

Ruby 初級者向けレッスン 第 21 回

okkez@Ruby関西, チカホリ@小波ゼミ

今回の内容

- オブジェクト指向
 - 歴史
 - 使うオブジェクト指向
 - 作るオブジェクト指向

今回のゴール

- オブジェクト指向に至った歴史を知る
- 使うオブジェクト指向を理解する
- 作るオブジェクト指向を知る

アンケート

- 構造化プログラミング
- オブジェクト指向
 - なにそれ？おいしいの？
 - 挫折した orz
 - それなりに使ってるよ
 - 達人級 (pragmatic)

歴史 - オブジェクト指向以前

- データ構造もへったくれも無い時代
 - 大昔
 - そこそこの規模(複雑さ)のソフトウェアで破綻
- データと処理を分ける時代
 - 構造化プログラミング -> ルーチンの抽象化
 - 大規模なソフトウェアにも対応できる (と言われていた)

歴史 - 構造化プログラミングの限界

- さらなる大規模(もっと複雑な)ソフトウェアの出現
 - データ構造と処理を分けるだけでは対応不可能に

歴史 - オブジェクト指向言語の登場

- Simula, Smalltalk
- C++, Java
- Python, javascript
- Ruby

使うオブジェクト指向

- 具体例
- 配布資料参照

使うオブジェクト指向

- 誰に何をして欲しいのか伝え、目的を達成する
- 既にあるものを活用する
- 組み込みクラス、標準添付ライブラリ

作るオブジェクト指向

- 世間で言われてる方のオブジェクト指向

オブジェクト指向とは？

- ソフトウェアの設計や開発において、操作手順よりも操作対象に重点を置く考え方。
- 関連するデータの集合と、それに対する手続き(メソッド)を「オブジェクト」と呼ばれる一つのまとまりとして管理し、その組み合わせによってソフトウェアを構築する。

私的オブジェクト指向

- 汎用の整理術
- 脳内スタック節約術
- DRY 実現のための考え方
 - Don't Repeat Yourself.

オブジェクトとは？

- 実際にある「もの」や「概念」
 - okkez, okkez のノートPC, などなど
- オブジェクトはデータと処理 (メソッド) をセットにしたもの
- オブジェクトは自分ができること (メソッド) を知っている

