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ArrayList

中村，高橋，小林，橋本

今後のスケジュール

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

- 09/21 ガイダンス
- 09/28 クラス
- 09/05 クラス＋継承
- 10/12 クラスと継承の復習（中村）
- 10/19 クラスと継承の再復習（高橋）＋中間課題提示（中村）
- 10/26 質問回
- 11/02 お休み
- 11/09 ファイル入出力＋中間課題提出締め切り（対面スタート）
- 11/16 ネットワーク
- 11/23 XML＋WebAPI
- 11/30 フィジカルコンピューティング（小林）
- 12/07 フィジカルコンピューティング（小林）
- 12/14 フィジカルコンピューティング（小林）
- 12/21 グループワーク（企画書を出す）
- 01/18 発表会（ホールにて）

中間試験断念

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た試験を実施できないため，中間試験（30点）実施を断念
- かわりに中間課題（30点）を出すことにしました

- まだ案の段階ですが
  - クラスを使って現実世界の模倣をする（drawやsetupの行数は最小限に）
  - クラスを学んだ学生向けの課題を作る

  などを出そうと考えています．

ちなみに

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1の最終課題は出せば満点というものではありませんでした
  - 提出した学生の平均点23点/30点
  - 課題として成り立っていない，課題の意図がつかめない，レギュレーションを守っていないなどもありました
- しっかり採点しますので，手を抜かずちゃんと取り組むようにしてください

今後のスケジュール

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

- 09/21 ガイダンス
- 09/28 クラス
- 09/05 クラス＋継承
- 10/12 クラスと継承の復習（中村）
- 10/19 クラスと継承の再復習（高橋）＋中間課題提示（中村）
- **10/26 質問回**
- 11/02 お休み

ここをうまく使ってください（対面可にするかも？）

- 11/09 ファイル入出力＋中間課題提出締め切り（対面スタート）
- 11/16 ネットワーク
- 11/23 XML＋WebAPI
- 11/30 フィジカルコンピューティング（小林）
- 12/07 フィジカルコンピューティング（小林）
- 12/14 フィジカルコンピューティング（小林）
- 12/21 グループワーク（企画書を出す）
- 01/18 発表会（ホールにて）

ということで

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

- 今日はまず先週の宿題を解説しながら復習をします
- 説明をしている最中に課題をといていたがために，課題を理解できていない人がいたので，ちゃんと聞くこと！
  - 特に，先週の課題を1つでもできなかった人

課題3-1:basic_CharaClass

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

- 配布するCharacterBaseクラスを継承し，CharacterBaseクラス内のcenterX，centerYを利用してその位置に自身のキャラクタを描画するようにせよ
  - 継承したクラスではdisplayActiveとdisplayInactiveをオーバーライドせよ
  - moveメソッドをオーバーライドして特殊な動き方をするように変更してもかまわないが，画面の外に出ていかないようにせよ
- なお配布プログラムではCharacterBaseクラスのキャラクタ？のみを描画しているが，自分で作成したクラス（継承したクラス）のキャラクタも一緒に表示するようにせよ
- クラスの名前は名前と番号が含まれるようにし、そのクラスのみ別ファイルとして作成せよ
  - （例）X-3-34 中村の場合、クラス名を「ExNakamura334」とする
- 来週配布して利用します

