Fast campus

컴퓨터공학 All in One

C/C++ 문법, 자료구조 및 심화 프로젝트 (나동빈)
제 47강 - C++의 생성자와 소멸자

Fast campus

학습 목표

C++의 생성자와 소멸자

1) C++에서 객체를 효과적으로 다루기 위한 생성자와 소멸자의 개념에 대해서 이해할 수 있습니다.

C++의 생성자와 소멸자

C++의 생성자

C++에서는 생성자(Constructor)를 이용해 객체를 생성함과 동시에 멤버 변수를 초기화할 수 있습니다. 생성자는 특별한 메소드로, 클래스의 이름과 동일한 이름의 메소드로 구현됩니다.

생성자는 반환 값이 없습니다. 생성자는 여러 번 정의되어 다양한 방법으로 객체를 초기화할 수 있습니다.

C++의 생성자와 소멸자

C++의 생성자 사용해보기 ①

```
class Character {
private:
  string name;
  int ragePoint;
  int hp;
  int damage;
public:
  Character(string name, int hp, int damage) {
    this->name = name;
    this->ragePoint = 0;
    this->hp = hp;
    this->damage = damage;
  }
  void show() {
    cout << name << "[" << ragePoint << "] " << hp << " " << damage << '\n' ;
  }
};
```

C++의 생성자와 소멸자

C++의 생성자 사용해보기 ②

```
int main(void) {
  Character character = Character("슬라임", 50, 10);
  character.show();
  system("pause");
}
```

C++의 생성자와 소멸자

C++의 기본 생성자

C++에서 별도로 생성자를 구현하지 않으면 기본 생성자(Default Constructor)가 사용됩니다. 기본 생성자는 매개변수를 가지지 않으며 멤버 변수는 0, NULL 등의 값으로 초기화됩니다.

C++의 생성자와 소멸자

C++의 기본 생성자 사용해보기

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

using namespace std;

class Character {
private:
  string name;
  int ragePoint;
  int hp;
  int damage;
public:
  void show() {
    cout << name << "[" << ragePoint << "] " << hp << " " << damage << '\n' ;
  }
};

int main(void) {
  Character character = Character();
  character.show();
  system("pause");
}
```

C++의 생성자와 소멸자

C++의 복사 생성자

C++는 복사 생성자(Copy Constructor)는 다른 인스턴스의 참조(Reference)를 인수로 받아서, 그 참조를 이용해 자신의 인스턴스를 초기화할 수 있도록 해줍니다. 대표적인 복사 방법인 깊은 복사(Deep Copy)를 이용해 만들어진 인스턴스는 기존의 인스턴스와 다른 메모리 공간에 할당되어 독립적입니다.

C++의 생성자와 소멸자

C++의 복사 생성자 사용해보기 ①

```
class Character {
private:
  string name;
  int ragePoint;
  int hp;
  int damage;
public:
  Character(string name, int hp, int damage): name(name), ragePoint(0), hp(hp), damage(damage) { }
  Character(const Character& other) {
    name = other.name;
    ragePoint = other.ragePoint;
    hp = other.hp;
    damage = other.damage;
  }
  void pointUp() { ragePoint++; }
  void show() {
    cout << name << "[" << ragePoint << "] " << hp << " " << damage << '\n';
  }
};
```

C++의 생성자와 소멸자

C++의 복사 생성자 사용해보기 ②

```
int main(void) {
  Character character1("슬라임", 10, 20);
  character1.pointUp();
  Character character2(character1);
  character2.pointUp();
  character1.show();
  character2.show();
  system("pause");
}
```

C++의 생성자와 소멸자

C++의 소멸자

C++의 소멸자(Destructor)는 객체의 수명이 끝났을 때 객체를 제거하기 위한 목적으로 사용됩니다. 객체의 수명이 끝났을 때 자동으로 컴파일러가 소멸자 함수를 호출합니다.

C++의 소멸자 또한 생성자처럼 클래스의 이름과 동일하며 물결 기호(~)를 이용해 정의할 수 있습니다.

C++의 생성자와 소멸자

C++의 소멸자 사용해보기 ①

```
class Character {
private:
  string name;
  int ragePoint;
  int hp;
  int damage;
public:
  Character(string name, int hp, int damage): name(name), ragePoint(0), hp(hp), damage(damage) { }
  ~Character() {
    cout << "[객체가 소멸됩니다.]\n";
  }
  void pointUp() { ragePoint++; }
  void show() {
    cout << name << "[" << ragePoint << "] " << hp << " " << damage << '\n';
  }
};
```

C++의 생성자와 소멸자

C++의 소멸자 사용해보기 ②

```
int main(void) {
  Character* character1 = new Character("슬라임", 10, 20);
  character1->pointUp();
  Character character2(*character1);
  character2.pointUp();
  character1->show();
  character2.show();

  delete character1; // 동적 할당을 이용했으므로 성공적으로 소멸됨
  delete &character2; // 동적 할당을 이용하지 않았으므로 오류가 발생함
  system("pause");
}
```

배운 내용 정리하기

C++의 생성자와 소멸자

1) C++의 생성자와 소멸자는 객체를 초기화하거나 제거할 때 사용할 수 있는 문법입니다.