Stack・Queue
待ち行列理論のすすめ
【授業内演習あり】

情報科　飯島　涼

今週の内容

- データ構造入門
  - スタック(Stack)
  - キュー(Queue)
- データ構造を応用したシミュレーション
  - 現実世界寄り
  - ソフトウェア寄り

データ構造

プログラムの中で，データをどのようにして格納したり，取り出したりするかという構造のこと

代表的なデータ構造

- Stack
- Queue
- リスト
- 辞書

など

データ構造

プログラムの中で，データをどのようにして格納したり，取り出したりするかという構造のこと

代表的なデータ構造

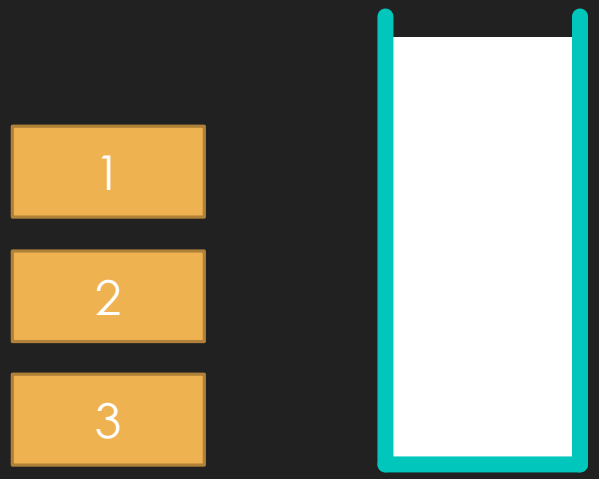
- Stack
- Queue
- リスト
- 辞書

など

スタック・キューとは

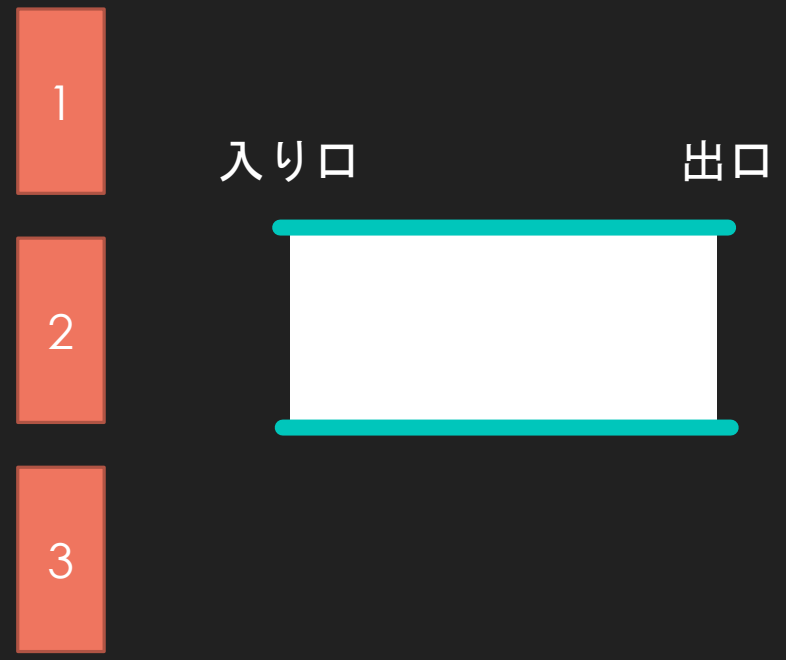
スタック (Stack, Last In First Out, LIFO)

- 1箇所からのみ出し入れができる箱



キュー (Queue, First In First Out, FIFO)

- 入り口，出口がそれぞれ別に1つずつ固定された箱



スタック・キューを何に使うのか?