こちらの問題について

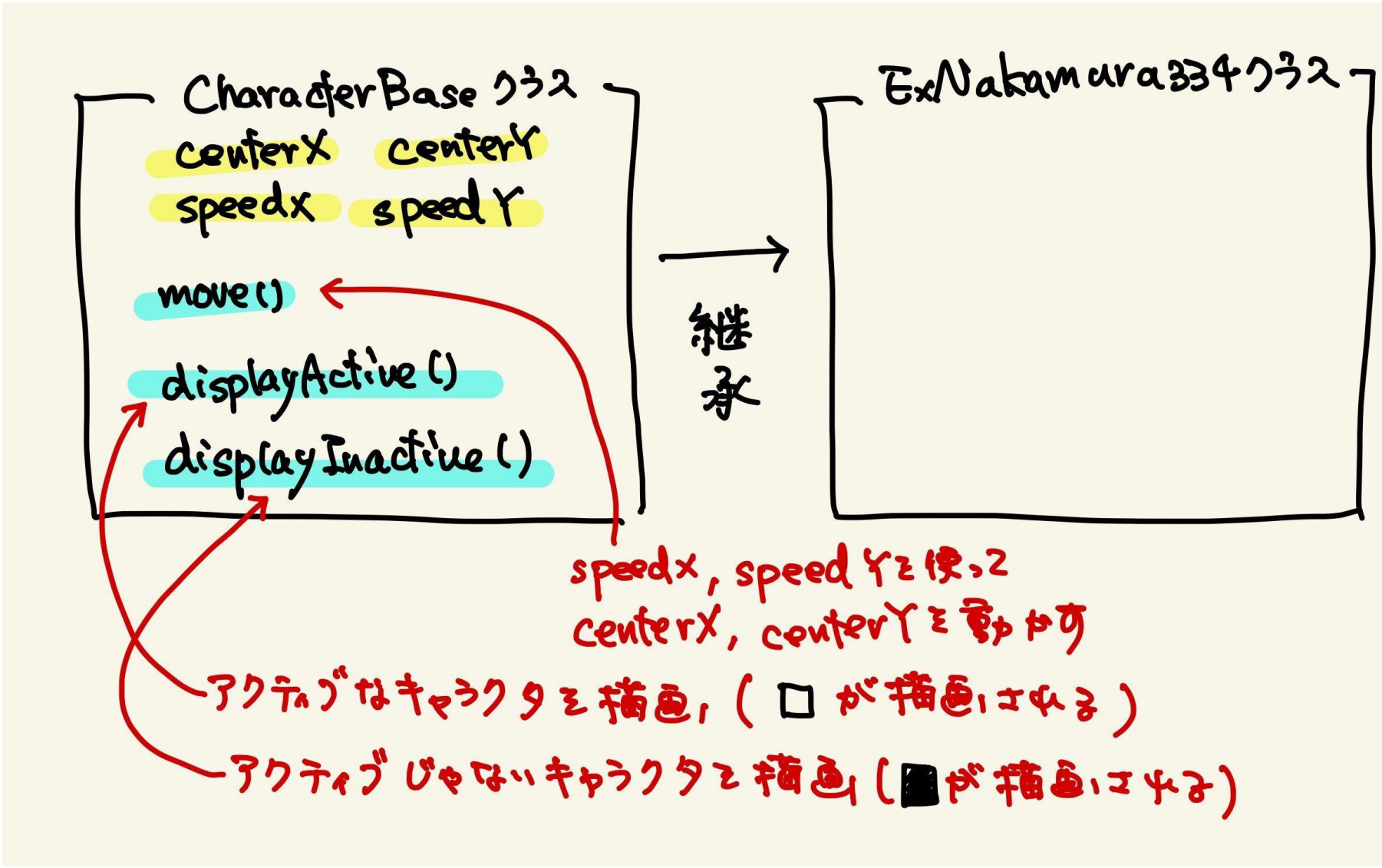
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室

- CharacterBaseクラスをいじってる人が多数いたようなので再度説明します
- 今日の演習課題に使いますので，確実にできるようになってください！

- クラス名（＆タブの名前）はEx名前組番号としてください
  - （例）1-3-34 中村の場合、クラス名は「ExNakamura334」
  - できあがったら，そのファイル（上の場合はExNakamura334.pde）をDiscordで研究室のメンバーに共有してください

