明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

プログラミング演習2
クラスと継承と色々

中村、小林、橋本、辻野

実験協力者募集中

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 現在、中村研では様々な実験協力者を随時募集しています
  - これから卒論・修論シーズンで他の研究室からも募集が多くかかると思います！
  - お金になることもありますし、研究室の雰囲気を知る切っ掛けにもなるので、興味のある人はFMS学科全体の「#募集-実験協力者」チャンネルへぜひ入りましょう！

講義スケジュール

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 09/23 第01回 ガイダンス＋前期の復習＋α
- 09/30 第02回 クラス
- 10/07 第03回 クラスと継承
- 10/14 第04回 クラスと継承と色々
- 10/21 第05回 復習
- 10/28 第06回 中間テスト（40点）
- 11/11 第07回 振り返りとファイル入出力
- 11/18 第08回 ネットワーク
- 11/25 第09回 Web API
- 12/02 第10回 フィジカルコンピューティング
- 12/09 第11回 フィジカルコンピューティング
- 12/16 第12回 フィジカルコンピューティング
- 12/23 第13回 色々
- 01/20 第14回 発表会

中間試験

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- メディア教室を使って講義資料＋課題＋宿題から75%はそのまま出す中間試験（確認テスト）を実施します。
  - 範囲は春学期＋秋学期の第1回～4回のもの
  - ただし解き方については制限を入れないため、解くことができればOKです！

- ぜひ、課題と宿題だけでもちゃんとできるようになっておいてください！

先週のまとめ

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 継承とArrayListを紹介
  - 継承は変数や関数を引き継ぐ
  - 継承されたものはまとめて扱うことが可能
  - ArrayListは柔軟な配列みたいなもので便利なのでマスターしよう！

先週のおさらい

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

```
ArrayList<ObjectBase> list = new ArrayList<ObjectBase>();

void setup() {
  size(400, 300);
  for (int i=0; i<10; i++)
    list.add( new CircleClass() );
  for (int i=0; i<5; i++)
    list.add( new SquareClass() );
  for (int i=0; i<5; i++)
    list.add( new CrossClass() );
  for (int i=0; i<3; i++)
    list.add( new TriangleClass() );
}

void draw() {
  background(255);
  for( int i=0; i<list.size(); i++ ){
    list.get(i).move();
    list.get(i).display();
  }
}
```

<ObjectBase> で型を定義

list.size()で数を取得

list.get(i)でi番目を取得

ポリモーフィズム

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- CircleClass、SquareClass、CrossClass、TriangleClassの親クラス（継承元）はObjectBaseで、同じメソッドを持っている

- ポリモーフィズム（多態性、多様性）
  - 複数の型に属することを許すこと
  - 親クラスから継承されたクラスのインスタンスは、それぞれ親クラスの型に代入できる。呼び出されるのは、継承されたクラスのメソッド
  - つまり、ObjectBaseとして定義しておき、CircleClassもSquareClassもCrossClassもTriangleClassも放り込んで、ただのメソッドで呼び出しが可能！

何に使えるの？

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

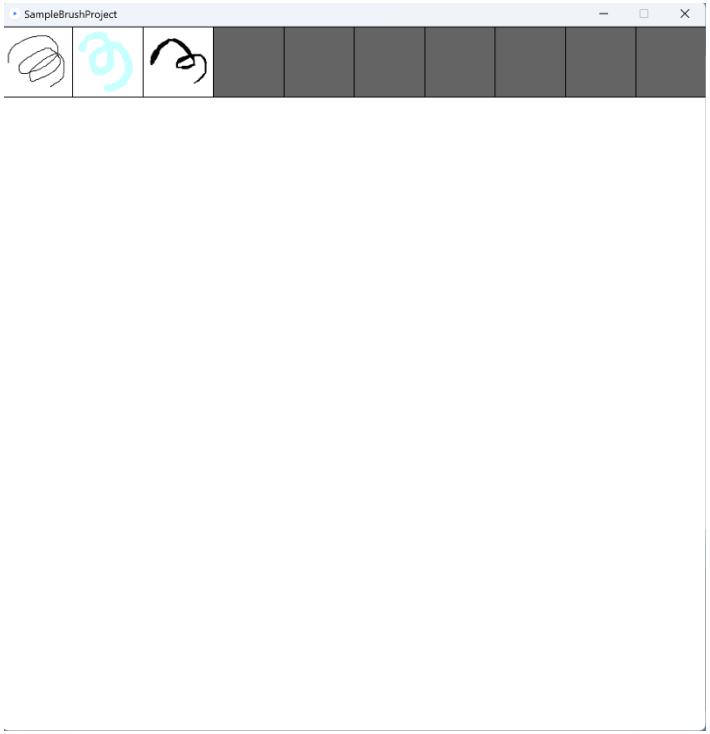
- 継承とかポリモーフィズムとかなんか色々あるけど何に使うの？
  - シューティングで色んな動きをする敵を作る
  - RPGゲームで色んな戦い方をする敵を作る
  - 色々な光り方・広がり方をする花火を作る
  - 色々な結晶の雪を各々の特性で降らせる
  - 草原に色々な草木花を生えさせる
  - ひとの混雑に関するシミュレーションをする

