



コンストラクタほか

応用編1日目

# コンストラクタ

- **コンストラクタ** ... インスタンス生成時一度だけ呼び出される特殊なメソッド
- メソッド名はクラス名は同一
  - Ex. クラス名が**Person**なら、コンストラクタ名も**Person**となる。
- **戻り値はない**
- 引数付き、引数なしの両方を指定することが出来る。

# コンストラクタの呼び出し

```
Person p1 = new Person();
```

```
Person p2 = new Person("太田隆",29);
```

```
class Person
{
    // コンストラクタ(引数なし)
    public Person()
    {
        ...
    }
    // コンストラクタ(引数あり)
    public Person(string name, int age)
    {
        ...
    }
    ...
}
```

- コンストラクタは複数定義可能（コンストラクタの**オーバーロード**）
- コンストラクタが複数あれば、引数の種類によって呼び出し先が変わる

# コンストラクタのないクラス

- コンストラクタは定義された場合だけ呼び出される
- コンストラクタは定義しなくても構わない
- 定義されていない場合はシステムが暗黙の裡に何もしないコンストラクタ（**デフォルトコンストラクタ**）を定義し、呼び出していると考える

# コンストラクタのthis

```
Person p1 = new Person();
```

番号は実行順序

```
class Person
```

```
{
```

```
// コンストラクタ(引数なし)
```

```
public Person():this("名無し",0)
```

```
{
```

```
...
```

```
③
```

```
}
```

```
// コンストラクタ(引数あり)
```

```
public Person(string name, int age)
```

```
{
```

```
...
```

```
}
```

```
...
```

```
}
```

name:名無し  
age:0

①

②

- コンストラクタの後に**this**を付けることにより、他の引数付きコンストラクタを呼び出すことが出来る。
- 他のコンストラクタを呼び出した後で、そのコンストラクタの処理に入る。

# ガーベージコレクタ①

- **SampleEx102**参照

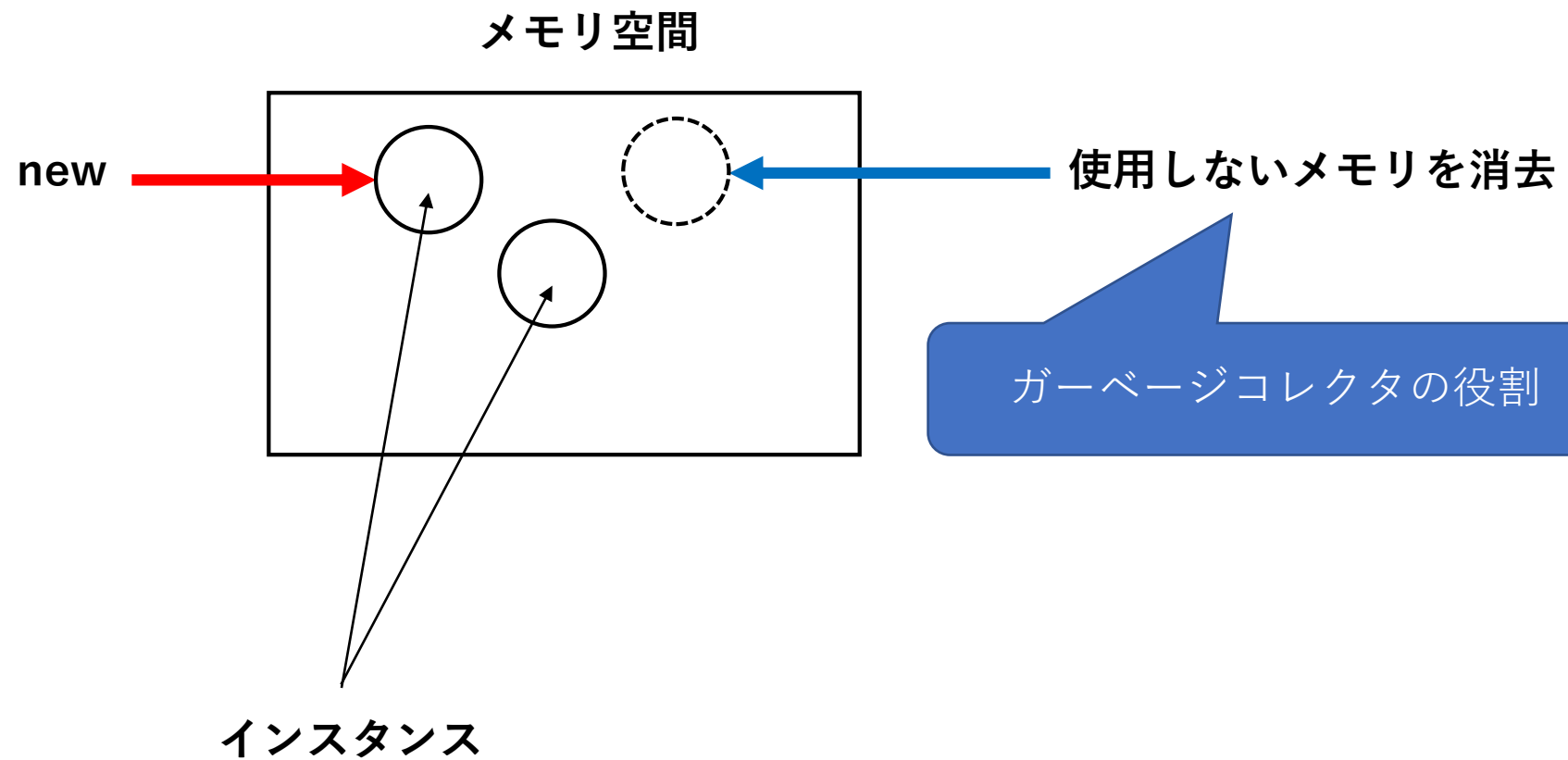
- ガーベージ … 英語でゴミを意味
- ガーベージコレクタ … ゴミ集め
- newを用いてインスタンスを生成 → メモリの中に蓄積
- インスタンス は**必要が無くなればメモリの中から消去**される
- この処理を行うのが、ガーベージコレクタの役割