クラスとは？

- オブジェクトのうち共通の性質を持った「くくり」
 - 人、大学、パソコン、などなど
- インスタンスを作る「雛型」

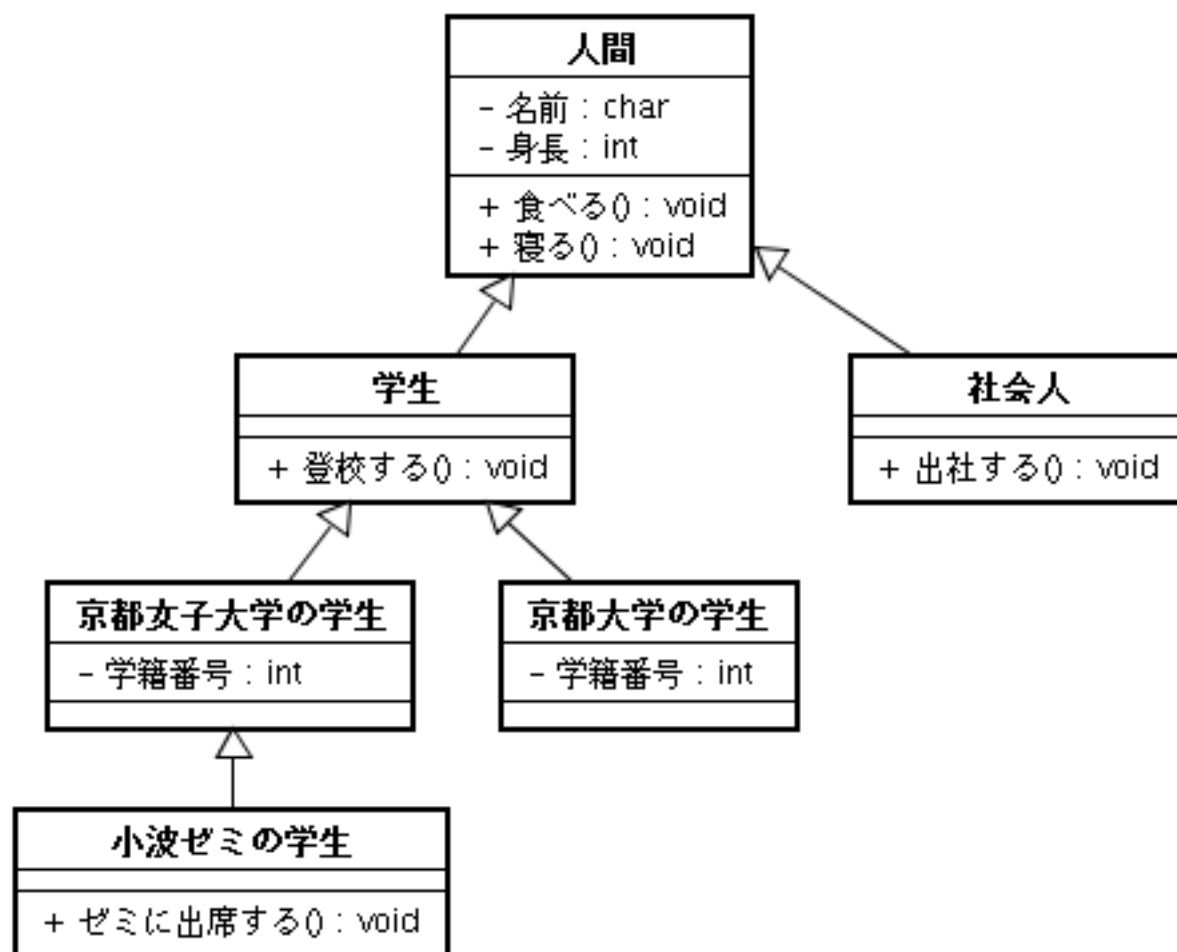
オブジェクト指向の三大要素

- 継承
- カプセル化
- ポリモルフィズム

継承

- 具体的なクラスは一般的なクラスの性質を引き継いでいる
 - 世の中のものをより一般的にくることを抽象化
 - 世の中のものをより具体的にくることを具象化

継承



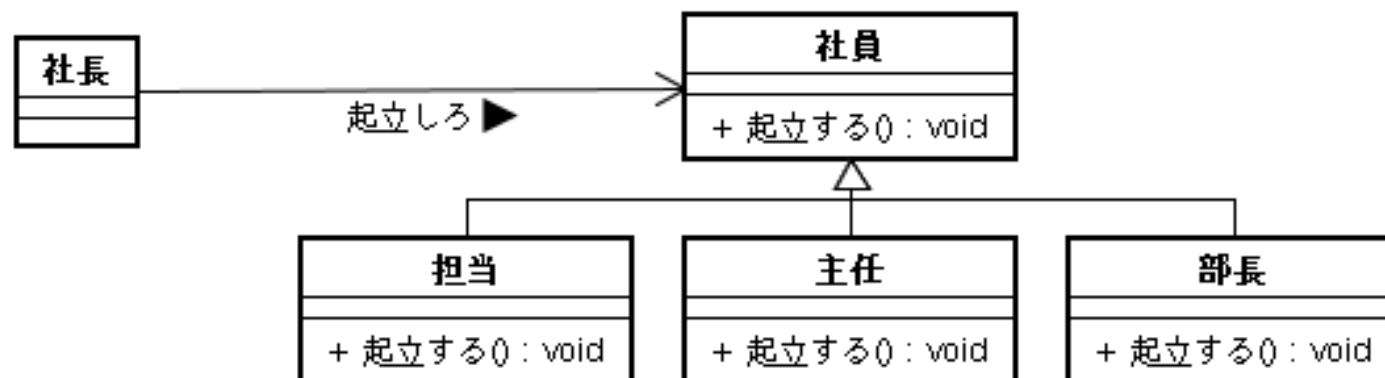
カプセル化

- 仕事 (メソッド) のやりかたは担当 (クラス) だけが知っていて、仕事を頼む人はその中身は知らない
 - オブジェクト内部の構造は外からはわからない -> 抽象データ型
 - オブジェクトの操作はメソッド経由で行う -> 変更が強くなる

ポリモルフィズム

- 究極奥儀
- 仕事を頼まれる側の種類が増えても、仕事を頼む方法は同じ

ポリモルフィズム



Ruby での継承

- 以下のように書くことができます

```
class SubClass < SuperClass  
end
```

Ruby での継承

```
csv.rb:12:  class IllegalFormatError < RuntimeError;  
            end
```

```
csv.rb:15:  class Cell < String
```

```
csv.rb:26:  class Row < Array
```

```
csv.rb:591: class StringReader < Reader
```

```
csv.rb:615: class IOReader < Reader
```

```
csv.rb:713: class BasicWriter < Writer
```

- この辺に出ています。

Ruby でのカプセル化

- インスタンス変数は外から変更できない
- 自動的にカプセル化は実現されている

Ruby でのポリモルフィズム

- Duck Typing
 - アヒルのように歩き、アヒルのようになくものはアヒルだ

Ruby でのポリモルフィズム

- IO, File, Tempfile, StringIO などなど
 - IO と File は継承関係にある
 - Tempfile は File クラスに処理を委譲している
 - StringIO は IO が持つメソッドを全て実装している

Ruby でのポリモルフィズム

- 実例
 - logger -> 出力先が IO オブジェクト (標準出力、ファイルなど)

まとめてコード例

- ちょっとだけコード例の説明をします

演習

- ウォーミングアップ
- 社長命令・起立!
- 給料はいくら?
- 取締役を追加
- ボーナスはいくら?

まとめ

- オブジェクト
 - 実際にある「もの」や「概念」
- クラス
 - オブジェクト共通の性質を持った「くくり」
- 継承
 - いろいろなクラス定義の共通化
- カプセル化
 - データ構造が変わっても、仕事の頼み方は同じ
- ポリモルフィズム
 - 仕事を頼まれる側の種類が増えても、仕事の頼み方は同じ

参考文献

- 初めてのプログラミング
 - <http://www.oreilly.co.jp/books/4873112923/>
- プログラミングRuby 第2版 言語編
 - <http://ssl.ohmsha.co.jp/cgi-bin/menu.cgi?ISBN=4-274-06642-8>
- たのしいRuby 第2版
 - http://shop.sbcr.jp/bm_detail.asp?sku=4797336617

今後の情報源

- 公式Webサイト
 - <http://www.ruby-lang.org/>
- Ruby リファレンスマニュアル刷新計画
 - <http://doc.loveruby.net/>
- 日本Rubyの会
 - <http://jp.rubyist.net/>
- Rubyist Magazine
 - <http://jp.rubyist.net/magazine/>
- okkez weblog
 - <http://typo.okkez.net/>