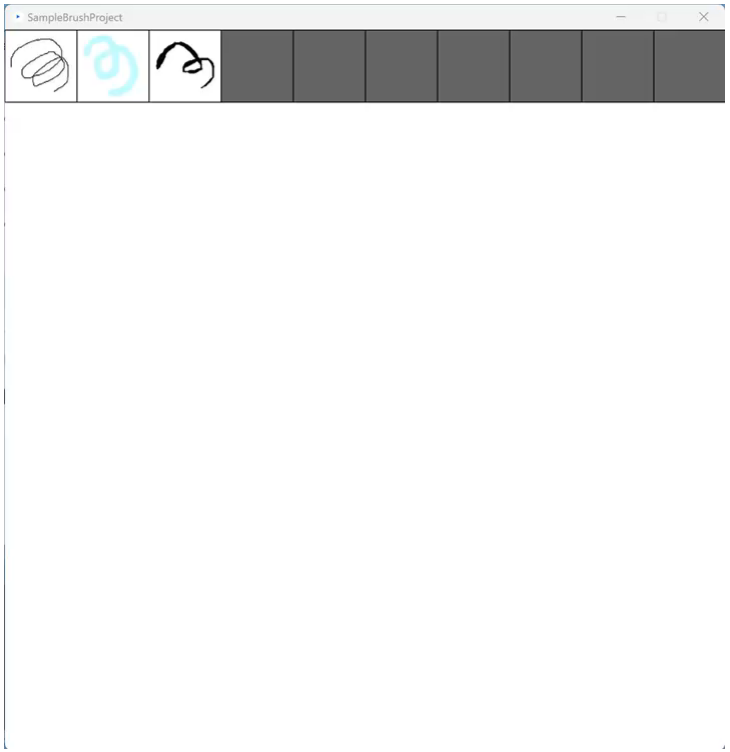
親クラスを定義して継承してクラスを作り
ArrayListで意識せずまとめて動かす！

継承でペイントツールを作る

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- SampleBrushProjectの拡張
  - SampleBrushProjectには、BaseBrushと、BaseBrushを継承したMyBrush1、MyBrush2がある。
  - BaseBrushクラスを継承して、他のペンを作ってみてどうでした？





覚える必要はないけれど

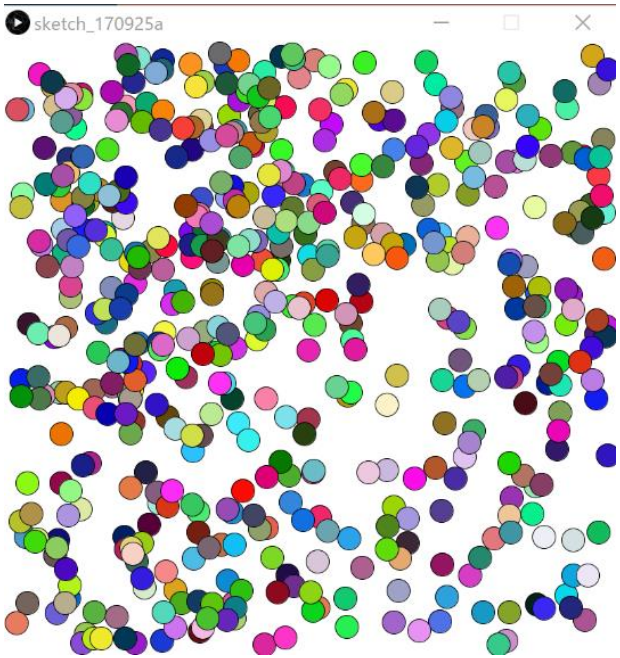
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- アップキャスト
  - 親クラスで定義されたものに，継承されたクラスのインスタンスが代入されると，自動的に親の型に変換されること
- ちなみに，ダウンキャストもあるよ
  - ダウンキャストは安全じゃないよ！

