

Programming Methodology

Practice Session #9

Polymorphism

Polymorphism

- 상속 관계에 있는 서로 다른 class들의 object들이 같은 이름의 function을 호출해도 서로 다른 역할을 수행할 수 있는 기능
- 상위 클래스의 함수를 **virtual**로 선언하고 하위 클래스에서 해당 함수를 **override**하면 객체의 **virtual** 함수를 호출 할 때 동적(dynamic)으로 해당 객체에 맞는 (**override** 된) 멤버 함수를 호출한다.

Is-a Relationship

- Base class **B**와 이를 상속받은 derived class **D**가 있다면 '**D is a B**' 혹은 '**D는 B이다**'의 관계가 성립한다.

```
class Person                                // Base class
{ ... };

class Professor : public Person             // Derived class
{ ... };

class Student : public Person              // Derived class
{ ... };

void main()
{
    Person * obj1 = new Professor;          // Professor is a Person.
    Person * obj2 = new Student;           // Student is a Person.
    // Student * obj3 = new Person         // 'Person is a Student'의 관계가
                                           // 성립하지 않으므로 error.
}
```

Polymorphism Example (1)

```
class Shape {  
    public:  
        virtual double area() {  
            return -1;  
        }  
};
```

```
class Circle : public Shape {  
    private:  
        double radius;  
    public:  
        virtual double area() {  
            return 3.14 * radius * radius;  
        }  
};
```

```
class Rectangle : public Shape {  
    private:  
        double a, b;  
    public:  
        virtual double area() {  
            return a * b;  
        }  
};
```

Polymorphism Example (2)

area() 함수를 **virtual**로 선언하지 않은 경우

```
int main()
{
    Shape* s1 = new Circle(10);
    Shape* s2 = new Rectangle(5, 10);

    cout << "C: " << s1->area();
    cout << "R: " << s2->area();
}
```



