**윈도우즈 프로그래밍 과제**

20234014 컴퓨터 공학과 이은정

1. using System;

2. using System.Collections.Generic;

3.

4. namespace \_5\_2\_과제

5. {

6. internal class Program

7. {

8. internal class MotherClass

9. {

10. public string data = "default\_data";

11. public override string ToString() => $"MotherClass: {data}";

12. }

13.

14. internal class childClass : MotherClass

15. {

16. public new string data = "child";

17. public int num = 0;

18. public MotherClass Clone()

19. {

20. return (MotherClass)this.MemberwiseClone();

21. }

22. public override string ToString() => $"childClass: {data}, num={num}";

23. }

24.

25. internal class childClass2 : MotherClass

26. {

27. public new string data = "child2";

28. public int num = 1;

29. public override string ToString() => $"childClass2: {data}, num={num}";

30. }

31.

32. static void Main(string[] args)

33. {

34. List<Object> list = new List<Object>() {

35. new childClass(), new childClass2(), new childClass2(),

36. new childClass(), new childClass(), new MotherClass()

37. };

38.

39. foreach (Object obj in list)

40. {

41. Console.WriteLine($"Object: {obj.ToString()}");

42. Console.WriteLine($"Type: {obj.GetType()}");

43. Console.WriteLine($"HashCode: {obj.GetHashCode()}");

44. Console.WriteLine(Console.ReferenceEquals(obj, list[1]));

45. Console.WriteLine(obj.Equals(list[0]));

46. Console.WriteLine($"ReferenceEquals(obj, list[1]): {Console.ReferenceEquals(obj, list[1])}");

47.

48. // 메서드 확인

49. if (obj.GetType() == typeof(childClass))

50. {

51. childClass c = (childClass)obj;

52. Console.WriteLine($"childClass num: {c.num}");

53. }

54. else if (obj.GetType() == typeof(childClass2))

55. {

56. childClass2 c2 = (childClass2)obj;

57. Console.WriteLine($"childClass2 num: {c2.num}");

58. }

59. else if (obj.GetType() == typeof(MotherClass))

60. {

61. MotherClass m = (MotherClass)obj;

62. Console.WriteLine($"MotherClass data: {m.data}");

63. }

64.