スタック (Stack)

- メモリやレジスタの容量を節約したい計算
  - 逆ポーランド記法による計算
  - Stackを用いると，メモリを節約しながら簡単に実装できる

キュー (Queue)

- 待ち行列の解析（待ち行列理論）
  - リアルな待ち行列
  - 交通渋滞
- オンラインリアルタイム動画(Zoom，Skypeなど)のパケット受信

PythonによるStackの実演

- Stackは，Pythonのリストで表現可能
  - Stackにnumberを入れる作業 => append(number) 関数
  - Stackから取り出す作業　=> pop() 関数
  - popで取り出されるのは、末尾の数
    => 最後に入れた数から順に取り出される

```
1 # Stack
2
3 S = []
4
5 S.append(2)
6 print(S)
7
8 S.append(5)
9 print(S)
10
11 S.append(8)
12 print(S)
13
14 S.pop()
15 print(S)
16
17 S.pop()
18 print(S)
19
20 S.pop()
21 print(S)
22
```

実行結果

```
[2]
[2, 5]
[2, 5, 8]
[2, 5]
[2]
[]
```

PythonによるQueueの実演

- queueライブラリをインポートする
- Q = queue.Queue()で，Qをqueueとして扱うことができるようになる.
  - Queueにnumberを入れる作業　=> Q.put(number)
  - Queueの入り口から取り出す作業 => Q.get()

⇒ 次のページで例のコード

PythonによるQueueの使い方

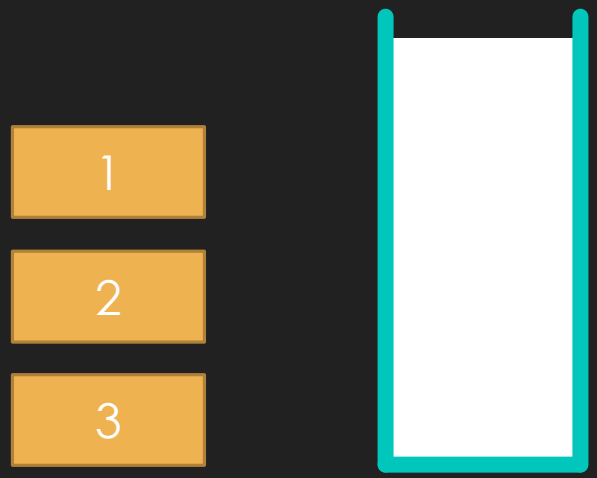
- Q.get()で先に追加したものから順に取り出していく
- empty()
  - Queueの中身が空かどうかを判定する

```python
import queue

Q = queue.Queue()

Q.put(2)
# [2]
Q.put(5)
# [2, 5]
Q.put(8)
# [2, 5, 8]

print(Q.get())
#[5, 8]
print(Q.get())
#[8]
print(Q.get())
#[]
```

```python
import queue

Q = queue.Queue()

Q.put(2)
# [2]
Q.put(5)
# [2, 5]
Q.put(8)
# [2, 5, 8]

while (not Q.empty()):
  print(Q.get())
```

```
2
5
8
```

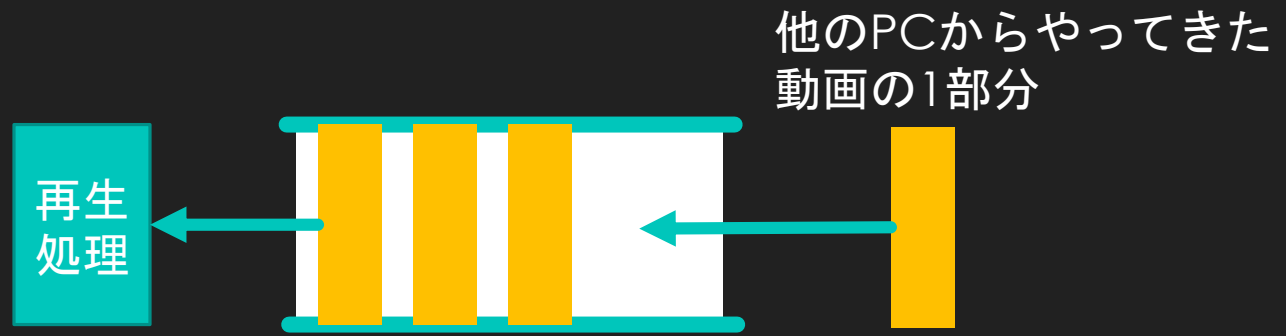
Stackのデータ構造でQueueのように最初に入れたものを取り出そうとするとどうなる？



