たい焼きでＪａｖａ

Ryoma Kondo
Tokyo Denki University at CPS Lab 2016

モノを作る設計図を作ろう

- 作りたいモノがどんなモノか考える
  - なにで作ろうか？
  - 何に使おうか？

- ちなみに…
  - 設計図をＪａｖａではクラス
  - モノをインスタンスやオブジェクトと呼びます

世の中のモノは全て属性と機能がある

- 属性って？
  - そのモノを説明するために必要なもの　　→　形容詞
  - 色や形状、状態などなど
  - Ｅｘ：[赤い] [８０ｇ] [新鮮な] りんご

- 機能って？
  - そのモノがする働きや動作　　→　動詞
  - Ｅｘ：りんごが[転がる]
    りんごが[落ちる]
    りんごが[腐る]

車の属性と機能は？

- 属性
  - タイプ「セダン」
  - 最高速度「時速１００Ｋｍ」
  - タイヤ「４個」
  - などなど

- 機能
  - 「走る」
  - 「ブレーキかける」
  - などなど

ゲームキャラクターの属性と機能は？

- 属性
  - 名前「スライム」
  - ＨＰ「１０」
  - ＭＰ「１」
  - 攻撃力「１」
  - 防御力「２」
  - 素早さ「１」
- 機能
  - 「攻撃する」
  - 「逃げる」

たい焼きの属性を考えてみよう

- 中身の具いろいろ
- 焼き加減も

- 最近はトッピングもあるらしい・・・

たい焼きの機能を考えてみよう

- 授業なのでしゃべるたい焼きにしましょう
  - 実際に食べて確かめられないので
- たい焼きが[しゃべる]　＝　自己紹介する
- 全国のたい焼き共通のキャッチフレーズも決めましょう

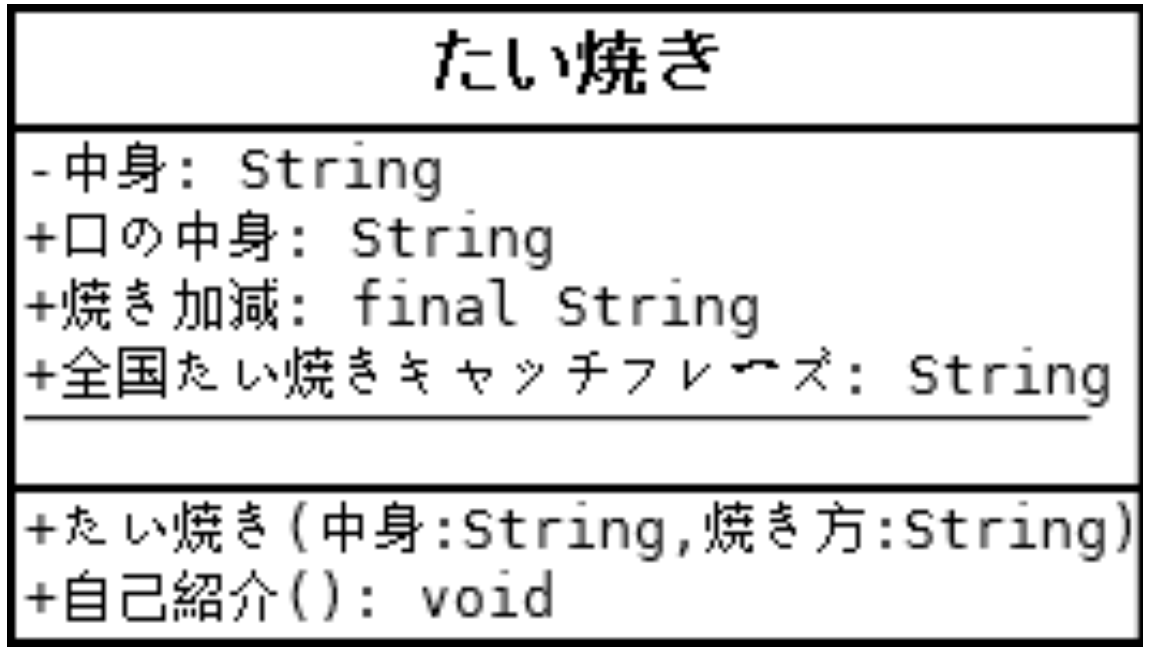
><(((‘< 「ぼくの中身はｘｘｘで、焼き加減はｘｘだよ！」

「キャッチフレーズ」

たい焼きの設計図（クラス）を作ろう

- 属性
  - 中身
  - 焼き加減
  - 口の中身
  - キャッチフレーズ
    （これは全たい焼き共通）

- 機能
  - モノを作る機能＝モノの名前と同じ
    これは設計図についてる＝コンストラクタという
  - 自己紹介する



モノの名前

モノの属性

モノの機能

クラス図

クラス図を元にたい焼きクラスを作る

- クラスの書き方

public class{
}

- コメント行は無視して良い

```
public class たい焼き {

    //================ 属性 ================

    //-------------------------------------



    //================ 機能 ================

    //-------------------------------------
}
```

たい焼きクラスの属性を書こう

- たい焼きの中身は外から
  分らないし変えられない
- 口の中身は外から
  分るし変えられる
- 焼き加減は外から
  分るけどいじれない
- キャッチフレーズは
  みんな共有

```
//================ 属性 ==================

//外から見えない
private          String    中身;

//外から見える
public           String    口の中身;

//外から見えるけど変えられない
public final     String    焼き加減;

//外から見えて、全てのたい焼きが共有している
public static    String    全国たい焼きキャッチフレーズ = "おいしいよ！";

//---------------------------------------
```

たい焼きクラスの機能を書こう１

- 新しいたい焼きを作る機能

１．中身・焼き加減を受け取る

２．自分自身の属性とする
　　thisが自分自身

```
//================== 機能 ====================

//新しいたい焼きを作る(newしたときに呼ばれる)
public たい焼き(String 中身, String 焼き加減) {
    this.中身     = 中身;
    this.焼き加減 = 焼き加減;
    return;
}
```

３．returnで作ったたい焼きを呼び出し元に返す
　　たい焼き入れ = new たい焼き(中身, 焼き加減);

たい焼きクラスの機能を書こう２

- 口に入ってると喋れない
- 中身について喋る
- 自分自身の属性にはthis.でアクセスできる

```
//しゃべる
public void 自己紹介(){

    //口に何か入ってたら喋れない
    if(this.口の中身 != null){
        System.out.println("><(((( '< 「モゴモゴ」");
        return;
    }

    System.out.print("><(((( '< 「ぼくの中身は" + this.中身 + "で、焼き加減は" + this.焼き加減 + "だよ！.
    System.out.println("「" + たい焼き.全国たい焼きキャッチフレーズ + "」");
    return;
}
```

たい焼きを作るシーンを作ってみよう

- シーンも、よくよく考えるとモノの一種
  →クラスで表現できる

- プログラムのスタート地点のMain関数を持つシーンを作る

```
public class たい焼きを作ってみる {

    //実行はmainメソッドからされる
    public static void main (String args[]){


    }

}
```

たい焼きクラスでたい焼きを作ろう

```
//実行はmainメソッドからされる
    public static void main (String args[]){

        //まず、たい焼きを入れる器を用意する
        たい焼き myたい焼き一号;

        //new で新しいたい焼きを作って器に入れる
        myたい焼き一号 = new たい焼き("あんこ", "カリカリ");

        //たい焼きができました！自己紹介してもらいましょう
        System.out.println("「1号さん自己紹介お願いします
        myたい焼き一号.自己紹介();
```