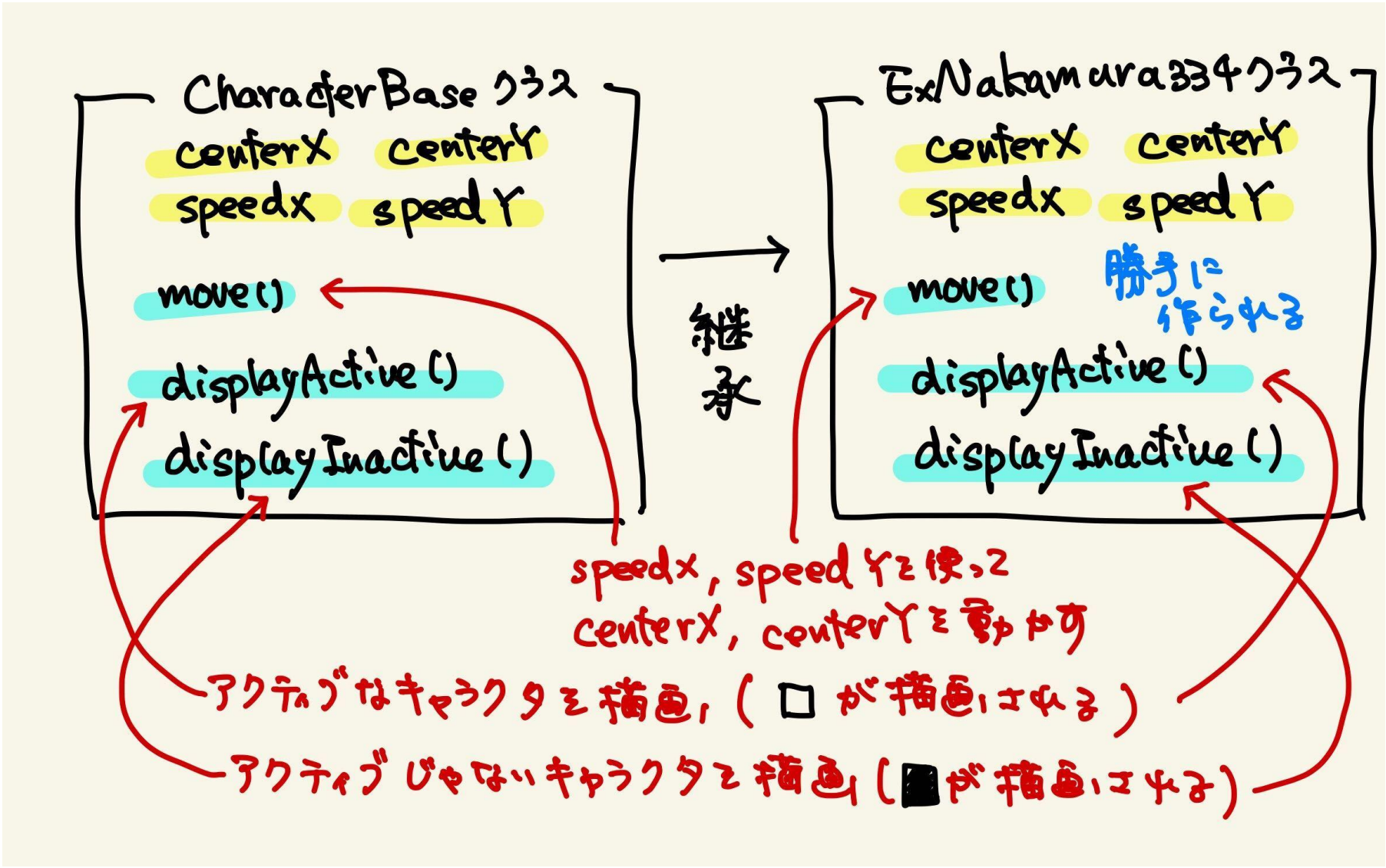
CharacterBaseの継承？

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



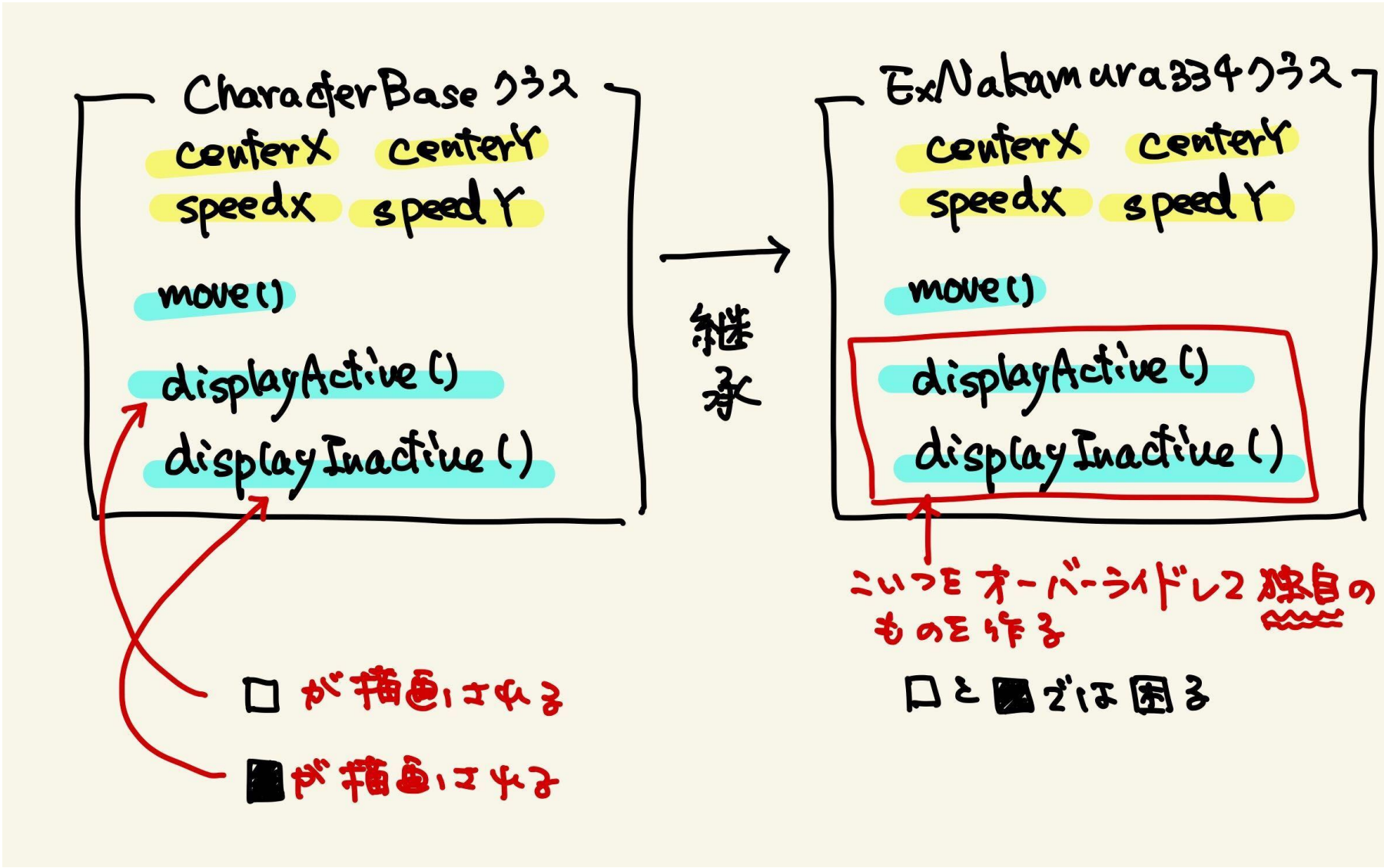
CharacterBaseの継承？

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



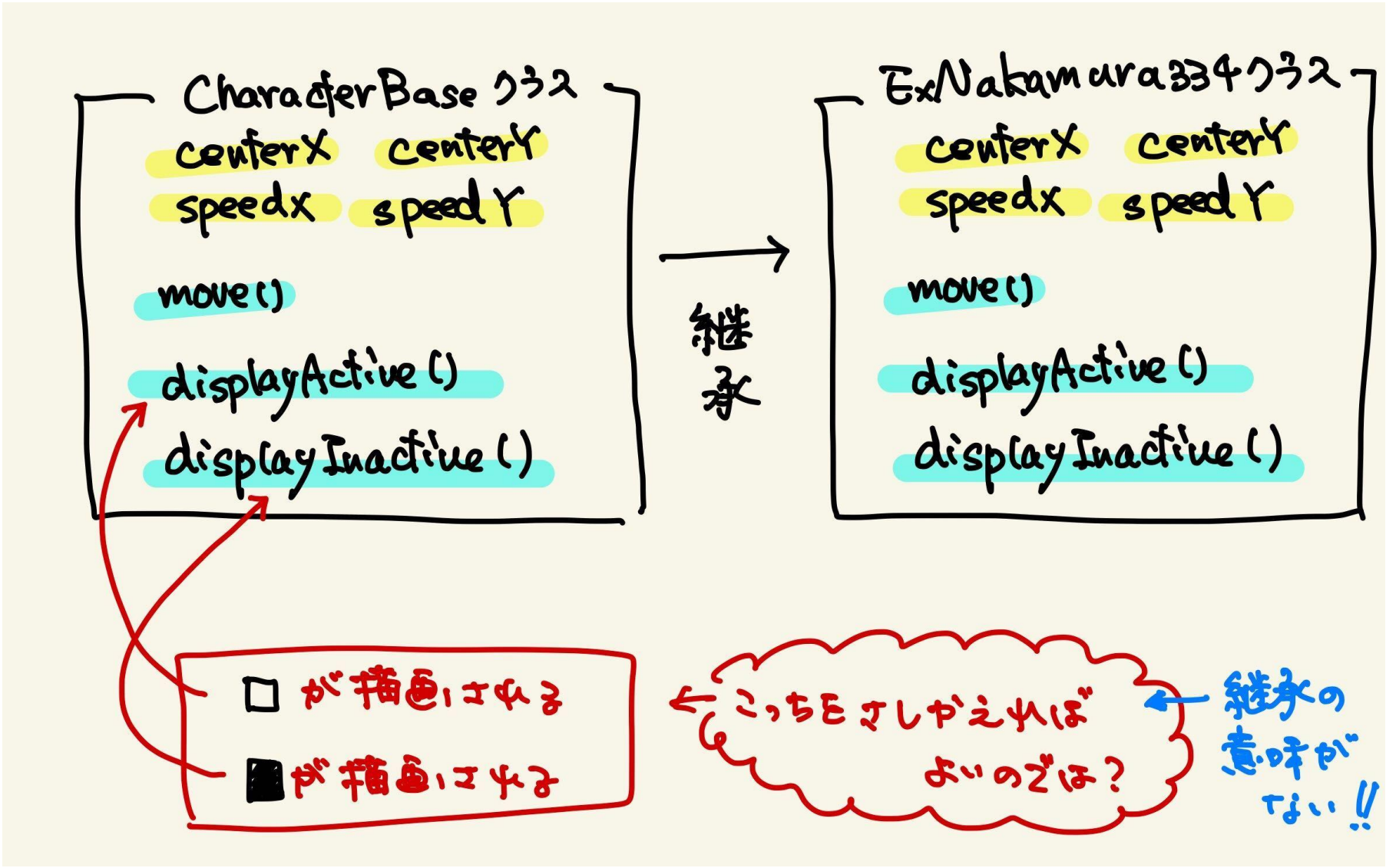
CharacterBaseの継承？

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



CharacterBaseの継承？

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



宿題3-1: hw_GenomeAnalyzer

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

- 以下の条件を満たすGenomeAnalyzerクラスを作成し，次のページで示すプログラムで動作させよ
  - コンストラクタの引数として塩基配列を文字列として与え，インスタンス変数でその塩基配列情報を保持するようにせよ
  - 返り値が実数値のcalcRatioA(), calcRatioT(), calcRatioG(), calcRatioC()というインスタンスメソッドを作成し，それぞれA,T,G,Cのそれぞれの出現率を返すようにせよ
    - わかる人は，引数をchar型にし，メソッドはcalcRatio()だけでもよい
  - 返り値が整数のlength()というインスンタンスメソッドを作成し，塩基配列の長さを返すようにせよ