65. Console.WriteLine();

66. }

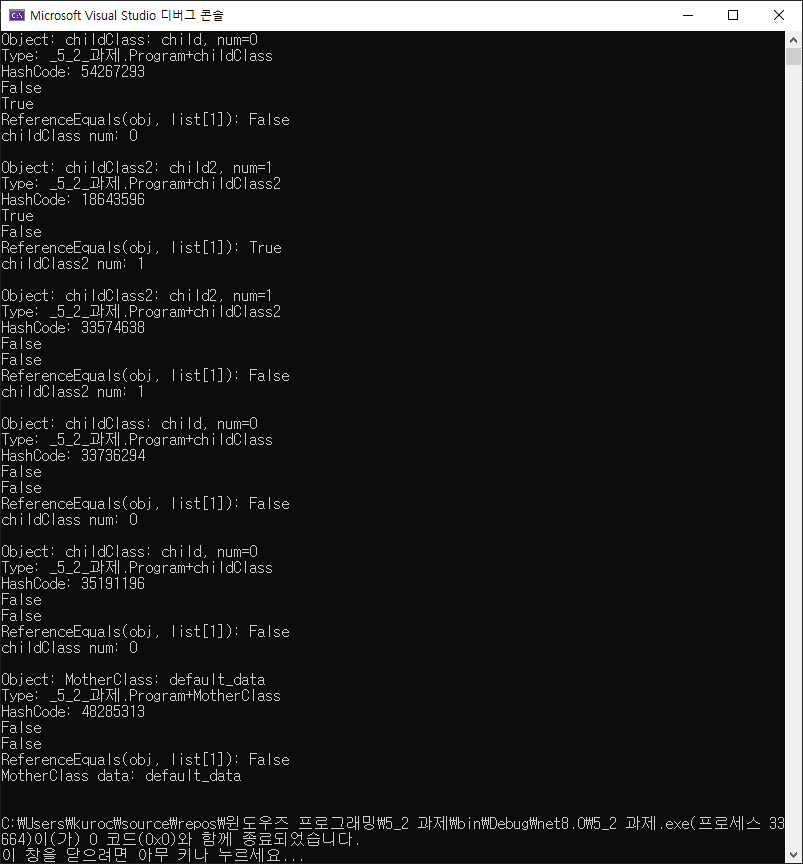
67. }

68. }

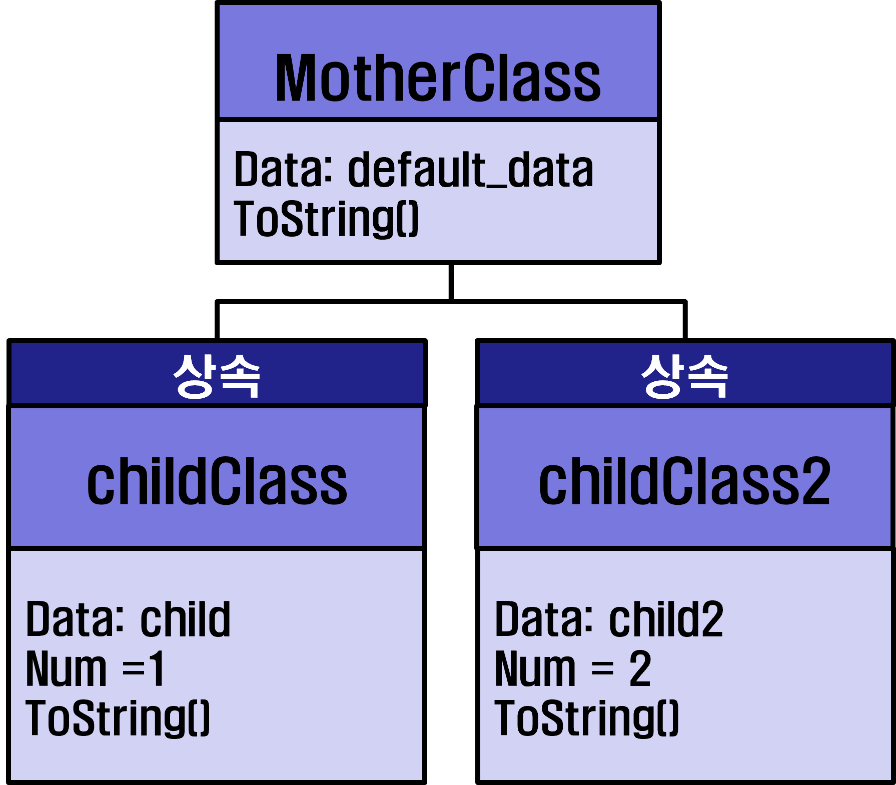
69. }

70.

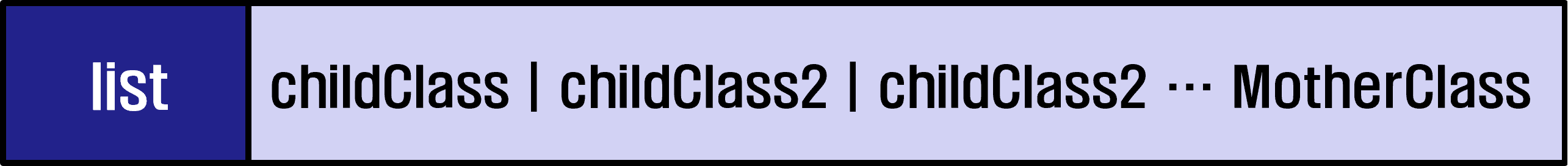
결과



코드 구조:



다음과 같은 구조로 클래스가 구성되어 있다.  
ToString()은 override를 하지 않을 시, 객체의 정보를 직관적으로 확인하기 어려워, 각 클래스에 맞게 override를 진행하였다.

이후, 아래와 같이 리스트에 클래스 인스턴스를 저장하였다.

ToString()은 각 객체가 가진 속성을 문자열로 출력한다. 이때, override를 진행하지 않을 경우 해당 객체는 namespace.클래스이름 과 같은 형식으로 출력된다. 따라서 이를 유용하게 사용하기 위해 override를 통해, 클래스의 구체적인 변수 속성까지 출력되도록 정의할 수 있다.  
예를 들어, 아래와 같이 출력 형식을 지정함으로써 각 객체의 내부 상태를 시각적으로 구분할 수 있다.

$"childClass2: {data}, num={num}";

GetType()은 객체의 실제 런타임 타입을 반환한다. 본 코드에서는 MotherClass를 상속한 두 개의 자식 클래스가 존재하고, 이들은 전부 Object 타입으로 리스트에 저장되어 있다. 따라서 각 객체의 정확한 타입을 확인하기 위해 GetType() 메서드를 사용하였다..

1. if (obj.GetType() == typeof(childClass))

2. {

3. childClass c = (childClass)obj;

4. Console.WriteLine($"childClass num: {c.num}");

5. }

6.

위와 같이 부모 타입인 Object나 MotherClass로 선언된 객체가 실제로는 childClass인지 childClass2인지 판단할 수 있다.

GetHashCode()는 해당 객체의 해시 코드를 정수 형태로 반환하는 메서드이다. 이 해시 값은 Dictionary나 HashSet과 같은 해시 기반 자료구조에서 객체를 빠르게 찾는 데 사용된다.

같은 객체는 동일한 해시 코드를 반환하기 때문에 빠른 검색이 가능하며, 실제로 위 코드의 실행 결과를 보면 각 객체는 서로 다른 해시 값을 출력하는 것을 확인할 수 있다.

Equals()는 두 객체의 값이 같은지를 판단한다.  
즉, 참조가 다르더라도 내부 데이터가 같다면 true를 반환할 수 있도록 override할 수 있다.  
아래 코드에서는 아래와 같이 list[0]과 비교하여 true 혹은 false를 출력하였다.

Console.WriteLine(obj.Equals(list[0]));

ReferenceEquals()는 두 객체의 참조가 정확히 같은지를 확인하는 메서드이다. 즉, 두 변수가 동일한 메모리 주소를 참조하고 있는지를 판단한다. Equals()와 달리 ReferenceEquals()는 override되지 않으며, 순수하게 객체 참조 비교에 사용된다.

아래 코드에서는 리스트의 각 요소와 특정 인덱스를 비교하여 동일한 인스턴스를 참조하는지를 확인하는 역할을 하였다.

Console.WriteLine(Console.ReferenceEquals ( obj, list[1] ));

emberwiseClone()은 객체를 복사하는 메서드이다. 객체 내부의 값을 그대로 복제한다. 해당 메서드는 protected로 선언되어 있기 때문에 외부에서 직접 호출할 수는 없고, 클래스 내부 메서드를 사용해 접근할 수 있다.

아래 코드에서는 MotherClassExtensions를 통해 리플렉션 기반으로 MemberwiseClone()을 호출해 복사본을 생성할 수 있다.

public MotherClass Clone()

{

return (MotherClass)this.MemberwiseClone();

}