

コンストラクタ

コンストラクタとは、クラス名と同名で、戻り値のないメソッド。シグネチャに戻り値型を指定すると、コンストラクタとはみなされないので注意。

引数のないコンストラクタは、特別にデフォルト・コンストラクタと呼ばれる。

コンストラクタに `private` を指定すると、そのコンストラクタは呼び出しできないので、`new` でインスタンスを作成できないコンストラクタとなる。

ところで、クラスのフィールドメンバのデフォルトアクセシビリティは、`private` である。したがって、アクセシビリティを明示しないコンストラクタは `private` となり、インスタンスを作成できない。

つまり、通常のコンストラクタには `protected internal` 以上のアクセシビリティを明示しなければならない。

コンストラクタがひとつも明示的に定義されていないクラスは、コンパイラが何もしないデフォルト・コンストラクタを用意して、それが呼び出される。

ひとつでも引数のあるコンストラクタが明示的に定義されていると、コンパイラは暗黙のデフォルト・コンストラクタを作成しない。

その場合にデフォルト・コンストラクタが必要であれば、クラス作成者が明示的に用意する必要がある。

クラスの場合、暗黙に変数フィールドが初期化される。

その後、コンストラクタが呼び出される。コンストラクタの最初の文が、スーパークラスのコンストラクタの明示的な呼び出しでない場合、コンパイラはスーパークラスのデフォルト・コンストラクタを呼び出そうとする。

スーパークラスにデフォルト・コンストラクタが存在しなければ、コンパイル・エラーになる。

static コンストラクタ

`static` コンストラクタの存在意義は、フィールドの `static` な値を初期化するためだけ。

`static` コンストラクタは、プログラム中で最初に `static` な値にアクセスが行われたときに呼び出される。

したがって、`static` コンストラクタはユーザが呼び出す必要はない。というか、暗黙に `private` であるので、呼び出せない。

アクセス修飾子を指定する必要もない。

static クラス

`static` クラスは `static` メンバの入れ物としての役割。

クラスと構造体の違い

クラスは参照型。構造体は値型。参照型は実体はヒープに置かれ、実体への参照によってアクセスする。

構造体は、すべてが暗黙に sealed。つまり、継承できない。
また、引数のあるコンストラクタを定義できない。

アクセシビリティ

アクセシビリティが適用される範囲は 2 つある。

1. 名前空間

public: アセンブリの外から利用可能であることを示す。

internal: アセンブリ内からのみ利用可能であることを示す。

2. クラス内（つまりフィールド）

種類は、5 種類。下記フィールドのアクセシビリティを参照。

クラスを超えてアクセスが可能かどうか、という分類。

名前空間のデフォルトのアクセシビリティは internal。

フィールドのアクセシビリティ

フィールドは、デフォルトで private。

フィールドの取れるアクセシビリティは 5 種類。

public -> protected -> internal -> protected internal -> private。