조하여 이들 비즈니스 객체 사이의 관계를 도출하고 정의한다. 다음 그림은 온라인 서점 관리에서 도출된 비즈니스 객체 모델의 예이다.





비즈니스 객체	설명
회원	회원의 이름과 주민등록번호, 전자메일주소, 직업, 우편번호, 주소, 전화번호, 이동전화번호, 전자메일을 수신할지의 여부, 확보한 마일리지 등의 정보를 표현한다
도서	고객이 검색 또는 주문한 도서의 정보를 표현한다. 여기에는 도서명, ISBN, 저자, 출판사, 출판일, 정가, 할인가, 표지사진, 도서내용, 구입 시 고객이 받게 되는 마일리지 금액 등의 정보가 포함된다
주문	주문 고객명과 전화번호 및 수신자의 이름과 전화번호, 우편번호, 주소가 포함되며, 결제 방법, 결제 완료 여부를 나타내는 상태, 배송 확인을 위한 배송번호 등의 주문 관련 정보도 포함된다. 또한, 고객이 주문한 도서 목록과 이들 도서의 총합계 금액, 주문 일시 정보도 포함된다
배송 요청	고객이 배송 요청을 한 정보를 표현한다. 여기에는 주문번호와 함께, 이름과 전화번호, 우편번호, 주소, 그리고 고객이 주문한 도서 목록이 포함된다
장바구니	고객이 선택하여 장바구니에 담은 도서의 목록을 관리한다
도서 품목	고객이 주문 또는 배송 요청을 한 도서 품목에 대한 정보를 포함한다. 여기에는 도서명, 수량, 가격 등이 포함된다. 주문 비즈니스 객체와 배송요청 비즈니스 객체는 최소한 하나 이상의 도서 품목 비즈니스 객체를 포함한다. 장바구니 비즈니스 객체는 도서 품목 비즈니스 객체를 포함하지 않을 수도 있다
배송	고객이 주문한 도서의 주문에서부터 현재까지의 배송 현황을 각 단계별로 표현하는 배송 상태 목록을 관리한다
우편번호	우편번호와 주소 정보를 표현한다

[사례 6.3] 비즈니스 객체 설명



Calculator 클래스 디자인하기 (5)

Calculator 클래스

- private 멤버변수

값1, 값2

- public 멤버함수

SetCalculator(double val1, double val2) - 값1, 값2 대입

GetAdd() - 덧셈 결과 리턴

GetSub() - 뺄셈 결과 리턴

GetMul() - 곱셈 결과 리턴

GetDiv() - 나눗셈 결과 리턴 // 출력 시 소수점 2자리 까지



SquCirDiff 클래스 디자인하기 (10)

SquCirDiff 클래스

- private 멤버변수

double 길이 // 원의 경우 '**반지름**', 정사각형의 경우 '**한 변의 길이/2**'

- public 멤버함수

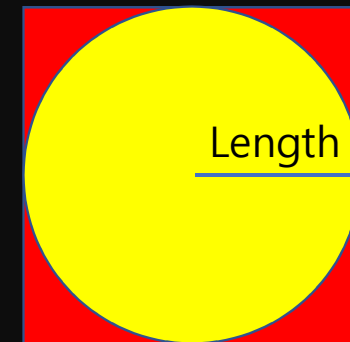
SetLength(double length) - 값 대입

GetSquRound() - 정사각형의 둘레 리턴

GetCirRound() - 원의 둘레 리턴

GetSquCirAreaDiff() - 정사각형에서 원의 넓이 차이 리턴

// PI 값은 3.14로 고정





Fraction 클래스 디자인하기 (18)

Fraction 클래스

- private 멤버변수

int numerator; // 분자

int denominator; // 분모

- public 멤버함수

void SetVal(int numerator, int denominator) - 값1, 값2 대입

Fraction GetAdd(Fraction val1, Fraction val2) - 덧셈 결과 Fraction 리턴

Fraction GetSub(Fraction val1, Fraction val2) - 뺄셈 결과 Fraction 리턴

Fraction GetMul(Fraction val1, Fraction val2) - 곱셈 결과 Fraction 리턴

Fraction GetDiv(Fraction val1, Fraction val2) - 나눗셈 결과 Fraction 리턴

void PrintFraction(Fraction val) - 출력 해주는 함수

-> 모든 결과 값은 기약분수의 형태로 리턴