クリックで増やしたい！

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ウインドウ800x800を用意し、CircleClassを利用してマウスクリックするたびにクリックした場所に丸が生まれ，上下左右にランダムな速度で動き、上下左右の端で跳ね返るプログラムを作成せよ。なお、クリックでいくらでも作成できるようにせよ。
- たくさん増やして管理するには ArrayListを使う！



コンストラクタで定義

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- インスタンスを作るときに初期位置をマウスの位置に！

```
class CircleClass {
  float x;
  float y;
  float vx;
  float vy;

  CircleClass() {
    x = mouseX;
    y = mouseY;
    vx = random(1,5);
    vy = random(1,5);
  }
```

コンストラクタ内で mouseX，mouseYを使う？

- これだと使い回せないので困る！！！
- 普通に作ったときにおかしくなる

関数を作る？

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 関数を作れば一応実現は可能

```
XXX.setPos(mouseX, mouseY);
```

  - 初期値を設定するだけなのでコンストラクタでやりたい

```
class CircleClass {
  float x;
  float y;
  float vx;
  float vy;

  CircleClass() {
    x = random(width);
    y = random(height);
    vx = random(1,5);
    vy = random(1,5);
  }

  void setPos(float _x, float _y) {
    x = _x;
    y = _y;
  }
```

コンストラクタで定義！

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- インスタンスを作るときに

```
new CircleClass(mouseX, mouseY)
```

みたいな感じで初期位置を明示的に決めたい！

コンストラクタとして座標を指定できるものを用意する！

```
class CircleClass {
  float x;
  float y;
  float vx;
  float vy;

  CircleClass() {
    x = random(width);
    y = random(height);
    vx = random(1,5);
    vy = random(1,5);
  }

  CircleClass(float _x, float _y) {
    x = _x;
    y = _y;
    vx = random(1,5);
    vy = random(1,5);
  }
```

new CircleClass(mouseX, mouseY)

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- クリックした場所から丸がうまれる！

```
ArrayList<CircleClass> circles = new ArrayList<CircleClass>();

void setup() {
  size(400, 300);
}

void draw() {
  background(255);
  for(int i=0; i<circles.size(); i++){
    circles.get(i).move();
    circles.get(i).display();
  }
}

void mousePressed() {
  circles.add( new CircleClass(mouesX, mouseY) );
}
```

newで指定！

オーバーロード！

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- CircleClass(){…}と
  CircleClass(float _x, float _y){…} と
  2つあってもよいの？
  - 引数の数とか型が違えば大丈夫！！

    CircleCLass() は

    ```
    CircleClass c = new CircleClass();
    ```

    CircleClass(float _x, float _y) は

    ```
    CircleClass c = new CircleClass(500.0, 100.0);
    ```

    で，それぞれ呼び出される．

オーバーロード（多重定義）
同じ関数の名前だけれど，引数の数や型を変えて振る舞いを変えることができるもの

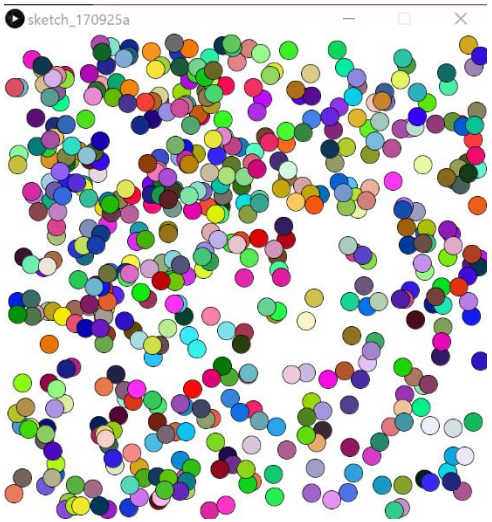
クリックによる丸の生成

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ウインドウ800x800を用意し，CircleClassを利用してマウスクリックするたびにクリックした場所に丸が生まれ，上下左右にランダムな速度で動き，上下左右の端で跳ね返るプログラムを作成せよ．なお，クリックでいくらでも作成できるようにせよ
- また，CircleClassを改良してそれぞれの丸の色は初期値としてランダムに設定するようにせよ

