

型情報

リフレクションAPI

私たち人間は、特定のクラス(型)の情報がほしい時
開発者のドキュメントを参照して仕様や公開されているメソッドなどを知ることができます

アプリケーションレベルでは稀なケースですが
開発環境などを作成する場合、クラスの型情報を実行時に必要とすることがあります
そこで、.NET は実行時にプログラムが型情報をえる手段を提供しています

型の物理的な情報を得るには **リフレクション API** が鍵を握ります
リフレクション API は **System.Reflection** 名前空間に定義されるクラスです
リフレクション API のクラスを全てここで解説することはできませんが
型情報を取得するまでの重要な操作方法のみを説明します
リフレクション API の全容を知りたい方は、Microsoft のドキュメントを参照してください

まず、型情報の基盤となるクラスが **System.Type** クラスです
このクラスは、次のように定義されています

```
public abstract class Type : MemberInfo, IReflect
```

このクラスを用いて型に問い合わせることができます
Type クラスを用いれば、必要とするほとんどの型情報を取得できるでしょう

では、Type クラスのインスタンスはどのように作ればよいのでしょうか？
Type クラスのコンストラクタは protected なので、外部からアクセスできません
Type のインスタンスを取得する方法はいくつかあるのですが
C# において一番簡単な方法は **typeof** 演算子を用いることです

typeof(type)

type には、問い合わせたい型を指定します
これで、指定した型の情報を格納する Type の参照が返ります

```
using System;

class Test {
    static void Main() {
        Type[] t = {
            typeof (Type) ,
            typeof (Int32) ,
            typeof (Console)
        };
        foreach (Type tmp in t) Console.WriteLine(tmp);
    }
}
```

このプログラムは typeof 演算子を用いて問い合わせたい型の Type を取得します
Type クラスの ToString() は情報を格納しているクラスの名前を返します
このプログラムを実行すると、次のようになるでしょう

System.Type
System.Int32
System.Console

Type クラスには、型情報を取得する様々なプロパティが存在します
目的に応じてドキュメントを調べて、取得したい情報のプロパティを利用するとよいでしょう

問い合わせたい型の Type インスタンスを得る方法はこれだけではありません
全ての型のルートクラスとなる Object クラスは
型情報を返す **GetType()** メソッドを提供します

```
public Type GetType();
```

このメソッドは、隠蔽されない限り全ての型がサポートすることになります
型の名前ではなく、インスタンスから情報を取得したい場合はこれを用いるとよいでしょう

```
using System;

class Test {
    static void Main() {
        CheckType('A');
        CheckType(50);
        CheckType("Kitty on your lap");
    }
    static void CheckType(Object obj) {
        Type t = obj.GetType();
        Console.WriteLine(t);
    }
}
```

Test クラスの CheckType() メソッドは Object 型の参照を受け取ります
そして、GetType() メソッドを用いて Type 型のインスタンスを作成し
型の名前をコンソールに出力します

System.Char
System.Int32
System.String

先ほどと違うのは、型名を指定するという静的な方法ではなく
オブジェクトから動的に Type を取得するという点です

さらに、文字列で型を指定して動的に Type を取得する方法もあります
この方法であれば、ユーザーに情報がほしいクラス名を指定させることも可能です
文字列で型情報を得るには静的な **Type.GetType()** メソッドを使います
Type クラスの GetType() 静的メソッドは、クラス名で型情報を取得できるのです

```
public static Type GetType(string typeName);
```

これ以外にも GetType() メソッドはオーバーロードされていますが
この場では最も簡単な、文字列だけを受け取る上のメソッドを紹介します

typeName には、型の名前をあらわす文字列を指定します
.NET ライブラリは言語に依存しないので、int などのエイリアスでは取得できません
名前空間も含めた、型のフルネームを指定しなければなりません

```
using System;

class Test {
    static void Main(string[] args) {
        if (args.Length == 0) return;

        Type[] t = new Type[args.Length];
        for (int i = 0 ; i < args.Length ; i++) t[i] = Type.GetType(args[i]);
        foreach (Type tmp in t) Console.WriteLine(tmp);
    }
}
```