C: -1
R: -1

Result

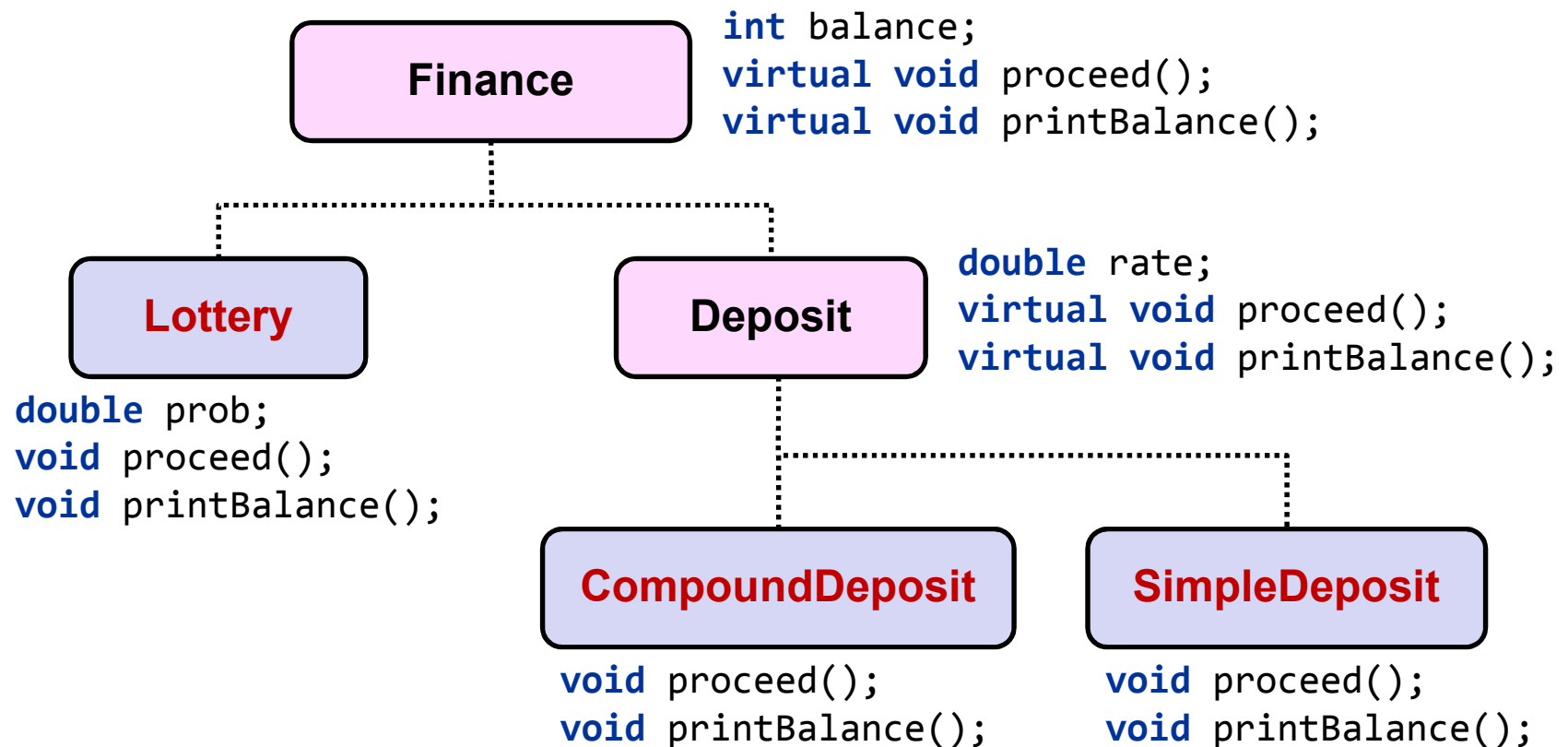


C: 314
R: 50

area() 함수를 **virtual**로 선언한 경우

Sample Practice (1)

- Class Hierarchy



Sample Practice (2)

- **SimpleDeposit, CompoundDeposit, Lottery** 클래스를 각각 구현한다.
 - Finance 클래스에 **virtual**로 선언된 `proceed()` 함수와 `printbalance()` 함수를 각각의 하위 클래스에 맞게 **재정의 (overriding)**한다.

Sample Practice (3)

- **Finance** class

- 최상위 클래스로서, 잔액을 나타내는 **int** type의 `balance` 변수와, 한 주기가 지났음을 나타내는 `proceed()` 함수, 잔액을 출력해주는 `printBalance()` 함수를 **virtual**로 선언한다.

- **Deposit** class

- `SimpleDeposit` 클래스와 `CompoundDeposit` 클래스의 상위 클래스로서, 이자율을 나타내는 **double** type의 `rate` 변수를 가진다.

Sample Practice (4)

- **SimpleDeposit** class
 - 단리로 이자를 계산한다.
 - 클래스를 생성할 때 초기 금액, 이율을 parameter로 받는다.
- **CompoundDeposit** class
 - 복리로 이자를 계산한다.
 - 클래스를 생성할 때 초기 금액, 이율을 parameter로 받는다.
- **Lottery** class
 - 매 달 일정 금액(price) 만큼의 금액이 줄어들며, 특정 확률(prob)로 상금(reward)를 얻는다.
 - 클래스를 생성할 때 초기 금액, 가격, 상금, 당첨 확률을 parameter로 받는다.

Sample Practice (5)

- 반드시 **상속과 virtual 키워드**를 사용하여 구현하여야 한다.
- Lottery 클래스의 `proceed()` 함수에서 당첨 여부는 `cstdlib.h`의 **`rand()`** 함수를 사용한다.
- 만약 필요하다면 다른 멤버 변수나 멤버 함수를 구현해도 된다.

Sample Practice (6)

```
void main()
{
    // Randomize
    time_t t;
    srand((unsigned)time(&t));

    Finance* list[3];

    list[0] = new SimpleDeposit(10000, 0.05);
    list[1] = new CompoundDeposit(10000, 0.03);
    list[2] = new Lottery(10000, 500, 50000, 0.2);

    for(int i=0; i<10; i++) // Each stages
    {
        cout << "[ Stage " << i << " ]" << endl;
        for(int j=0; j<3; j++)
        {
            list[j]->proceed();
            list[j]->printBalance();
        }
        cout << endl;
    }
}
```

Sample Practice (7)

```
[ Stage 0 ]  
SimpleDeposit: $10500  
CompoundDeposit: $10300  
Lottery: $9500
```

```
[ Stage 1 ]  
SimpleDeposit: $11000  
CompoundDeposit: $10609  
--Congratulations!! --  
Lottery: $59500
```

```
[ Stage 2 ]  
SimpleDeposit: $11500  
CompoundDeposit: $10927  
Lottery: $59000
```

```
[ Stage 3 ]  
SimpleDeposit: $12000  
CompoundDeposit: $11254  
Lottery: $58500
```

```
[ Stage 4 ]  
SimpleDeposit: $12500  
CompoundDeposit: $11591  
Lottery: $58000
```

```
[ Stage 5 ]  
SimpleDeposit: $13000  
CompoundDeposit: $11938  
Lottery: $57500
```

```
[ Stage 6 ]  
SimpleDeposit: $13500  
CompoundDeposit: $12296  
Lottery: $57000
```

```
[ Stage 7 ]  
SimpleDeposit: $14000  
CompoundDeposit: $12664  
Lottery: $56500
```

```
[ Stage 8 ]  
SimpleDeposit: $14500  
CompoundDeposit: $13043  
Lottery: $56000
```

```
[ Stage 9 ]  
SimpleDeposit: $15000  
CompoundDeposit: $13434  
Lottery: $55500
```