  ［ヒント］色情報を持つインスタンス変数を追加する

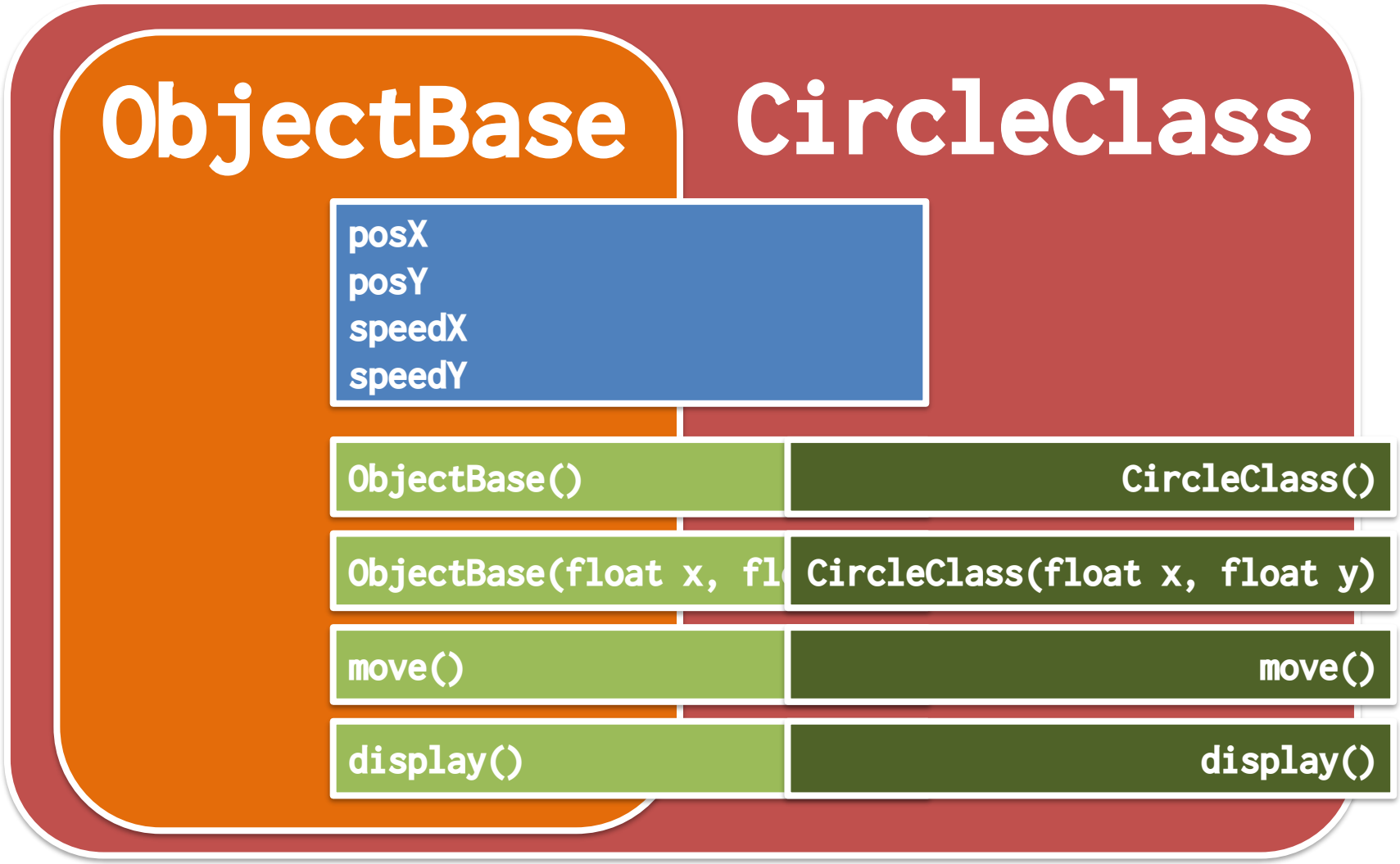
  - int red;
  - int green;
  - int blue;



注意

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 引数がある場合は要コンストラクタの定義



親コンストラクタの呼び出し

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 空っぽで同じ引数のコンストラクタを作る

```
class ObjectBase {
  float x;
  float y;
  float vx;
  float vy;
  ObjectBase(){
    x = random(width);
    y = random(height);
    vx = random(-5, 5);
    vy = random(-5, 5);
  }
  ObjectBase(float _x, float _y){
    x = _x;
    y = _y;
    vx = random(-5, 5);
    vy = random(-5, 5);
  }
```

```
class CircleClass extends ObjectBase {

  CircleClass(){

  }

  CircleCLass(float _x, float _y){
    super(_x, _y);
  }
```

クリックで敵を倒すゲーム

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 独特の動きをする5個の○と、同じく独特の動きをする5個の□、5個の×をクリックにより倒すゲームを作成せよ

どう実現する？

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- EnemyRootクラスを作り下記のクラスを継承して作成する
  - EnemyCircle
  - EnemySquare
  - EnemyCross

- どんな変数が必要だろうか？
  - 場所、速度、生きているか
- どんなメソッドが必要だろうか？
  - 初期化、移動、描画、当たり判定

EnemyRootクラス

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
class EnemyRoot
{
  float x;
  float y;
  float vx;
  float vy;
  boolean live;