このプログラムは、コマンドライン引数で得られた文字列の型情報を表示します
例えば、test.exe 実行ファイルを生成したとして、次のように動作します

```
C:\dir_name>test System.Char System.Console
System.Char
System.Console
```

この方法を用いれば、プログラムが明示的に型情報を要求しなくても
ユーザーに文字列として取得したい型情報を入力させることができます

メンバ情報

前記したように、リフレクション API を網羅して説明することはできませんが
ここでは、簡単に Type からクラスのメンバ情報を取得する方法を説明しましょう

リフレクション API は System.Reflection 名前空間に属するクラスのことです
フィールドやメソッドの情報を物理レベルで問い合わせる方法を提供します

クラスのメンバ全体の情報を取得するには、リフレクション API の一つ
System.Reflection.MemberInfo クラスを使います
このクラスは System.Type クラスの親クラスに属するクラスでもあります

```
public abstract class MemberInfo : ICustomAttributeProvider
```

このクラスを用いれば、メンバの情報を取得することができます
メンバの情報を取得する手段は様々ですが、この場では最も簡単な方法である
Type クラスのインスタンスメソッド GetMembers() メソッドを用いる方法を使いましょう

```
public MemberInfo[] GetMembers();
```

このメソッドは、Type オブジェクトが持つ型が保有するメンバの情報を
MemberInfo 型のオブジェクトの配列として返します
MemberInfo はそれぞれ1つのメンバを表すので、配列の数はそのままメンバの数となります
ここで返される列挙されたメンバ情報は、基底クラスの情報も含まれます

```
using System;
using System.Reflection;

class Test {
    static void Main(string[] args) {
        if (args.Length == 0) return;

        Type t = Type.GetType(args[0]);
        MemberInfo[] mi = t.GetMembers();
        foreach (MemberInfo tmp in mi) Console.WriteLine(tmp);
    }
}
```

このプログラムは、コマンドライン引数で調べたいクラスの名前を指定すると
そのクラスのメンバ情報がコマンドラインに列挙されます

しかし、メンバをフィールドとメソッドに区別し
それぞれに特殊化した情報をより確実に取得し、操作したい場合
フィールドは **System.Reflection.FieldInfo** クラスを用い
メソッドは **System.Reflection.MethodInfo** クラスを用います

```
public abstract class FieldInfo : MemberInfo
public abstract class MethodInfo : MethodBase
```

それぞれのインスタンスを取得するには MemberInfo 同様に
Type クラスの **GetFields()** メソッドと **GetMethods()** を使います
GetFields はフィールド情報を列挙した FieldInfo オブジェクトの配列
GetMethods はメソッド情報を列挙した MethodInfo オブジェクトの配列を返します

```
public FieldInfo[] GetFields();
public MethodInfo[] GetMethods();
```

これらのオブジェクトを取得すれば、より精密のメンバ情報を得られます
この他にもイベントやプロパティ、コンストラクタを取得する方法もありますが
それらについても、ほとんど同様の手法でプログラムすることができます
詳しくは Microsoft のドキュメントを参照してください

```
using System;
using System.Reflection;

class Test {
    static void Main(string[] args) {
        if (args.Length == 0) return;

        Type t = Type.GetType(args[0]);
        FieldInfo[] field = t.GetFields();
        MethodInfo[] method = t.GetMethods();

        if (field.Length != 0)
            Console.WriteLine("-----Fields-----");
        foreach (FieldInfo tmp in field) Console.WriteLine(tmp);

        if (method.Length != 0)
            Console.WriteLine("-----Methods-----");
        foreach (MethodInfo tmp in method) Console.WriteLine(tmp);
    }
}
```

コマンドライン引数から型名を指定すれば
その型のフィールドとメソッド情報をコンソールに出力します

typeof

typeof(type)

指定した型の Type オブジェクトを返します
C# 固有のエイリアスを指定することも可能です

type - 型を指定します

[前のページへ](#)

[戻る](#)

[次のページへ](#)