プログラミング基礎演習I

柴田祐樹

東京都立大学　情報科学科

2022年度

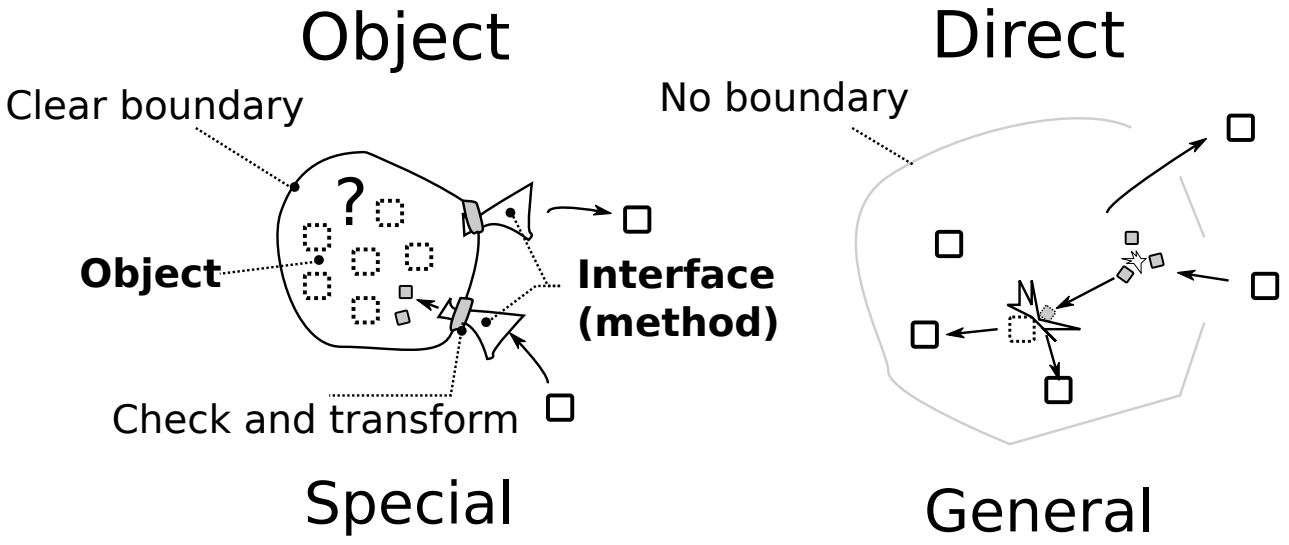
BPP第12回

予定

- オブジェクト指向
- クラスとオブジェクトの基本
- 補間法（Interpolation method）
- 課題解説

BPP第12回

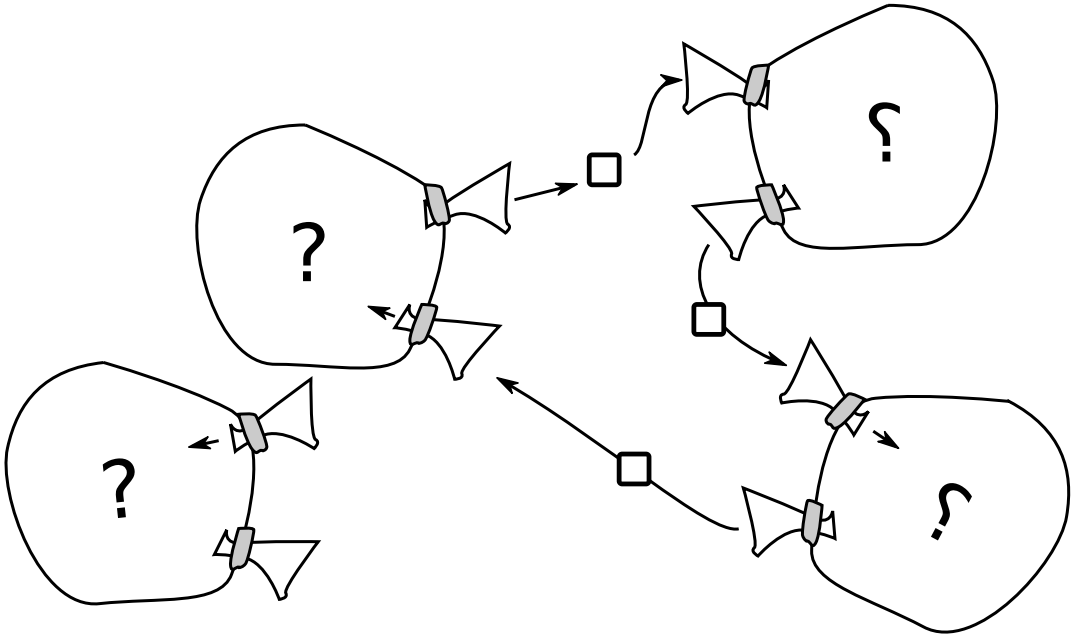
オブジェクト指向



- 内部を隠蔽して使いやすく
  - 例: 掃除機を使うときスイッチ（interface）を押す
  - モータの回転（内部状態）を気にしている人はいないだろう

BPP第12回

オブジェクト指向



- 同じものをたくさん作成したいという需要

2022年度 東京都立大学 情報科学科 3/17

BPP第12回

オブジェクト指向

- 設計図を定義: class
- 設計図からオブジェクトを生成 -> 変数に格納
- インタフェースによりオブジェクトを操作

BPP第12回

オブジェクト指向

- class（設計図，雛形）の定義，オブジェクトの生成，インタフェースの利用の基本（実演）
- インタフェース: 関数で実装，内部状態: 変数で実装
- 内部状態の変数: self. で参照．self はオブジェクトの仮の名前

```
 1: class myClass:
 2:     def __init__(self): # 初期化関数．内部状態の変数等を定義して初期化
