

質問1 voidとはどんな型なのか？

回答

void “戻り値無し”を表わすプレースホルダであり、データ型ではない
*void is a placeholder indicating no return type.*¹⁾

*¹⁾著書「The Java Programming Language 4th edition」サンマイクロシステムズ社より

質問2 なぜコンストラクタは戻り値がないのに void が指定できないのか？

回答

メソッドとコンストラクタはその宣言の形やオーバーロードが可能であることから大変に類似している。しかし、次に示す定義によりこれらは明確に区別される。

メソッドの定義

修飾子 戻り値の型 メソッド名 (引数リスト)

```
{  
    本 体  
}
```

修飾子 public, private など

戻り値 メソッドが返す戻り値の型
値を返さない場合は void を指定

メソッド名 メソッドの名前

引数リスト 引数とその型の組からなるリスト
引数がない場合は空にする (void も指定しない)

本 体 処理コード

コンストラクタの定義

修飾子 クラス名 (引数リスト)

```
{  
    本 体  
}
```

修飾子 public, private など

引数リスト 引数とその型の組からなるリスト
引数がない場合は空にする (void も指定しない)

本 体 処理コード

コンストラクタの定義より、その宣言には戻り値の宣言を含めることができないことが分かる。この理由であるが、文献1で調べたところ関連する記述は見当たらなかった。次のサンプルを見てほしい。コンストラクタの宣言に戻り値の宣言を含めるとメソッドとして認識される。戻り値の型の宣言の有る／無しがこれらを明確に区別するための働きをしている。

```
class Test{
    public Test(){
        System.out.println("This is Constructor!!");
    }
    public void Test(){
        System.out.println("This is not Constructor!!");
    }
}
```

```
class Main{
    public static void main(String[] args){
        Test tt=new Test();
        tt.Test();
    }
}
```

>java Main

This is Constructor!!

This is not Constructor!!

-- Press any key to exit (Input "c" to continue) --