



객체지향프로그래밍 실습

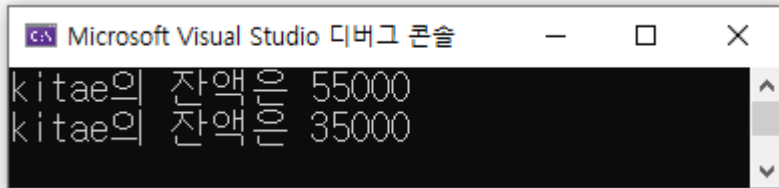


나사렛대학교
KOREA NAZARENE UNIVERSITY

실습 예제 1 – 문제 3

- 은행에서 사용하는 프로그램을 작성하기 위해, 은행 계좌 하나를 표현하는 클래스 Account가 필요하다. 계좌 정보는 계좌의 주인, 계좌 번호, 잔액을 나타내는 3 멤버 변수로 이루어진다. main() 함수의 실행 결과가 다음과 같도록 Account 클래스를 작성하라.

```
int main() {  
    Account a("kitae", 1, 5000); // id 1번, 잔액 5000원, 이름이 kitae인 계좌 생성  
    a.deposit(50000); // 50000원 저금  
    cout << a.getOwner() << "의 잔액은 " << a.inquiry() << endl;  
    int money = a.withdraw(20000); // 20000원 출금. withdraw()는 출금한 실제 금액 리턴  
    cout << a.getOwner() << "의 잔액은 " << a.inquiry() << endl;  
}
```



```
Microsoft Visual Studio 디버그 콘솔  
kitae의 잔액은 55000  
kitae의 잔액은 35000
```

Account 클래스 선언부

```
class Account {  
    string name;  
    int id;  
    int balance;  
  
public:  
    Account(string n, int i, int bal);  
    void deposit(int money);  
    int withdraw(int money);  
    int inquiry();  
    string getOwner() { return name; }  
};
```

Account 클래스 구현부

```
Account::Account(string n, int i, int bal) {  
    name = n;  
    id = i;  
    balance = bal;  
}
```

```
void Account::deposit(int money) {  
    balance += money;  
}
```

```
int Account::inquiry() {  
    return balance;  
}
```

```
int Account::withdraw(int money) {  
    // 저축된 돈이 찾는 금액보자 작은 경우  
    if (money > balance) {  
        money = balance;  
        balance = 0;  
    }  
    else  
        balance -= money;  
  
    return money; // 실제 찾은 금액 리턴  
}
```

실습 예제 2 – 문제 5

- 랜덤 수를 발생시키는 Random 클래스를 만들자. Random 클래스를 이용하여 랜덤한 정수를 10개 출력하는 사례는 다음과 같다. Radom 클래스가 생성자, next(), nextInRange()의 3개의 멤버 함수를 가지도록 작성하고 main() 함수와 합쳐 하나의 cpp 파일을 구현하라.

```
int main() {
    Random r;
    cout << "-- 0에서 " << RAND_MAX << "까지의 랜덤 정수 10 개--" << endl;
    for (int i = 0; i < 10; i++) {
        int n = r.next(); // 0에서 RAND_MAX(32767) 사이의 랜덤한 정수
        cout << n << ' ';
    }
    cout << endl << endl << "-- 2에서 " << "4 까지의 랜덤 정수 10 개 --" << endl;
    for (int i = 0; i < 10; i++) {
        int n = r.nextInRange(2, 4); // 2에서 4 사이의 랜덤한 정수
        cout << n << ' ';
    }
    cout << endl;
}
```



```
Microsoft Visual Studio 디버그 콘솔
-- 0에서 32767까지의 랜덤 정수 10 개--
25631 28703 28841 19937 29794 30285 16497 8647 28724 22673
-- 2에서 4 까지의 랜덤 정수 10 개 --
3 4 4 3 3 4 4 2 4 2
```

Random 클래스 선언부

// 임의의 정수를 랜덤하게 제공하는 클래스

```
class Random {
```

```
public:
```

```
    Random(); // 생성자. 랜덤 seed를 설정한다.
```

```
    int next(); // 랜덤 정수를 리턴한다.
```

```
    int nextInRange(int low, int high); // low와 high 사이의 랜덤 정수를 리턴한다.
```

```
};
```

Random 클래스 구현부

```
Random::Random() {  
    srand((unsigned)time(0)); // 임의의 seed를 설정하여 할 때마다 다른 랜덤 수가 나오게 한다.  
}
```

```
int Random::next() {  
    return rand(); // 0에서 RAND_MAX 사이의 랜덤한 정수 리턴  
}
```

```
int Random::nextIntInRange(int low, int high) {  
    int range = (high - low) + 1;  
    return low + (rand() % range); // low와 high 사이의 랜덤 정수를 리턴한다.  
}
```

