

객체지향프로그래밍

HW03

C211123 이준선

2023년 3월 16일

1 코드

```
1 #include <iostream>
2 #include <ctime>
3 #include <cstdlib>
4 #include <random>
5
6 /// @brief uniform distributed random even number generator
7 class EvenRandom
8 {
9 public:
10     /// @brief seed 설정
11     EvenRandom();
12
13     /// @brief 랜덤 짝수 리턴
14     /// @return random even number
15     int next();
16
17     /// @brief low와 high 사이의 랜덤 짝수 리턴
18     /// @param low minimum value
19     /// @param high maximum value
20     /// @return [low, high] 사이의 랜덤 짝수
21     int nextInRange(const int low, const int high);
22
23 private:
24     std::random_device random_device;
25     std::mt19937 mt_engine;
26     std::uniform_int_distribution<int> uniform_distributor;
27 };
28
29 EvenRandom::EvenRandom()
30     : mt_engine(random_device())
31 {
32     // 최대 분포를 RAND_MAX의 절반으로 설정 (결과값에 * 2를 하기 위함)
33     uniform_distributor.param(std::uniform_int_distribution<int>::
34         param_type(0, RAND_MAX / 2));
35 }
36
37 int EvenRandom::next()
38 {
39     return uniform_distributor(mt_engine) * 2;
```

```

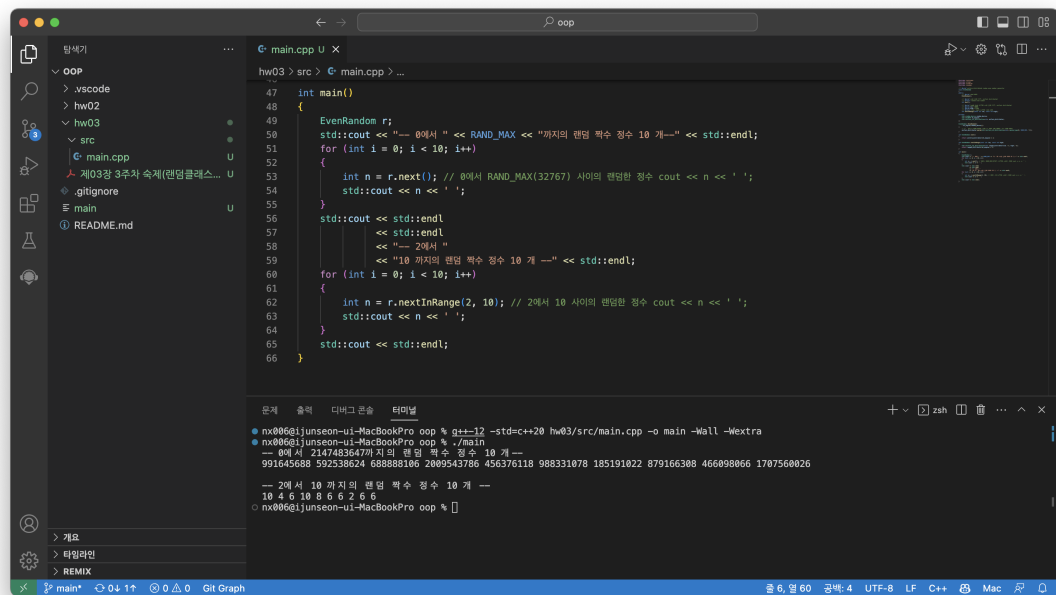
39 }
40
41 int EvenRandom::nextInRange(const int low, const int high)
42 {
43     std::uniform_int_distribution<int> ranged_distributor(low / 2,
44     high / 2);
45     return ranged_distributor(mt_engine) * 2;
46 }
47
48 int main()
49 {
50     EvenRandom r;
51     std::cout << "-- 0에서 " << RAND_MAX << "까지의 랜덤 짝수 정수 10
52     개--" << std::endl;
53     for (int i = 0; i < 10; i++)
54     {
55         int n = r.next(); // 0에서 RAND_MAX(32767) 사이의 랜덤한 정수
56         std::cout << n << ' ';
57     }
58     std::cout << std::endl
59     << std::endl
60     << "-- 2에서 "
61     << "10까지의 랜덤 짝수 정수 10 개 --" << std::endl;
62     for (int i = 0; i < 10; i++)
63     {
64         int n = r.nextInRange(2, 10); // 2에서 10 사이의 랜덤한 정수
65         std::cout << n << ' ';
66     }
67     std::cout << std::endl;
68 }