 3:         self.x = 0
 4:     def add(self):
 5:         self.x += 1
 6:     def get(self):
 7:         return self.x
 8:
 9: a = myClass() # オブジェクトを生成し a に格納，__init__ を呼び出し
10: a.add()      # インタフェース add を利用.
11: a.add()
12: b = a.get()  # インタフェース get を利用
13: print(b)
```

2022年度 東京都立大学 情報科学科 5/17

オブジェクト指向

- 例2（実演）: 最大値，最小値を記憶するクラス Stats
  - 初期化: 最大値，最小値の変数を定義して初期化
  - インタフェース: join(self, x): self はオブジェクト，x が新しい値
  - getMax, getMin を値を得るインタフェースとして実装

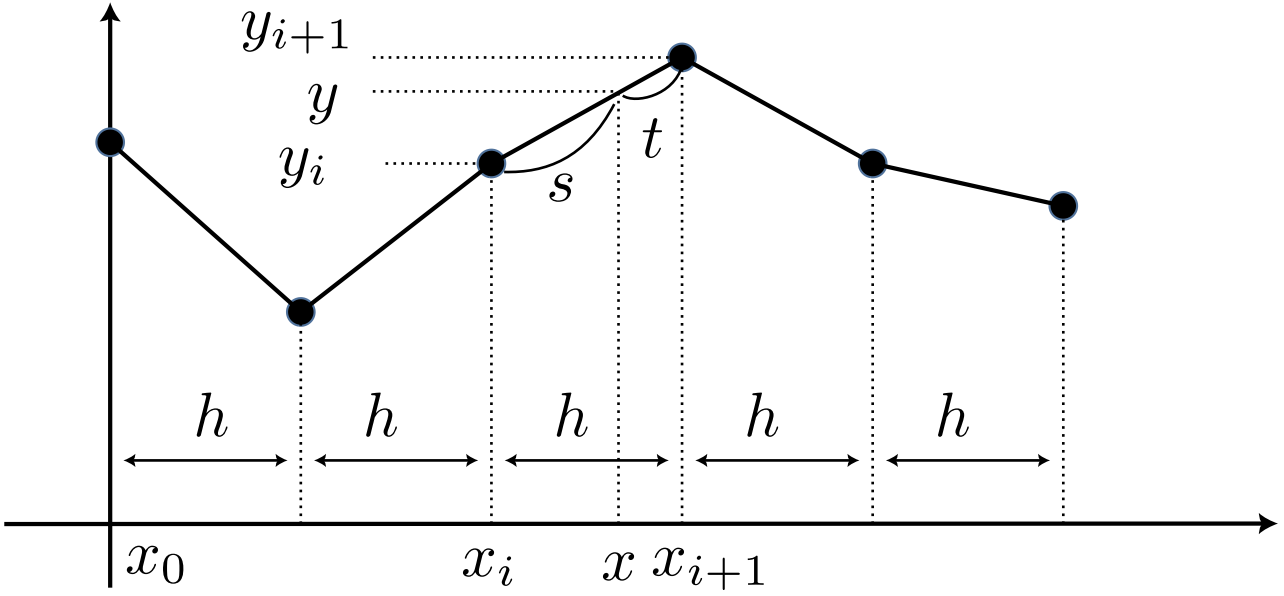
オブジェクト指向

- 練習課題（各自実践）: Stats に平均と合計を計算する機能と，それら値を取得するインタフェースを実装
- 名前の例: getAverage, getSum
- 計算する場所: 合計は join が呼ばれる毎に，平均は join が呼ばれた回数を記録し，getAverage が呼ばれたときに和の値から計算

