2020년도 2학기 컴퓨터공학설계및실험I

3주차 예비 보고서

20170175 김태안

1. 실습 목적

C++는 객체지향 프로그래밍 개념을 도입하여 클래스(Class)를 확장하고 수정하는 수단으로 클래스 상속(Class Inheritance)을 제공한다. 이번 실험에서는 효율적인 프로그래밍을 위하여 클래스의 상속 방법을 익혀보도록 한다.

1. 관련 이론

**객체 지향 프로그래밍**

많은 프로그래밍 언어가 개발되면서 언어는 패러다임을 갖게 되었다. 그 중에서 객체 지향 프로그램은 프로그램을 상호 작용하는 객체의 집합으로 보는 관점이다. 각 객체는 class의 instance로 구성되어 있고, class는 서로 상속 관계에 있는 계층 구조이다. 객체 지향 프로그래밍의 특징은 1) 현실 세계의 문제를 잘 반영하는 모델을 구현할 수 있고, 2) 모듈화되어 있으며, 3) 소프트웨어의 재사용이 가능하고, 4) 각 객체가 병렬적으로 실행될 수 있다.

객체 지향 프로그래밍을 구현하기 위해 많은 기법을 사용하는데, 그 중 하나가 추상 데이터 타입 (Abstract Data Type, ADT)이다. ADT는 캡슐화와 정보 은닉을 지원하는 데이터 타입이다. 캡슐화란 데이터와 코드를 하나로 묶어 하나의 데이터로 정의되는 것이다. 정보 은닉은 객체의 표현과 연산자를 숨김으로써 외부의 접근을 막는 것을 의미한다. 이러한 특징은 소프트웨어 구성 요소들의 상호 의존성을 떨어트려 소프트웨어 신뢰도를 높이고 수정 가능성을 열어 둔다.

1. 실습 방법

**RangeArray**

주어진 문제는 RangeArray라는 클래스를 만드는 것이다. RangeArray는 배열의 인덱스를 사용자가 임의로 지정할 수 있다. 예를 들어 RangeArray a의 인덱스를 -5부터 5로 설정했다면, a[-5], a[-4], a[-3], a[-2], a[-1], a[0], a[1], a[2], a[3], a[4], a[5]를 가진다.

**구현**

RangeArray가 필요한 기능은 다음과 같다.

* Array a( ), b ( ): RangeArray 선언
* length: RangeArray의 크기 반환
* [ ] (left value): RangeArray에 원소 삽입
* [ ] (right value): RangeArray의 값을 반환
* print(): RangeArray의 모든 내용 출력

RangeArray는 Array 클래스에서 기능을 상속받아 구현한다. 그러면 Array의 기능을 수정해 RangeArray에서도 사용할 수 있다. Array의 기능을 이용해 RangeArray를 다음처럼 구상할 수 있다.

**Array a( ), b ( )**

a와 b에 시작 인덱스와 끝 인덱스를 입력 받고, 그 크기만큼 Array를 선언해 RangeArray와 길이가 같은 Array를 구현할 수 있다.

**length: RangeArray의 크기 반환**

len에 생성 때 입력받은 a, b를 통해 계산한 길이를 저장했다가 반환한다.

**[ ]**

입력받은 i의 값이 RangeArray가 아닌 Array에서는 몇 번째 값인지 계산한 뒤, Array의 [ ] 연산을 실행한다.

**print()**

수정할 필요 없이 Array의 print()를 그대로 사용한다.

1. 기타

**C++**

C++은 Standard C를 사용한 프로그래밍 언어이다. C의 문법과 기능을 사용하고, 객체 지향 프로그래밍이 가능해 많이 쓰이며, 특히 메모리에 대한 직접 접근이 자주 필요한 분야 (예: 게임 프로그래밍)에서 자주 볼 수 있다.

**C++의 기본 문법**

C++의 입출력은 iostream을 통해 이루어진다. cout을 통해 printf처럼 출력할 수 있다. 이때 %d 없이 변수명을 이용해 숫자를 출력할 수 있다. 그리고, \n 뿐만 아니라 endl을 이용해 줄바꿈을 할 수 있다.

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){  int num = 123;  cout << “Hello World” << endl <<num;  } |
| Hello World123 |

cin을 이용해 입력을 받을 수 있다.

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  int num;  cin >> num;  cout << num;  } |
| 123 <<입력  123 <<출력 |

C++은 class를 사용할 수 있다. class는 다음과 같이 선언할 수 있다.

|  |
| --- |
| class human {  protected:  int age = 16;  public:  void age(){  cout<<age<<endl;  }  } |

class로부터 변수를 상속받을 수 있다.

|  |
| --- |
| class student : public human  {  protected:  int score = 100;  public:  void ageAndScore(){  cout<<age<<endl<<score<<endl;  }  } |

1. 참고 문헌

“C++ Language.” *Cplusplus.com*, [www.cplusplus.com/doc/tutorial/](http://www.cplusplus.com/doc/tutorial/).

Smith, Ricard. *Standard for Programming Language C++*. ISO/IEC, 2019.

낭종호 교수님. *프로그래밍언어 강의자료*. 서강대학교 컴퓨터공학과, 2020