## ガーベージコレクタ②

- ガベージコレクタは原則的に自動的に動き出す
- プログラムからガーベージコレクタの起動を促すことが可能  
ガーベージコレクタの起動を促す

```
GC.Collect();
```

# ガーベージコレクタ③





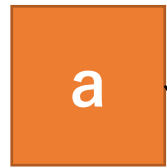
# メモリの開放

- メモリから参照がとれるとガーベージコレクションの解消になる
- 参照を外すには、**null(ヌル)**を代入するなどの手段がある
- nullとは、英語で、「**何もない**」ということの意味する
- nullを代入するとその変数が、どんなオブジェクトも参照していない状態にすることを意味

```
a = null;
```

# nullによる解放の注意点

① `Foo a = new Foo();`



① Fooクラスのインスタンス生成し、Foo型の変数aで参照

② `Foo b = a;`



② ①で生成したインスタンスをFoo型の変数bでも参照

③ `a = null;`



③ aの参照をnullにして解除してもbからの参照は残る

- このような場合、ガーベージコレクタによってオブジェクトが消去されることはない。
- 消去するためには**全ての参照をクリア**する必要がある。

# メモリ容量の取得

- 以下の処理によって、現在、使用されているメモリがどれだけあるのかを調べることがきる

```
GC.GetTotalMemory(false);
```

# ジェネレーション

- **new**でインスタンスが生成されるメモリ領域を**ヒープ**と言う
- ヒープに生成されるオブジェクトは、その生成期間に応じて、3種類に分けられる
- ジェネレーションはオブジェクトの生存期間の短いものから、0～2に分けられる
- 下の方法で、特定のジェネレーションのデータに対してのガーベージコレクタを発動させることができる

```
GC.Collect(ジェネレーションの数);
```

# デストラクタ

- コンストラクタとは逆の概念
- オブジェクトが破棄される時に呼び出される特殊なメソッド
- 引数を必要とせず、アクセス修飾子も省略する
- メソッド名の前に～（**チルダ**）をつけることにより定義
- コンストラクタと違い明示的に呼び出すことはない

デストラクタの定義

～(クラス名) ()