  - 引数として文字列パターンを与え，返り値としてその文字列が塩基配列内で現れる回数を整数値で返すcountPatternメソッドを作成せよ．ただし，「AAAA」は「AAA」が2回とカウントするようにせよ
    - 引数プログラム内でそれぞれcountPattern("AAA"), countPattern("TTT"), countPattern("GGG"), countPattern("CCC")のそれぞれの出現回数を出力するようにせよ

宿題3-1: hw_GenomeAnalyzer

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


```
GenomeAnalyzer analyzer;

void setup() {
  size(400, 300);
  // hw_Covid19Genome の塩基配列を利用せよ↓
  analyzer = new GenomeAnalyzer("すんごく長い塩基配列");

  println("塩基配列の長さは" + analyzer.length() + "文字");

  println("Aの出現率", analyzer.calcRatioA());
  println("Gの出現率", analyzer.calcRatioG());
  println("Tの出現率", analyzer.calcRatioT());
  println("Cの出現率", analyzer.calcRatioC());

  println("AAAの出現回数", analyzer.countPattern("AAA"));
  println("GGGの出現回数", analyzer.countPattern("GGG"));
  println("TTTの出現回数", analyzer.countPattern("TTT"));
  println("CCCの出現回数", analyzer.countPattern("CCC"));
}
```

宿題3-2: hw_JankenGuriko

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

- グリコというゲーム（グーで勝つと+3，チョキで勝つと+5，パーで勝つと+6の点が入る）がある
  - チョキで+6が一般的だがここでは+5とする
- 毎回じゃんけんを繰り返し，10000点を先に獲得したほうが勝ちになる
- 配布する hw_JankenGuriko 内のJankenAgentクラスを継承し，自身のエージェントを作成せよ
- battleメソッドに自身のエージェントと，敵のエージェントを引数として与え，10000点になるまで戦わせて勝ちましょう！
  - 少なくとも JankenAgent自身，AgentNakamura334には勝ちましょう
  - できれば，知り合いのAgentももらって，勝ちましょう！！

宿題3-3: hw_CalcVector

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

- 2つのベクトルをメソッドでセットし，様々な計算を可能とするVectorOperationクラスを作成し，次ページのプログラムで動作を確認せよ
  - 引数を実数値の配列としてベクトルを与え，インスタンス変数としてベクトルを配列として管理するようにする setVectorA(), setVectorB()という2つのメソッドを作成せよ
  - 返り値が実数のcalcVectorMagnitudeA(), calcVectorMagnitudeB()メソッドを作成し，それぞれのベクトルの大きさを求めて返すようにせよ
