게임공학과 20203324 최종현

2주차 리포트

프로그래밍이란? 컴퓨터에게 특정 작업을 수행하도록 지시하는 일련의 과정이다. 컴퓨터는 컴퓨터가 이해할 수 있는 언어로 이해할 수 있는 명령어를 작성해야 한다. 이 명령어들을 모아둔 것이 '프로그램' 이다.

프로그래밍을 할 때 자료형을 구분해야 하는 이유는 무엇일까?

1. **컴퓨터는 각 자료형에 따라 필요한 메모리 공간을 할당**한다.
2. 컴퓨터는 모든 자료를 이진수로 저장한다.
3. 정수형보다 실수형이 좀 더 적은 메모리 공간을 차지한다.
4. 코드의 가독성이 올라가기 때문에 유지보수성이 향상 된다.

자료형이 필요한 이유는 무엇일까?

1. int(4바이트) , char(1바이트) , double(8바이트).
2. 자료형이 없다면 모든 변수가 같은 크기의 메모리를 차지해야 하기 때문에 비효율적이다.
3. **데이터의 목적과 의미를 명확하게 표현**해야 한다.
4. 나이는 정수 int , 키는 소수점을 포함해야 해서 float or double을 사용해야 한다.
5. 자료형이 다르면 연산 결과가 달라질 수 있다.
6. int a = 5/2; 의 결과와 float b = 5.0/2; 결과가 다르다.
7. 문자, 정수, 실수 등으로 저장한다. 이것들을 컴퓨터에 저장하기 위해서 자료형이 필요하고 구분하는 것이다.
8. 자료형은 메모리 관리, 데이터의 사용 목적 의미 명확, 오류 방지, 연산의 정확성 , 성능 최적화를 위해 꼭 필요하다.

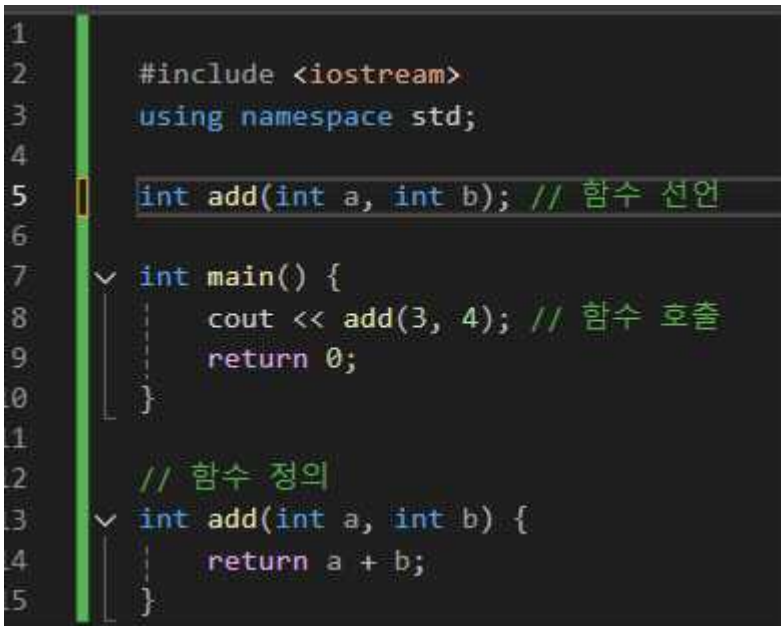
개발자들은 데이터를 보호하기 위해서 어떤 방법을 생각했을까

1. 강의시간에 구조체/클래스라는 개념을 배웠다.
2. 구조체 : 여러 개의 데이터 타입을 하나로 묶어주는 사용자 정의 데이터 타입이다.
3. 클래스 : 객체를 만들기 위한 설계도 데이터와 기능을 하나로 묶어 관리하는 사용자 정의 데이터 타입이다.
4. 객체지향 프로그래밍에서 가장 중요한 개념이다.
5. 클래스는 상속과 다형성을 활용할 수 있는 장점이 있다.
6. 클래스는 전체적으로 보면 데이터에 가깝다고 할 수 있다.
7. 보안을 위해서 사용하는 이유는 **접근 지정자 때문이다.**
8. 접근 지정자가 있기 때문에 데이터를 보호할 수 있고 효율적으로 관리할 수 있다.