```

Listing 1: main.cpp

코드 1의 raw format 버전은 본 문서 아래에 담았음.

2 결과



The screenshot shows a VS Code editor window with a C++ file named `main.cpp` open. The code defines a `main` function that uses a random number generator to produce two sets of 10 random numbers. The first set is generated using `rand()` and the second set using `randInRange(2, 10)`. The output of the program is displayed in the terminal window at the bottom, showing the execution command and the resulting random numbers.

```
47 int main()
48 {
49     EvenRandom r;
50     std::cout << "— 0에서 " << RAND_MAX << "까지의 랜덤 짝수 10 개—" << std::endl;
51     for (int i = 0; i < 10; i++)
52     {
53         int n = r.next(); // 0에서 RAND_MAX(32767) 사이의 랜덤한 짝수 cout << n << ' ';
54         std::cout << n << ' ';
55     }
56     std::cout << std::endl;
57     << std::endl;
58     << "— 2에서 "
59     << "10까지의 랜덤 짝수 10 개 —" << std::endl;
60     for (int i = 0; i < 10; i++)
61     {
62         int n = r.nextInRange(2, 10); // 2에서 10 사이의 랜덤한 짝수 cout << n << ' ';
63         std::cout << n << ' ';
64     }
65     std::cout << std::endl;
66 }
```

```
nx006@junseon-ui-MacBookPro oop % g++-12 -std=c++20 hw03/src/main.cpp -o main -Wall -Wextra
nx006@junseon-ui-MacBookPro oop % ./main
— 0에서 2147483647까지의 랜덤 짝수 10 개 —
991645688 592538624 688888106 2009543786 456376118 988331078 185191022 879166308 466098066 1707560026
— 2에서 10까지의 랜덤 짝수 10 개 —
10 4 6 10 8 6 6 2 6 6
nx006@junseon-ui-MacBookPro oop %
```

