

03

학습 목표

1. 실세계의 객체와 C++ 객체에 대해 이해한다.
2. C++ 클래스를 작성할 수 있다.
3. 객체를 생성하고 활용할 수 있다.
4. 생성자와 소멸자를 알고 작성할 수 있다.
5. private, protected, public 접근 지정자를 이해한다.
6. 인라인 함수의 목적을 이해하고 활용할 수 있다.

소멸자(destructor)

3

- 객체가 소멸될 때 필요한 마무리 작업을 수행하는 멤버함수
 - ▣ 동적할당 받은 메모리 해제(4장), 파일 저장 및 닫기(12장), 등
- **객체가 소멸되는 시점에서 자동으로 호출되는 멤버함수**
- 소멸자 이름은 클래스 이름 앞에 ~(틸드)를 붙임
- 소멸자는 리턴 타입을 선언하지 않음
 - ▣ 리턴값 없음 -> void 타입도 안됨
- 소멸자는 매개 변수 없음 -> 중복 불가능
 - ▣ 소멸자는 한 클래스 내에 오직 한 개만 작성 가능
- 객체 소멸시 오직 한번만 호출
 - ▣ 사용자가 임의로 호출할 수 없음
 - ▣ 객체가 메모리에서 소멸 직전 자동 호출됨

소멸자(destructor)

4

```
class Circle {  
    Circle();  
    Circle(int r);  
    ...  
    ~Circle();  
};
```

소멸자 선언

리턴 타입도 없고
매개 변수도 없음

소멸자 구현

```
Circle::~~Circle() {  
    ...  
}
```

소멸자는 오직 하
나만 존재

디폴트 소멸자(destructor)

5

- 개발자가 클래스에 소멸자를 작성하지 않으면 컴파일러에 의해 디폴트 소멸자가 자동으로 추가됨
- 디폴트 소멸자는 매개변수가 없고 아무 일도 하지 않는 소멸자

```
class Circle {  
public:  
    int radius;  
    double getArea();  
};
```



```
class Circle {  
public:  
    int radius;  
    double getArea();  
    Circle( );  
    ~Circle( );  
};  
Circle::Circle( ){ }  
Circle::~~Circle( ){ }
```

소멸자(destructor)

6

- 객체 생성의 의미
 - ▣ 객체생성 또는 변수선언 하면 메모리를 할당 받고 사용가능 하게 됨
 - ▣ 운영체제가 관리하는 메모리 공간을 프로그램에서 사용할 수 있게 됨을 의미
- 객체 소멸의 의미
 - ▣ 함수가 종료되거나 프로그램이 종료되면 객체가 소멸됨
 - ▣ 프로그램에서 사용하던 메모리 공간을 운영체제에 반납하고 더 이상 그 공간을 사용할 수 없다는 의미

예제 3-7 Circle 클래스에 소멸자 작성 및 실행

7

```
#include <iostream>
using namespace std;
class Circle {
public:
    int radius;
    Circle();
    Circle(int r);
    ~Circle();           // 소멸자 선언
    double getArea();
};
Circle::Circle() {
    radius = 1;
    cout << "반지름 " << radius << " 원 생성" << endl;
}
Circle::Circle(int r) {
    radius = r;
    cout << "반지름 " << radius << " 원 생성" << endl;
}
```

예제 3-7 Circle 클래스에 소멸자 작성 및 실행

8

```
Circle::~~Circle() {           // 소멸자 구현
    cout << "반지름" << radius << "원 소멸" << endl;
}

int main() {
    Circle donut;
    Circle pizza(30);

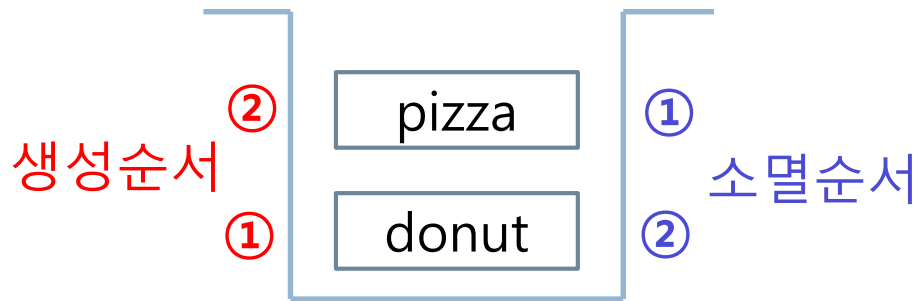
    return 0;
}
```

반지름 1 원 생성
반지름 30 원 생성
반지름 30 원 소멸
반지름 1 원 소멸

예제 3-7 Circle 클래스에 소멸자 작성 및 실행

9

- 지역 변수(객체)는 스택 메모리에 할당 -> Last In First Out
- main함수가 호출되면 donut -> pizza 순서로 생성
- main함수가 종료하면 pizza -> donut 순서로 소멸
- 객체는 생성순서의 반대 순서으로 소멸됨