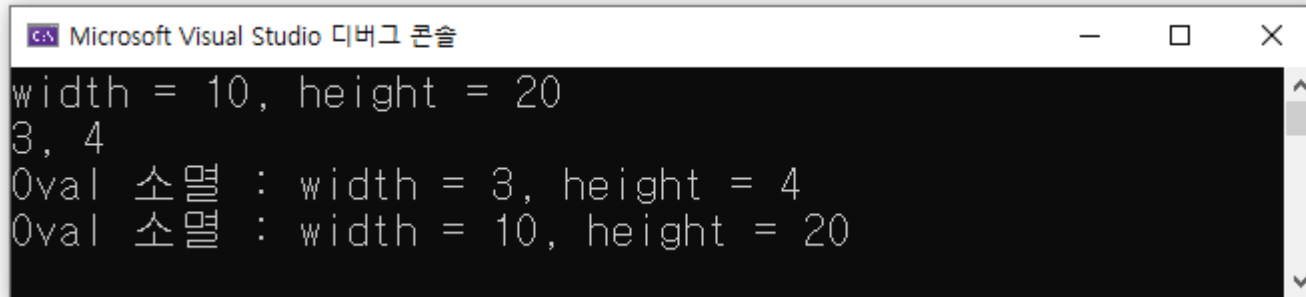
실습 예제 3 – 문제 9

- Oval 클래스는 주어진 사각형에 내접하는 타원을 타원을 추상화한 클래스이다. Oval 클래스의 멤버는 모두 다음과 같다. Oval 클래스를 선언부와 구현부로 나누어 작성하라.
 - 정수값의 사각형 너비와 높이를 가지는 width, height 변수 멤버
 - 너비와 높이 값을 매개 변수로 받는 생성자
 - 너비와 높이를 1로 초기화하는 매개 변수 없는 생성자
 - width와 height를 출력하는 소멸자
 - 타원의 너비를 리턴하는 getWidth() 함수 멤버
 - 타원의 높이를 리턴하는 getHeight() 함수 멤버
 - 타원의 너비와 높이를 변경하는 set(int w, int h) 함수 멤버
 - 타원의 너비와 높이를 화면에 출력하는 show() 함수 멤버

실습 예제 3 – 문제 9

- Oval 클래스를 활용하는 코드의 사례와 실행 결과는 다음과 같다.

```
// main() 함수
int main() {
    Oval a, b(3, 4);
    a.set(10, 20);
    a.show();
    cout << b.getWidth() << ", " << b.getHeight() << endl;
}
```



The screenshot shows the 'Microsoft Visual Studio 디버그 콘솔' (Debug Console) window. The output text is as follows:

```
width = 10, height = 20
3, 4
Oval 소멸 : width = 3, height = 4
Oval 소멸 : width = 10, height = 20
```

Oval 클래스 선언부

```
class Oval {  
    int width, height;  
public:  
    Oval();  
    Oval(int w, int h);  
    ~Oval();  
    int getWidth();  
    int getHeight();  
    void set(int w, int h);  
    void show();  
};
```

Oval 클래스 구현부

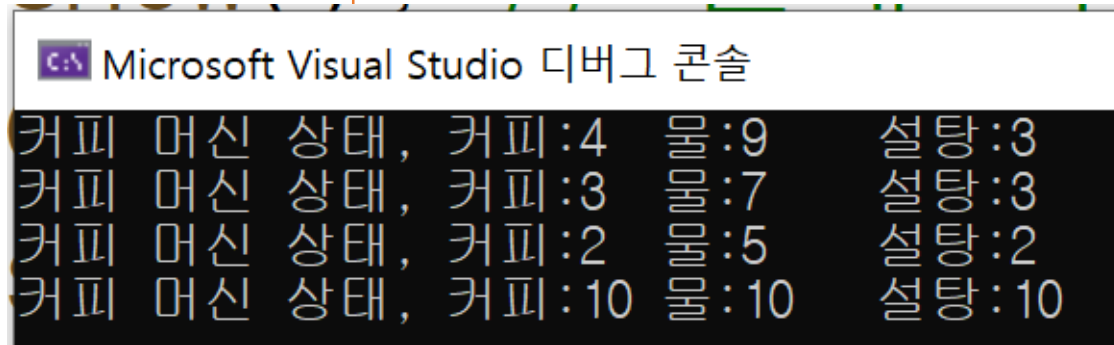
```
Oval::Oval() {  
    width = height = 1;  
}  
Oval::Oval(int w, int h) {  
    width = w; height = h;  
}  
Oval::~~Oval() {  
    cout << "Oval 소멸 : ";  
    show();  
}  
void Oval::set(int w, int h) {  
    width = w; height = h;  
}
```

```
void Oval::show() {  
    cout << "width = " << width << ", "  
    << "height = " << height << endl;  
}  
int Oval::getWidth() {  
    return width;  
}  
int Oval::getHeight() {  
    return height;  
}
```