BPP第12回

補間

- 少ないデータ量で完全な関数を近似する方法
- $i = \lfloor x/h \rfloor$: 格子の数を計算，小さい最寄りの整数へ丸める
- $s = (x - x_i)/h, t = 1 - s, y = y_{i+1}s + y_i t$

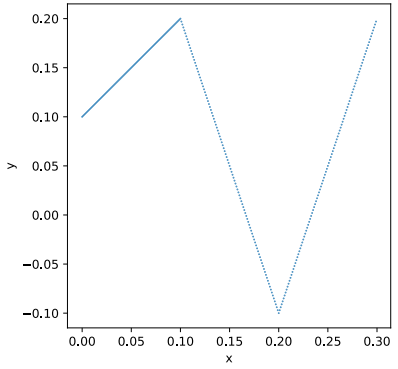


2022年度 東京都立大学 情報科学科 8/17

BPP第12回

補間

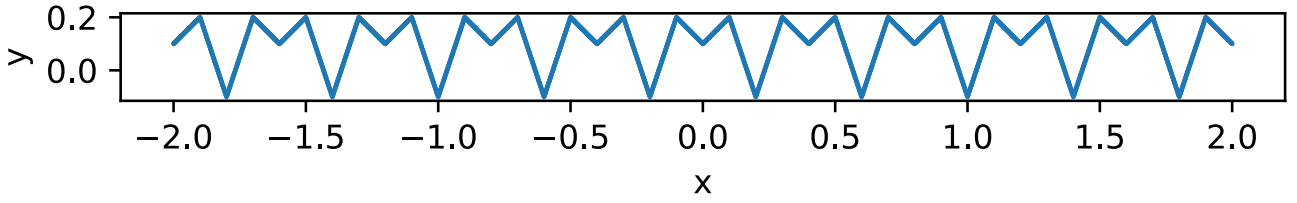
- 補間の実演:
  - $x = (0, 0.1, 0.2, 0.3)$
  - $y = (0.1, 0.2, -0.1, 0.2)$
  - $h = 0.1$, $\lfloor a \rfloor$ には math.floor(x)を利用, import math で導入
  - 補間関数: $f(x), x \in [0, 0.3)$
  - 描画する点: $x_i = \Delta x i, y_i = f(x_i), \Delta x = 0.001$



BPP第12回

補間

- 周期補間の計算（各自実践）：剰余演算（%）を使い，周期的に補間するように変更
- 格子の両端の添字 i_1, i_2 を計算後，周期的になるよう修正した添字 i'_1, i'_2 を計算
- $\hat{i}_1 = i_1 \% N$：あまりを計算 => $\hat{i}_1 < 0$ の可能性
  $i'_1 = (\hat{i}_1 + N) \% N$：正に修正して再びあまりを計算，飛び出した分を切り捨て（ $\hat{i}_1 > N$：ありえない）



BPP第12回

補間

- class: クラス（雛形）を定義
- __init__: 変数の初期化および予約を行う特別な関数
- self: メンバ変数の参照に利用
- . : メンバ参照演算子
- def: メンバ関数を定義