스택 메모리

-> 입구가 하나인 컵의 구조와 같음

생성자/소멸자 실행 순서

10

- 객체가 선언된 위치에 따른 분류
 - ▣ 지역 객체: 함수 내부에 선언된 객체로서, 지역변수와 같음
 - ▣ 전역 객체 : 함수의 외부에 선언된 객체로서, 전역변수와 같음
- 객체 생성 순서
 - ▣ 전역 객체는 프로그램이 실행될 때 선언된 순서대로 생성
 - ▣ 지역 객체는 함수가 호출될 때 선언된 순서대로 생성
- 객체 소멸 순서
 - ▣ 함수가 종료하면, 지역 객체가 생성된 순서의 역순으로 소멸
 - ▣ 프로그램이 종료하면, 전역 객체가 생성된 순서의 역순으로 소멸

생성자/소멸자 실행 순서

11

실행순서

프로그램시작

초기화작업->전역변수, 전역객체 생성

main함수시작->지역변수, 지역객체 생성

함수1시작->지역변수, 지역객체 생성

⋮

함수1종료->지역변수, 지역객체 소멸

⋮

함수n시작->지역변수, 지역객체 생성

⋮

함수n종료->지역변수, 지역객체 소멸

main함수종료->지역변수, 지역객체 소멸

정리작업->전역변수, 전역객체 소멸

프로그램종료

예제3-8 지역객체와 전역객체의 생성 및 소멸 순서

12

```
#include <iostream>
using namespace std;
class Circle {
public:
    int radius;
    Circle();
    Circle(int r);
    ~Circle();
    double getArea();
};
Circle::Circle() {
    radius = 1;
    cout << "반지름 " << radius << " 원 생성" << endl;
}
Circle::Circle(int r) {
    radius = r;
    cout << "반지름 " << radius << " 원 생성" << endl;
}
```

예제3-8 지역객체와 전역객체의 생성및소멸 순서

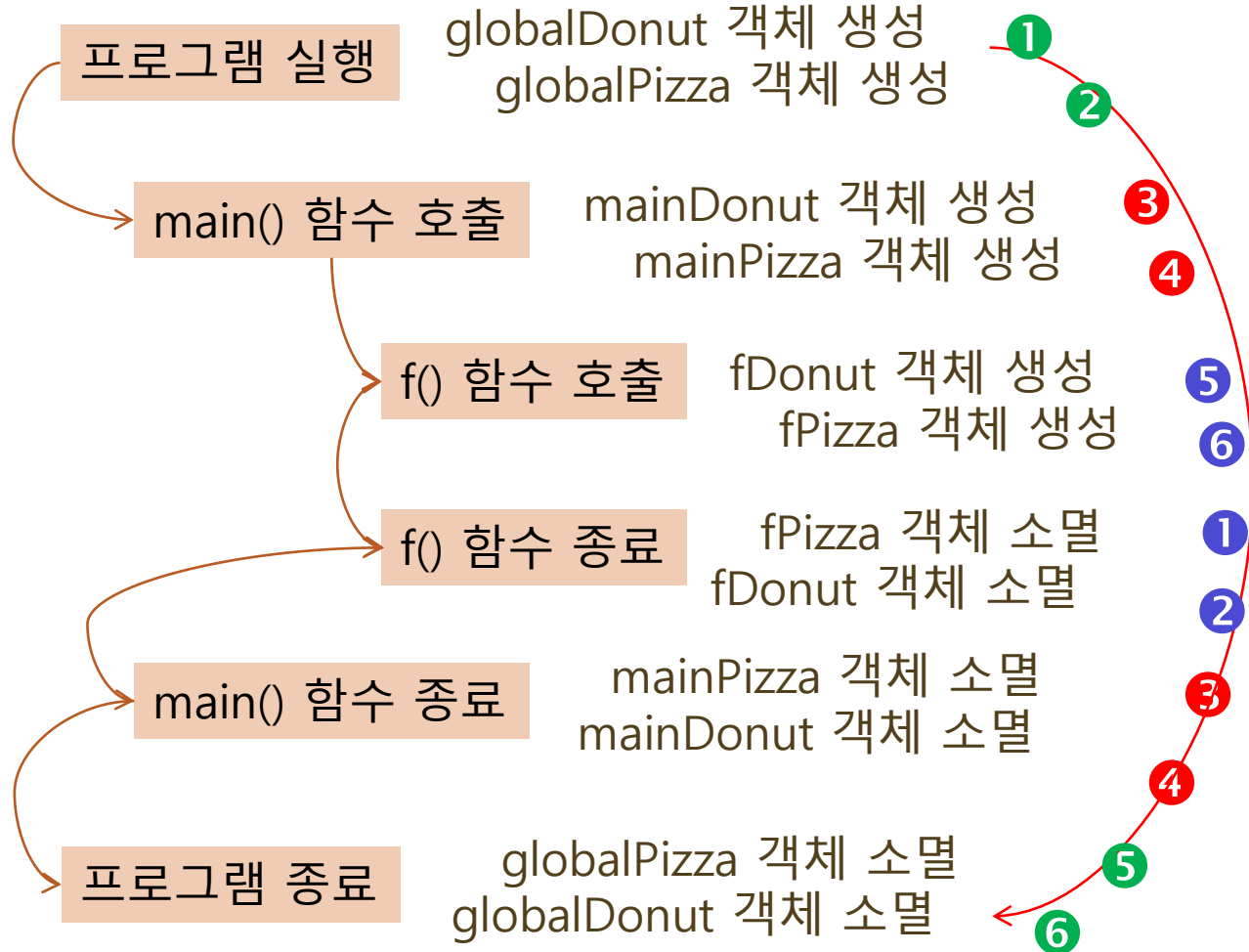
13

```
Circle::~Circle() {  
    cout << "반지름" << radius << "원 소멸" << endl;  
}  
double Circle::getArea() {  
    return 3.14 * radius * radius;  
}  
Circle globalDonut(1000); // 전역객체 생성  
Circle globalPizza(2000); // 전역객체 생성  
void f() {  
    Circle fDonut(100); // 지역객체 생성  
    Circle fPizza(200); // 지역객체 생성  
}  
int main() {  
    Circle mainDonut; // 지역객체 생성  
    Circle mainPizza(30); // 지역객체 생성  
    f();  
}
```