LAB 1 – 문제 4

- CoffeeMachine 클래스를 만들어보자. main() 함수와 실행 결과가 다음과 같도록 CoffeeMachine 클래스를 작성하라.
 - 에스프레소: 커피 1, 물 1 소비 • 아메리카노: 커피 1, 물 2 소비 • 설탕 커피: 커피 1, 물 2, 설탕 1 소비
- CoffeeMachine 클래스에는 어떤 멤버 변수와 어떤 멤버 함수가 필요한지 잘 판단하여 작성하라

```
int main() {  
    CoffeeMachine java(5, 10, 3); // 커피량:5, 물량:10, 설탕:6으로 초기화  
    java.drinkEspresso(); // 커피 1, 물 1 소비  
    java.show(); // 현재 커피 머신의 상태 출력  
    java.drinkAmericano(); // 커피 1, 물 2 소비  
    java.show(); // 현재 커피 머신의 상태 출력  
    java.drinkSugarCoffee(); // 커피 1, 물 2, 설탕 1 소비  
    java.show(); // 현재 커피 머신의 상태 출력  
    java.fill(); // 커피 10, 물 10, 설탕 10으로 채우기  
    java.show(); // 현재 커피 머신의 상태 출력  
}
```



Microsoft Visual Studio 디버그 콘솔

커피	머신	상태,	커피:4	물:9	설탕:3
커피	머신	상태,	커피:3	물:7	설탕:3
커피	머신	상태,	커피:2	물:5	설탕:2
커피	머신	상태,	커피:10	물:10	설탕:10

LAB 2 – 문제 6

- 문제 5번을 참고하여 짝수 정수만 랜덤하게 발생시키는 EvenRandom 클래스를 작성하고 EvenRandom 클래스를 이용하여 10개의 짝수를 랜덤하게 출력하는 프로그램을 완성하라. 0도 짝수로 처리한다.
 - 생성자, next(), nextInRange()의 멤버함수를 가지도록 클래스 생성

```
int main() {
    EvenRandom r;
    cout << "-- 0에서 " << RAND_MAX << "까지의 랜덤 정수 10 개--" << endl;
    for (int i = 0; i < 10; i++) {
        int n = r.next(); // 0에서 RAND_MAX(32767) 사이의 랜덤한 정수
        cout << n << ' '; // RAND_MAX 상수는 <cstdlib> 헤더파일에 선언되어 있는 32767 정수
    }
    cout << endl << endl << "-- 2에서 " << "10 까지의 랜덤 정수 10 개 --" << endl;
    for (int i = 0; i < 10; i++) {
        int n = r.nextInRange(2, 10); // 2에서 10 사이의 랜덤한 정수
        cout << n << ' ';
    }
    cout << endl;
}
```

Microsoft Visual Studio 디버그 콘솔

```
-- 0에서 32767까지의 랜덤 정수 10 개--
10884 26572 11112 12584 13642 32382 7766 18532 12820 3562

-- 2에서 10 까지의 랜덤 정수 10 개 --
10 2 4 10 8 2 2 4 10 4
```

LAB 3 – 문제 10

- 더하기(+), 빼기(-), 곱하기(*), 나누기(/)를 수행하는 4개의 클래스 Add, Sub, Mul, Div를 만들고자 한다.
 - 공통 멤버
 - int 타입 변수 a, b: 피연산자
 - void setValue(int x, int y) 함수: 매개변수 x,y를 멤버 a,b에 복사
 - int calculate() 함수: 여산을 실행하고 결과 리턴
 - main() 함수
 - Add, Sub, Mul, Div 클래스 타입의 객체 a, s, m, d를 생성
 - 키보드로부터 두 개의 정수와 연산자를 입력받음
 - a, s, m, d 객체 중에서 연산을 처리할 객체의 setValue() 함수를 호출한 후, calculate()를 호출하여 결과를 화면에 출력
 - 프로그램은 무한루프
- 클래스의 선언부와 구현부를 헤더 파일과 cpp 파일로 나누어 프로그램을 작성하라.



THANKS FOR YOUR ATTENTION



나사렛대학교
KOREA NAZARENE UNIVERSITY