

03_ジェネリックと例外処理

<スライド06>

GenericTest1.cs

```
public class GenericTest1 : MonoBehaviour
{
    void Start()
    {
        int a = 10;
        int b = 20;
        float c = 0.1f;
        float d = 0.2f;

        //格納されている値を確認
        Debug.Log("a:" + a + ",b:" + b + ",c:" + c + ",d:" + d);

        SwapValue<int>(ref a, ref b);
        SwapValue<float>(ref c, ref d);

        //格納されている値を確認
        Debug.Log("a:" + a + ",b:" + b + ",c:" + c + ",d:" + d);
    }

    //ジェネリック型
    private void SwapValue<T>(ref T a, ref T b)
    {
        T temp = a;
        a = b;
        b = temp;
    }
}
```

```
public class GenericTest2 : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
        Weapon weapon1 = new Weapon();
        Item item1 = new Item();

        weapon1.SetName("武器1");
        item1.SetName("アイテム1");

        ShowMessageWeapon(weapon1);
        ShowMessageitem(item1);

        ShowMessage<Weapon>(weapon1);
        ShowMessage<Item>(item1);
    }

    private void ShowMessageWeapon(Weapon w)
    {
        Debug.Log(w.GetName());
    }

    private void ShowMessageitem(Item i)
    {
        Debug.Log(i.GetName());
    }

    private void ShowMessage<T>(T obj) where T : Base
    {
        Debug.Log(obj.GetName());
    }
}
```

```
}  
}  
  
public class Base  
{  
    public string name;  
  
    public void SetName(string val)  
    {  
        name = val;  
    }  
  
    public string GetName()  
    {  
        return name + "を持っている。";  
    }  
}  
  
public class Weapon : Base  
{  
    public int Attack;  
}  
  
public class Item : Base  
{  
    public int count;  
}
```

<スライド07>

Test1.cs (問題)

```
public class Test1 : MonoBehaviour
{
    void Start()
    {
        int a = 10;
        float b = 0.1f;
        string c = "test";

        Save("aaa", a);
        Save("bbb", b);
        Save("ccc", c);
    }

    private void Save(string key , int value)
    {
        /*保存処理*/
    }

    private void Save(string key , float value)
    {
        /*保存処理*/
    }

    private void Save(string key , string value)
    {
        /*保存処理*/
    }
}
```

<スライド07>

Test1.cs (回答)

```
public class Test1 : MonoBehaviour
{
    void Start()
    {
        int a = 10;
        float b = 0.1f;
        string c = "test";

        Save<int>("aaa", a);
        Save<float>("bbb", b);
        Save<string>("ccc", c);
    }

    private void Save<T>(string key , T value)
    {
        /*保存処理*/
    }
}
```

<スライド08>

GenericTest3.cs

```
public class GenericTest3 : MonoBehaviour
{
    void Start()
    {
        GenericClassTest<int> test1 = new GenericClassTest<int>();
        test1.SetObject(1);
        Debug.Log(test1.GetObject());

        GenericClassTest<string> test2 = new GenericClassTest<string>();
        test2.SetObject("テスト");
        Debug.Log(test2.GetObject());
        GetComponent<Rigidbody>();
    }
}

public class GenericClassTest<T>
{
    private T _item;

    public void SetObject(T item)
    {
        _item = item;
    }

    public T GetObject()
    {
        return _item;
    }
}
```

<スライド11>

ErrTest.cs

```
public class ErrTest : MonoBehaviour
{
    public GameObject gObj;

    // Start is called before the first frame update
    void Start()
    {
        try {
            gObj.SetActive(false);
        } catch (System.Exception e) {
            Debug.LogWarning($"GameObjectをアタッチしてください。 : {e}");
        }
    }

    // Update is called once per frame
    void Update()
    {
        try {
            gObj.SetActive(false);
        } catch (System.Exception e) {
            //何も表示しない
        }
    }
}
```