반지름 1000 원 생성
반지름 2000 원 생성
반지름 1 원 생성
반지름 30 원 생성
반지름 100 원 생성
반지름 200 원 생성
반지름 200 원 소멸
반지름 100 원 소멸
반지름 30 원 소멸
반지름 1 원 소멸
반지름 2000 원 소멸
반지름 1000 원 소멸

예제3-8의 지역객체와 전역객체의 생성/소멸과정

14



실습과제1

15

- 객체의 생성과 소멸의 의미를 설명하라.
- 프로그램 실행 후 main함수가 실행되기전에 어떤 일이 처리되는지 조사해보라.

실습과제2

16

- 삼각형 클래스 Triangle 를 만들고 다음 결과가 나오도록 멤버변수, 멤버함수, 생성자, 소멸자를 구현하라.

// 클래스 정의

```
int main() {  
    Triangle tri1;                //밑변,높이 1로 초기화  
    cout << " 삼각형의 면적은 " << tri1.getArea() << endl;  
  
    Triangle tri2(2, 4);          //밑변=2,높이=4로 초기화  
    cout << " 삼각형의 면적은 " << tri2.getArea() << endl;  
}
```

밑변 1높이 1인 삼각형 생성
삼각형의 면적은 0.5
밑변 2높이 4인 삼각형 생성
삼각형의 면적은 4
밑변 2높이 4인 삼각형 소멸
밑변 1높이 1인 삼각형 소멸

실습과제3

17

- 실습과제 2번에서 모든 객체를 전역객체로 생성하고 다음 결과가 나오도록 해보시오.

밑변 4높이 8인 삼각형 생성
밑변 2높이 2인 삼각형 생성
삼각형의 면적은 2
삼각형의 면적은 16
밑변 2높이 2인 삼각형 소멸
밑변 4높이 8인 삼각형 소멸

실습과제4

18

- 구 클래스 Sphere 를 만들고 만들고 다음 결과가 나오도록 멤버 변수, 멤버함수, 생성자, 소멸자를 구현하라.

// 클래스 정의

```
int main() {  
    Sphere sph1;           // 반지름 =1로 초기화  
    cout << "구의 부피는" << sph1.getVolume() << endl;  
  
    Sphere sph2(3);        // 반지름 =3로 초기화  
    cout << " 구의 부피는" << sph2.getVolume () << endl;  
}
```

반지름 1인 구 생성
구의 부피는 3.45
반지름 3인 구 생성
구의 부피는 110.42
반지름 3인 구 소멸
반지름 1인 구 소멸

실습과제5

19

- 실습과제 4번에서 모든 객체를 전역객체로 선언하고 다음 결과가 나오도록 수정해 보시오.

```
반지름 10인 구 생성  
반지름 20인 구 생성  
구의 부피는 4186.66  
구의 부피는 33493.33  
반지름 20인 구 소멸  
반지름 10인 구 소멸
```

과제 제출 방법

20

- 소스코드, 라인단위의 주석, 실행결과를 포함하는 pdf파일을 작성한 후 eclass 과제 게시판에 업로드, **반드시 하나의 pdf파일로 업로드할 것**
- 기한 : 과제 게시판에 마감시간 참조
- 실행결과를 캡처할 때 글자를 알아보기 쉽게 확대해서 캡처할 것.
- 소스코드의 첫 부분은 아래처럼 제목,날짜,작성자(학번,이름)를 작성할 것

```
// *****  
//   제   목   : 정수 4개의 평균을 구하는 프로그램  
//   날   짜   : 2023년 9월10일  
//   작성자   : 15010101 홍길동  
// *****  
  
// 소스코드 작성
```