  - 返り値が実数のcalcInnerProduct()メソッドを作成し，内部にもつ2つのベクトルの内積を求めて返すようにせよ
  - 返り値が実数のcalcCosSimilarity()メソッドを作成し，内部に持つ2つのベクトルのコサイン類似度を計算して返すようにせよ．2つのベクトルの類似度は，次ページ以降に示すコサイン類似度を利用することで求めることができる
  - 返り値が実数のcalcArea()メソッドを作成し，内部の2つのベクトルから面積を求め，返すようにせよ．2つのベクトルからなる三角形の面積は，次ページ以降の式で求めることが可能である

宿題3-3: hw_CalcVector

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

- setVectorAとsetVectorBで指定されるベクトルの次元（配列の要素数）は，同じであると決めつけて良い

```
void setup() {
  VectorOperation vo = new VectorOperation();
  float[] vectorA = {10, 0, 1, 9};
  float[] vectorB = {8, 1, 1, 4};
  vo.setVectorA(vectorA);
  vo.setVectorB(vectorB);

  println("Aの大きさ", vo.calcVectorMagnitudeA());
  println("Bの大きさ", vo.calcVectorMagnitudeB());

  println("内積", vo.calcInnerProduct());
  println("コサイン類似度", vo.calcCosSimilarity());
  println("面積", vo.calcArea());
}
```