  EnemyRoot() {
    x = random(width);
    y = random(height);
    vx = random(-10, 10);
    vy = random(-10, 10);
    live = true;
  }
```

ObjectBase と似てる
動くものは似がち

```
  void move() {
    x = (x + vx + width) % width;
    y = (y + vy + height) % height;
  }

  void hit(int _x, int _y) {
    if((int)x == _x && (int)y == _y){
      live = false;
    }
  }

  void display() {
    if(live) {
      point(x, y);
    }
  }

  boolean isLive() {
    return live;
  }
}
```

- 継承して move を工夫し色々面白くする
  - EnemyCircle
    - めちゃくちゃ速い
  - EnemySquare
    - 不規則運動
  - EnemyCross
    - 途中で消える

クラスの基本はここまで

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ポイント：カプセル化
  - 中の変数はいじらせずに、メソッドで命令できるようにします（エラーのもとなので、変数は基本的に直接いじらない！）
  - これが上手にはまると、きれいなコードになっていき、再利用可能性が高まります。

- まぁクラスの勘所は、何度も設計して身についていくものですので、恐れず設計してみて下さい。で、あーこうしたら良かったと気づいて下さい。

PVector型

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- ベクトル表現を扱うクラス
  - .x と .y の実数値で2次元の値を格納
  - PVector v = new PVector(x, y); で定義
    - なお、引数は実数値 float
  - v.add(v2); vにv2を足し合わせる
  - v.sub(v2); vからv2を引く
  - v.mult(n); vをn倍する
  - v.rotate(theta); vをtheta回転する

PImage型 / 画像型

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 画像を格納および描画するクラス
  - .width や .height で画像の縦横のサイズ取得
  - .resize() で画像サイズを変更可能
  - .save() で画像を保存可能
  - .filter() で各種フィルタをかけることが可能

```
PImage img;
size(400, 400);
img = loadImage("画像ファイル名");
img.filter(BLUR, 6);
image(img, 0, 0);
```

```
フィルタ例 ()内はオプション
THRESHOLD (0-1.0)
GRAY
OPAQUE
INVERT
POSTERIZE (2-255)
BLUR (1以上. 半径)
ERODE
DILATE
```

http://www40.atwiki.jp/spellbound/pages/1800.html
http://processing.org/reference/PImage_filter_.html

音楽の再生

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室


```
import processing.sound.*;
SoundFile bgm; //BGM格納用の変数
int x = 0;
int vx = 4;

void setup()
{
  size(400, 400);
  bgm = new SoundFile(this, "bgm.mp3"); //mp3ファイルを指定する
  bgm.loop(); //bgmを再生
}

void draw()
{
  background(0, 0, 0);
  x = x + vx;
  if(x >= width)
  {
    x = width;
    vx = -vx;
  }
  else if(x <= 0)
  {
    x = 0;
    vx = -vx;
  }
  circle(x, 200, 20);
}
```

SoundFile で音楽再生

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- 準備
  - 音楽再生ライブラリ（便利関数群）の読み込み
    - `import processing.sound.*;`
  - SoundFileの変数を初期化し，初期化
    - **SoundFile はサウンド関係を扱うクラス**
    - `SoundFile bgm;`
    - `bgm = new SoundFile(this, "ファイル名");` でファイル読込
    - `bgm.loop(); // 繰り返し再生`
    - `bgm.stop(); // 停止`
  - SoundFile に効果音を読み込む
    - `SoundFile se = new SoundFile(this, "ファイル名");`
    - `se.play(); // 効果音の再生`

Stringクラスのメソッド

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- charAt(num);　　num文字目の文字を返す（0から始まる）
- indexOf(文字列);　入力された文字列が何文字目か？
- indexOf(文字列, n);n文字目以降で入力された文字列が何文字目か？
- length();　　入力された文字の文字数を返す
- substring(x);　x文字目から最後までを出力
- substring(x, y); x文字目からy-1文字目までを出力
- toLowerCase();　全てを小文字に変換する
- toUpperCase();　全てを大文字に変換する
- replace(文字列A, 文字列B);
  - 文字列Aを文字列Bに変更する
- split(文字列);　文字列を分割

http://processing.org/reference/String.html
http://docs.oracle.com/javase/6/docs/api/java/lang/String.html

クラス学習の参考資料

明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- The Nature of Code
  - Processingでクラスを使いつつ，物理世界の再現について学ぶことができるうえ，ニューラルネットワークとかも学べるよ！
  - 英語も難しくないから学習に良いよ！
  - https://natureofcode.com/book/