- 取り出すのに時間がかかる
  - リストの活用で、pop(number)でnumber目のリストの要素が取り出せることを言ったが、この操作には取り出すのに時間がかかっている。
  - アプリやゲームをスムーズに動かすためには、利用用途に応じたデータ構造を適切に選ぶ必要がある

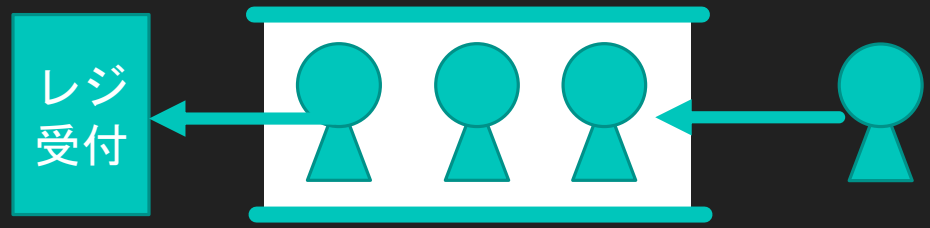
Queueの利用用途

- 待ち行列の解析（待ち行列理論）
  - リアルな待ち行列
  - 交通渋滞
- オンラインリアルタイム動画(Zoom，Skypeなど)のパケット受信



Queueの利用用途

- 待ち行列の解析（待ち行列理論）
  - リアルな待ち行列
  - 交通渋滞



- オンラインリアルタイム動画(Zoom，Skypeなど)のパケット受信

同じ現象として解析が可能になる

シミュレーション実験

- 実際に交通渋滞や，客の到着を予測するのに使用されるシミュレーションをやってみる

  - 関連分野:
    - 数学（統計）
    - 心理学
    - 待ち行列理論
    - ネットワーク解析
    - 交通渋滞

客の到着シミュレーション

- シミュレーションの仮定
  - 流行りの廃れたタピオカ屋
  - 1秒あたりの客の到着確率を，p=0.05とする.
  - 客は，ランダムに1人ずつ到着する (一緒の来店は1人とカウント)

- プログラム上の規定
  - random 関数を用いて、客がランダムに到着する様子をシミュレーションする
  - [1秒経過すること=for文の1回分のループ] と置き換える

- 導きたいもの
  - 最後の客が到着してから、次の客が到着するまでにどのくらいの時間が経過するか? (=客の到着時間間隔)
  - 客の到着時間間隔 < サービスを提供する時間　となると行列が発生し、レジの増員・効率化・機械化が求められる

【授業内演習】客の到着シミュレーション

以下の要求にしたがってプログラムを書いてみてください。

新しくタピオカ屋を開店するために、原宿におけるタピオカ屋の客の到着をシミュレーションすることにしたRは、以下の要件で客の到着をシミュレーションすることにした。

- 1秒あたりの客の到着確率は，p=0.05であることがわかっている.
- 客は，ランダムに1人ずつ到着する (一緒の来店は1人とカウント、整数時間ぴったりにしか来ないと考えてよい)
- for文が1回まわるプログラムの動作を、1秒の経過とみなして、TIME_END=60秒シミュレーションする。
- 客の到着間隔(最後の客が到着してから、次の客が到着するまでの時間)を変数など(time_dif)で管理し、リストに格納していく(time_dif_list など)。
- time_dif_listの要素をヒストグラム化する。
- ヒストグラムができたら、TIME_END秒は、60, 360, 1000, 3600 など、だんだん増やしながらヒストグラムの形の変化を観察してください。

方針

- 授業内でちょっと考えてもらって、30分くらいたったら説明します。