그림 1: 실행 결과

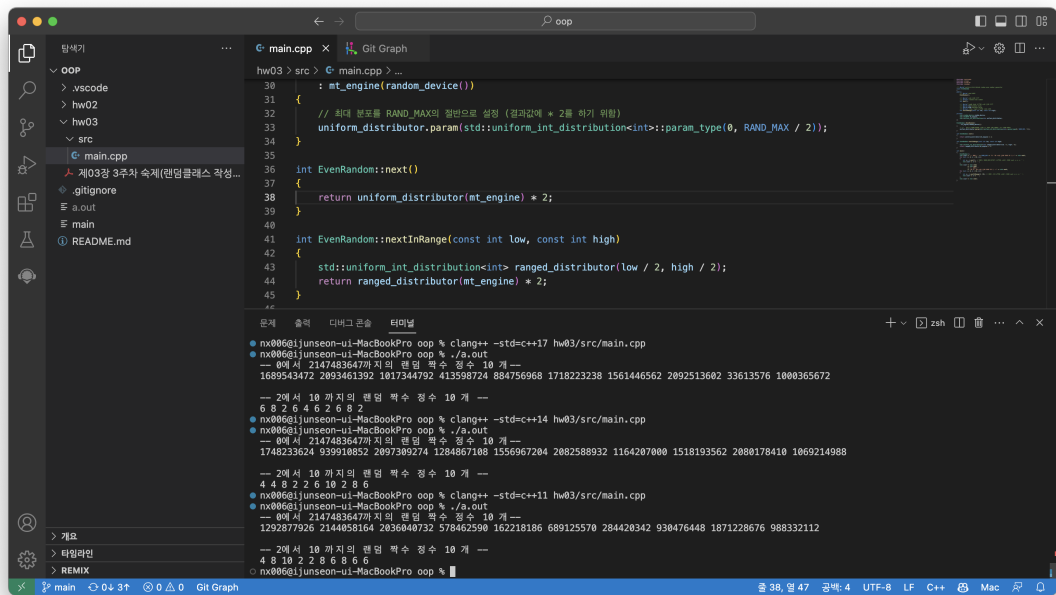


그림 2: Clang result

3 참고

random 헤더는 C++11부터 도입됐기 때문에 컴파일러 옵션에 `-std=c++11`, 혹은 그 이상의 버전을 붙여주어야 한다.

그림 1에서 최댓값이 2147483647로 설정됐다. 과제 명세서에 적힌 값과 다른데, 컴파일러 혹은 C++ 버전에 따라 `RAND_MAX`의 값은 달라질 것이므로 이를 굳이 수정하지는 않았다. C++ 버전은 C++20, 컴파일러는 g++-12 버전을 사용했다.

사실 컴파일러를 g++이 아닌 clang으로 바꾸고, C++ 버전을 11, 14, 17로 바꾸어도 과제 명세서에 적힌 값은 얻을 수 없었다(그림 2). 이는 애초에 `cstdlib`에서 매크로로 정의된 `RAND_MAX`의 값이 `0x7fffffff`로 int 범위의 최댓값(약 21억)으로 정의됐기 때문이다.

4 코드(RAW)

코드 1의 raw format 버전을 아래 담았다.

```
#include <iostream>
#include <ctime>
#include <cstdlib>
#include <random>

/// @brief uniform distributed random even number generator
class EvenRandom
{
public:
    /// @brief seed 설정
    EvenRandom();

    /// @brief 랜덤 짝수 리턴
    /// @return random even number
    int next();

    /// @brief low와 high 사이의 랜덤 짝수 리턴
    /// @param low minimum value
    /// @param high maximum value
    /// @return [low, high] 사이의 랜덤 짝수
    int nextInRange(const int low, const int high);

private:
    std::random_device random_device;
    std::mt19937 mt_engine;
    std::uniform_int_distribution<int> uniform_distributor;
};

EvenRandom::EvenRandom()
    : mt_engine(random_device())
{
    // 최대 분포를 RAND_MAX의 절반으로 설정 (결과값에 * 2를 하기 위함)
    uniform_distributor.param(std::uniform_int_distribution<int>::param_type(0, RAND_MAX / 2));
}

int EvenRandom::next()
{
    return uniform_distributor(mt_engine) * 2;
}

int EvenRandom::nextInRange(const int low, const int high)
{
    std::uniform_int_distribution<int> ranged_distributor(low / 2, high / 2);
    return ranged_distributor(mt_engine) * 2;
}

int main()
{
    EvenRandom r;
    std::cout << "-- 0에서 " << RAND_MAX << "까지의 랜덤 짝수 정수 10 개--" << std::endl;
```

```

for (int i = 0; i < 10; i++)
{
    int n = r.next(); // 0에서 RAND_MAX(32767) 사이의 랜덤한 정수 cout << n << ' ';
    std::cout << n << ' ';
}
std::cout << std::endl
        << std::endl
        << "-- 2에서 "
        << "10 까지의 랜덤 짝수 정수 10 개 --" << std::endl;
for (int i = 0; i < 10; i++)
{
    int n = r.nextIntInRange(2, 10); // 2에서 10 사이의 랜덤한 정수 cout << n << ' ';
    std::cout << n << ' ';
}
std::cout << std::endl;
}

```