第二回　オブジェクト指向の基本　～　クラスとインスタンス

1. そもそも「オブジェクト」って？

オブジェクト指向の定義は前回書いたとおり、**「処理の対象をオブジェクト単位で分割し、オブジェクト同士のメッセージのやり取りにより処理を記述する」**というものです。ここで重要なのは**「オブジェクト」**と**「メッセージ」**というキーワードなのですが、今回は初めに「オブジェクト」の意味から説明を始めようと思います。

オブジェクトとは和訳するとズバリ「物」という意味で、定義が難しい言葉です。※1では、「物」とは何でしょうか？プログラム上において、**「オブジェクト（物）」とは自分を表すための要素を持っているもの**、と解釈します。

- 「四角形」という物は「高さ、幅」という要素を持っている
- 「人間」という物は「年齢、性別、名前、身長、体重」という要素を持っている※2
- 「パソコン」という物は「メーカー名、CPU、メモリ、HD、…」という要素を持っている
- 「おにぎり」という物は「大きさ、形、中身」という要素を持っている

と、このように身の回りにある「物」は大抵の場合何かしら「それ」自体を表すための要素を持っていると思います。そしてこの考えはプログラム上においてもそのまま置き換えることが可能なのです。

たとえば、「人間」を「物」として見た場合・・・	
	年齢　：　32 性別　：　女 名前　：　三沢　良子 身長　：　162cm 体重　：　56kg ←の「物」はこれらの要素を持っている！ ※あくまでイメージです。実在の人物とは一切関係ありません。

2. 簡単なオブジェクトの例

オブジェクト（物）はプログラム上では**「クラス（class）」**という単位で定義されます。実際に「四角形」というオブジェクトを例にして考えてみましょう。上にも書いたように、四角形は「高さ、幅」という要素を持っています。他にも「色」、「質量」、「材質」など色々要素はあるかもしれませんが、今回は話を簡単にするため四角形を現す要素は「高さ」と「幅」のみとします。では、実際に四角形クラスを書いて見ましょう。

```
public class Rectangle
{
        int height;
        int width;
}
```

1に書いた定義に従うと、四角形クラスを作るのに必要な記述はこれだけです。Rectangle（長方形）はクラス名、そして int 型変数の height, width がそれぞれ高さ、幅を表す四角形の要素というわけですね。この height, width を Rectangle クラスの**「フィールド（またはデータ）」**と呼びます。高さと幅の値が四角形の要素というわけですね。なお、最初につけた「public」という意味深なキー

ワードについては、今は説明しても関係ない場所なので、次々回の第四章までは「魔法の言葉」として無視することにします。それでは同様にして「人間」クラスも書いてみます。

```
public class Person
{
        int age;
        int sex;
        string name;
        int height;
        int weight;
}
```

これも簡単に解説すると、Person というクラスを定義し、要素（フィールド）として age（年齢），sex（性別），name（名前）※3，height（身長），weight（体重）を持たせているわけですね。

3. クラスとインスタンス

さて、上で二つのクラスを宣言しました。しかし！プログラムにおいてはクラスだけでは実際に使うことはできないのです。「クラス」はあくまでそのオブジェクト（物）を定義するための言わば設計図のようなものであり、ちょっと難しい言葉を使うと「抽象的な概念」に過ぎないのです。ではどうすればよいのでしょうか？

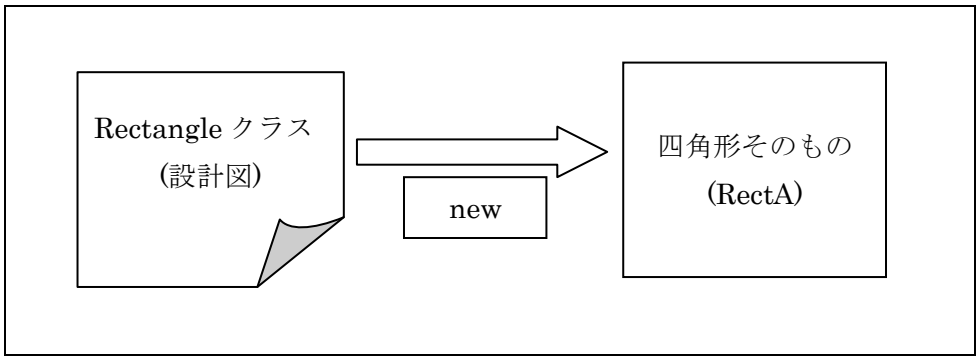
答えはシンプルで、クラスが「抽象的な概念」ならば「具体的な実物」を作ればいい！という話です。先ほど例に挙げた Rectangle クラス（四角形）を元に考えて見ましょう。四角形は「高さ・幅」という要素を持っている、という話をしました。ここに、「高さ7cm、幅5cm」の四角形を考え、Rectangle クラスを元にこの四角形を生成してみます。プログラムに書くとこんな感じです。

```
Rectangle RectA = new Rectangle();
```

何やらややこしい感じですね。でも、大抵のプログラミング言語ではこのように書くはずです。順を追って見てみます。まず、「RectA」というのは変数名で、その左についている「Rectangle」はその変数が何のクラスの実体であるかを示しています。わかりやすく言うと、Rectangle 型の RectA という変数を宣言したのだ、と思ってください。※4そして右側の new Rectangle()というのは、Rectangle クラスを元に実体を作ります、という意味を表しています。

「ちょっと待った、これじゃ”高さ7cm、幅5cm”の情報がないじゃないか」と思った方もいるでしょう。全く持ってその通りで、このままでは RectA の高さ、幅が未定義なんですね。ただ、今の段階でこの問題を解決する方法を説明しようとすると大変なので、とりあえず今回はこのまま行きます。もう少しだけ我慢してください。

さて、このようにして作られた RectA は「Rectangle クラスを元にして作られた実体」です。このような実体のことをクラスの**「インスタンス」**と呼びます。プログラムを構成するための部品（オブジェクト）をクラスとして設計しておき、new という演算子を用いてクラスのインスタンスを生成する。これがオブジェクト指向における基本となります。



4. おわりに

今回はここでおしまいとなります。この章では**「オブジェクト」「フィールド」「クラス」「インスタンス」**というキーワードが出てきました。これらは全て今後の基本となってくるので、今のうちにしっかりと定義を覚えておきましょう。

次回からは最初にスルーしていた「メッセージ」の説明と、「インスタンス」の生成についてもう少し詳しい解説をしていこうと思います。お楽しみに。

※1　：実はこれが「オブジェクト指向」とは何か？という問題をややこしくしている要因。物と言われると抽象的すぎてうまく説明できないことが多い。色んな解説サイトで様々な定義がされているが、それも実は人によって解釈がまちまちだったりする。

※2　：突き詰めれば人間を表す最大の要素は「DNA」。次回説明する「同一性（アイデンティティ）」とも関わる。

※3　：string は文字列を表す変数型のこと。C 言語にはないが、C#や Java 等の言語では標準で実装されている。

※4　：「Rectangle 型の変数」と言ったが、実は普段慣れ親しんでいる int や char、bool なども同じ。広義では int というのも「整数値クラス」となっており、int num というのは正確には int クラスのインスタンス num を宣言していることと同義。（ただし、純粋オブジェクト指向言語の場合。）詳しくは第五章で説明を行う。