BPP第12回

課題

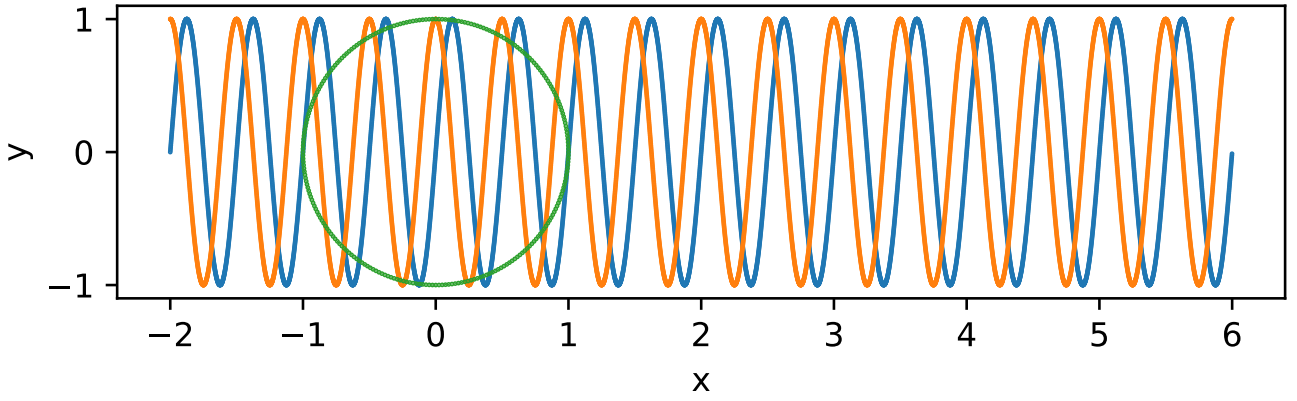
- 後述の課題を記載した ipynb 形式のファイルを提出
- 提出場所: kibaco の「第12回課題」
- 期限: 2023年1月18日24時（JST）
- 一つの課題は一つのセル内で完結すること
- どの課題かわかるようにコメントをセルの最初に記載すること
- 最後のセルに，授業の感想を記載

課題

- 課題1（6点）：高速で高精度な sin, cos 計算クラス myMath の設計
  - 初期化関数 __init__(self, N, M): オイラー法で計算した sin, cos をリストに格納: (sin, cos について: introduction-trigonometric-f.pdf), で計算
  - オイラー法: $[0, 2\pi)$ の区間を NM 等分して計算， M 点ずつ間引いて N だけリストに保存（ $N = 128, M = 16384$ など）
  - 与えられた引数 θ の $\sin\theta, \cos\theta$ を周期補間で計算する関数 sin, cos をそれぞれ定義
  - 描画して動作を確認，12th-hint.png を参考にコードを構築
  - 事前の計算は高精度に，実際の利用時は低負荷になるよう設計

課題

- 描画の例:
  - $\sin(4\pi t), \cos(4\pi t), t \in [-2, 6]$ を描画
  - オイラー法では $[0, 2\pi)$ のみを計算したが，この範囲外の引数についても正しく計算していることを確認
  - 間引き: 剰余演算を利用（ if i%M == 0: など）



課題

- 課題2（自由課題）：最適貨幣組の計算
  - 貨幣の組 $x = (1, 5, 10, 50, \dots)$ に対して，与えられた金額 A を表現する組み合わせのうち，枚数が最小のものを計算
  - 例: $A = a_1x_1 + a_2x_2 + \dots a_Nx_N$ を求める，ただし a_i, x_i はそれぞれ i 番目貨幣の枚数，金額である．
  - ヒント: 金額の大きい方から入る限り貨幣を用意，用意した貨幣の合計を A から引いて，次に小さい貨幣と処理を継続

課題

- 式の定義
  - $i = 1, 2, \ldots, N$：貨幣を識別する添字
  - x_i, a_i：i 番目の貨幣の金額，使う貨幣の枚数
  - a^A：$A = \sum_{i=1}^{N} a_i x_i$ を満たす中で $\sum_{i=1}^{N} a_i$ が最小となる a_i の組
  - $\bar{A} = 1/K \sum_{A=1}^{K} \sum_{i=1}^{N} a_i^A$：最小使用貨幣の期待値
  - $\hat{A} = \max\{\sum_{i=1}^{N} a_i^A | A = 1, 2, \ldots, K\}$：考えうるすべての金額の最小必要枚数の最大値
  - $\tilde{A} = \sum_{i=1}^{N} \max\{a_i^A | A = 1, 2, \ldots, K\}$：どんな金額でも払えるように所持するべき貨幣の組の合計枚数の最小値

BPP第12回

課題

- 課題3（自由課題）：最適貨幣組の計算による期待枚数の計算
  - 金額: $A = 1, 2, 3, \ldots, 10000$ について以下の場合を計算
  - 1: 日本円 1, 5, 10, 50, 100, 500, 1000, 5000, 10000 で計算
  - 2: 2進数 1, 2, 4, ..., 8192 で計算
  - 3: 会田先生の提案 1, 3, 10, 30, 100, 300, 1000, 3000, 10000 で計算
- 柴田の結果（ $\bar{A}$, $\hat{A}$, $\tilde{A}$ の順に記載）：
  - 日本円: 10, 20, 20
  - 会田先生: 8.4, 16, 20
- 会田先生の提案は貨幣の利用頻度を16%削減する．2進数は？