함수란 무엇일까?

1. 함수 : 특정한 작업을 수행하는 코드의 묶음
2. 입력(매개변수)을 받아서 연산을 수행한 뒤 결과값을 반환하는 역할을 한다.
3. 왜 사용할까? -> 1.코드의 재사용 같은 코드를 반복해서 작성할 필요가 없다. 2. 가독성 증가 코드를 더 쉽게 이해하고 관리 가능
4. 함수는 크게 정의 , 선언 , 호출로 나눌 수 있다.
5. 함수 정의 : 함수의 내용(몸체)을 작성하는 것을 의미한다.
6. 구체적으로 어떤 작업을 수행할지 구체적으로 구현해야 한다.

7. 함수 선언 : 컴파일러에게 이런 함수를 만들었고 사용할 것이다 라고 미리 알려주는 것이다.
8. 함수 정의 전에 선언하면, 순서에 관계없이 함수 사용 가능하다.



9. 함수 호출 : 정의된 함수를 실제로 실행하는 것이다.
10. 호출 시에는 함수 이름과 인수(Argument)를 사용해야한다.

실습결과

```cpp
#include <iostream>


using namespace std;

int main()
{
    int a = 10; // a 라는 변수를 선언 그 안에 10이라는 값을 저장
    int b = 21;
    cout<<"Hello World!\n"<< endl;
    cout<<"Hello \t World!"<< endl; // /t 를 사용하면 띄어쓰기 기능이 있다.
    cout<<"Hello "<< "World!" << endl;
    cout<<"Hello"<< "World!" << endl; // 연속으로 출력


    cout << 5 << endl; // 상수 5의 값을 출력
    // 어디에 저장될까? -> CPU의 레지스터 라는 곳에서 처리가 된다고 함


    cout << "a:"<< a << endl; // 정수 a 의 값을 출력한다.
    cout << "a:10" << endl; // a:10 이라는 문자열을 출력한다.
    cout << "a"<<":10" << endl;  // a :10 이라는 문자열을 출력한다.

    // 출력은 동일하게 보이지만 출력을 어떤 방식으로 하는지에 따라서 달라진다.
```

```
Hello World!

Hello    World!
Hello World!
HelloWorld!
5
a:10
a:10
a:10
```

```cpp
cout << a + 5 << endl; // 그저 a에 5를 더한 값을 보여줘
cout << "a + 5" << endl; // a+5 문자열이 출력된다.

cout << a + b << endl;
cout << a - b << endl;
cout << a * b << endl;
cout << a % b << endl;
```

```
31
-11
210
10
```

```
cout << a / (double)b << endl; // 실수로 나옴 메모리에는 int 4 바이트를
                               //  할당하지만 계산할 때는 일시적으로 실수로 사용된다.
                               // 이렇게 바꾼 b를 저장하려면 어떻게 해야할까?
// 먼저 나눈값을 저장할 변수를 선언해야 한다.
// 실수를 저장하기 위해서니 double로 선언하겠다.
double newb = a / (double)b;
cout << newb << endl;
b = newb;
cout << b; // 0이 출력되는 모습을 볼 수 있음
// 왜 0이 나올까?
// b 는 처음에 int로 선언되었기 때문에 실수는 처리하지 못한다.
// 그렇기 때문에 소수점은 출력하지 못하기 때문에 0 으로 출력이 된다.
// 소수점을 출력하기 위해서는 double로 다시 선언을 해야한다.
```

```
0.47619
0.47619
0
```