惑星と衛星の様なオブジェクト

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

ObjectBaseを継承し，800x600のウインドウ内で，任意の場所x，yから任意の速度で移動する3つの赤色の円（直径30ピクセルの惑星）を描画し，右端・左端・上端・下端に来ると跳ね返るようにする．また，赤色の円（惑星）には円の中心から50の距離があるところに1つの衛星（直径10ピクセル）があり，5度ずつ円の周りを回転するようにせよ

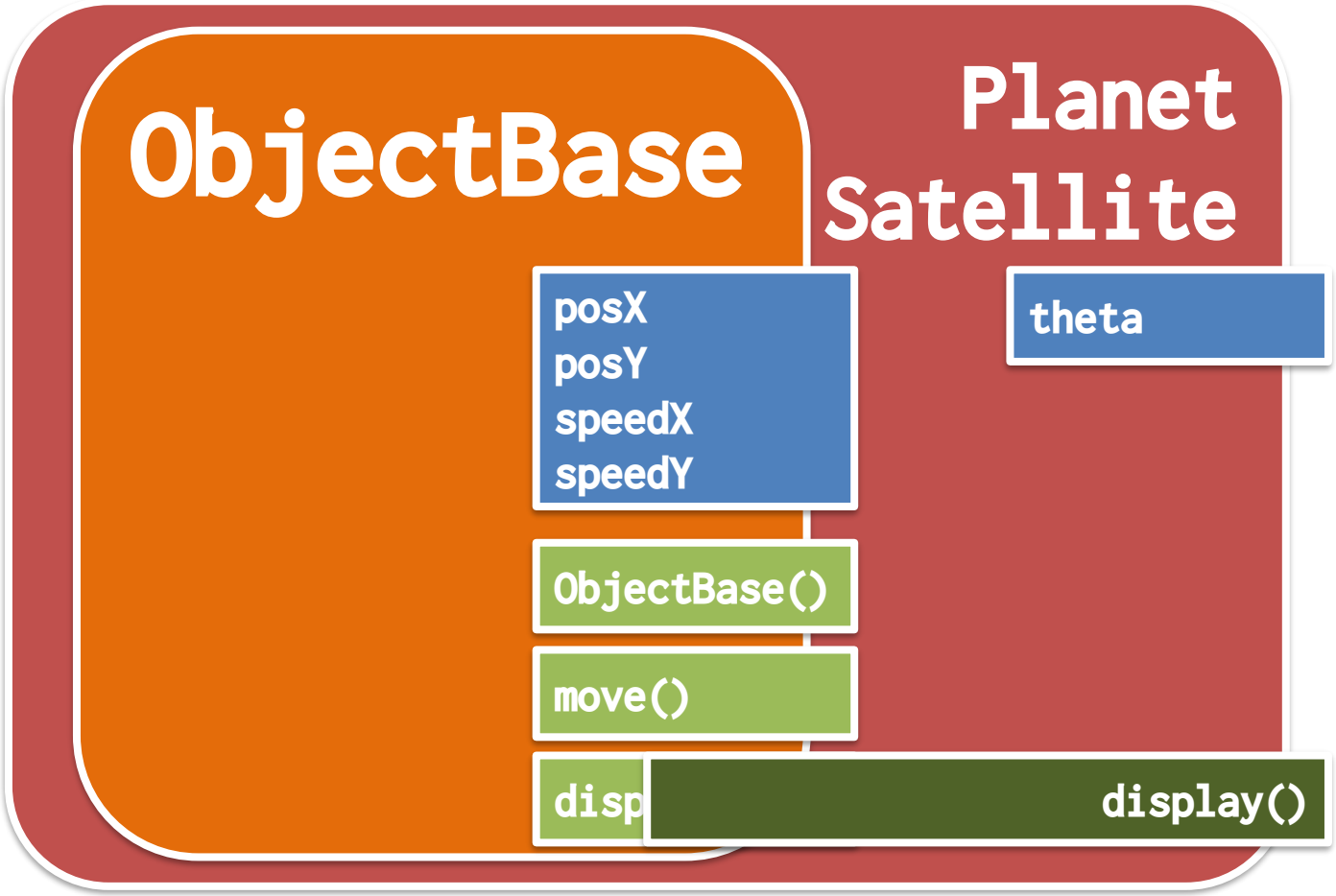
- 考え方
  - 円の中心（x，y）から衛星の方向の角度（0〜360度）をthetaとすると，衛星の座標は

```
(x+50*cos(radians(theta)), y+50*sin(radians(theta)))
```

継承すると・・・

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

- ObjectBase から継承して PlanetSatellite を作成し，theta という変数を追加し，display() というメソッドを上書き（オーバーライド）



惑星と衛星の様なオブジェクト

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

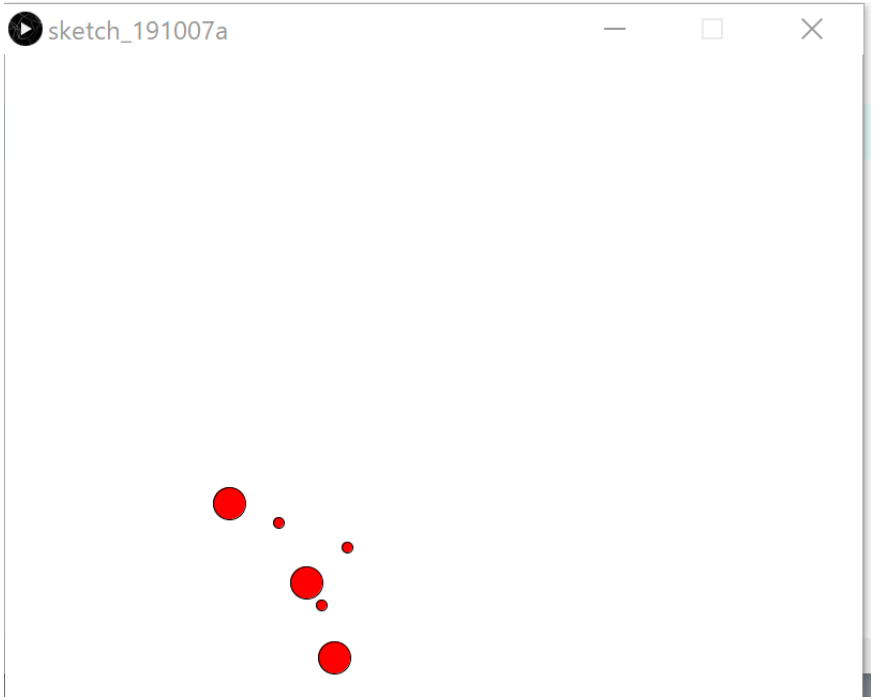
- ObjectBaseを継承して，変数を追加する

```
class PlanetSatellite extends ObjectBase
{
  int theta;

  void display()
  {
    fill( 255, 0, 0 );
    ellipse(posX, posY, 30, 30);
    theta = theta + 5;
    float sx = posX + 50*sin(radians(theta));
    float sy = posY + 50*cos(radians(theta));
    ellipse(sx, sy, 10, 10);
  }
}
```

惑星と衛星の様なオブジェクト

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


```
PlanetSatellite ps1;
PlanetSatellite ps2;
PlanetSatellite ps3;
void setup()
{
  size( 800, 600 );
  ps1 = new PlanetSatellite();
  ps2 = new PlanetSatellite();
  ps3 = new PlanetSatellite();
}

void draw()
{
  background(255);
  ps1.move();
  ps2.move();
  ps3.move();
  ps1.display();
  ps2.display();
  ps3.display();
}
```



惑星と衛星の様なオブジェクト

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

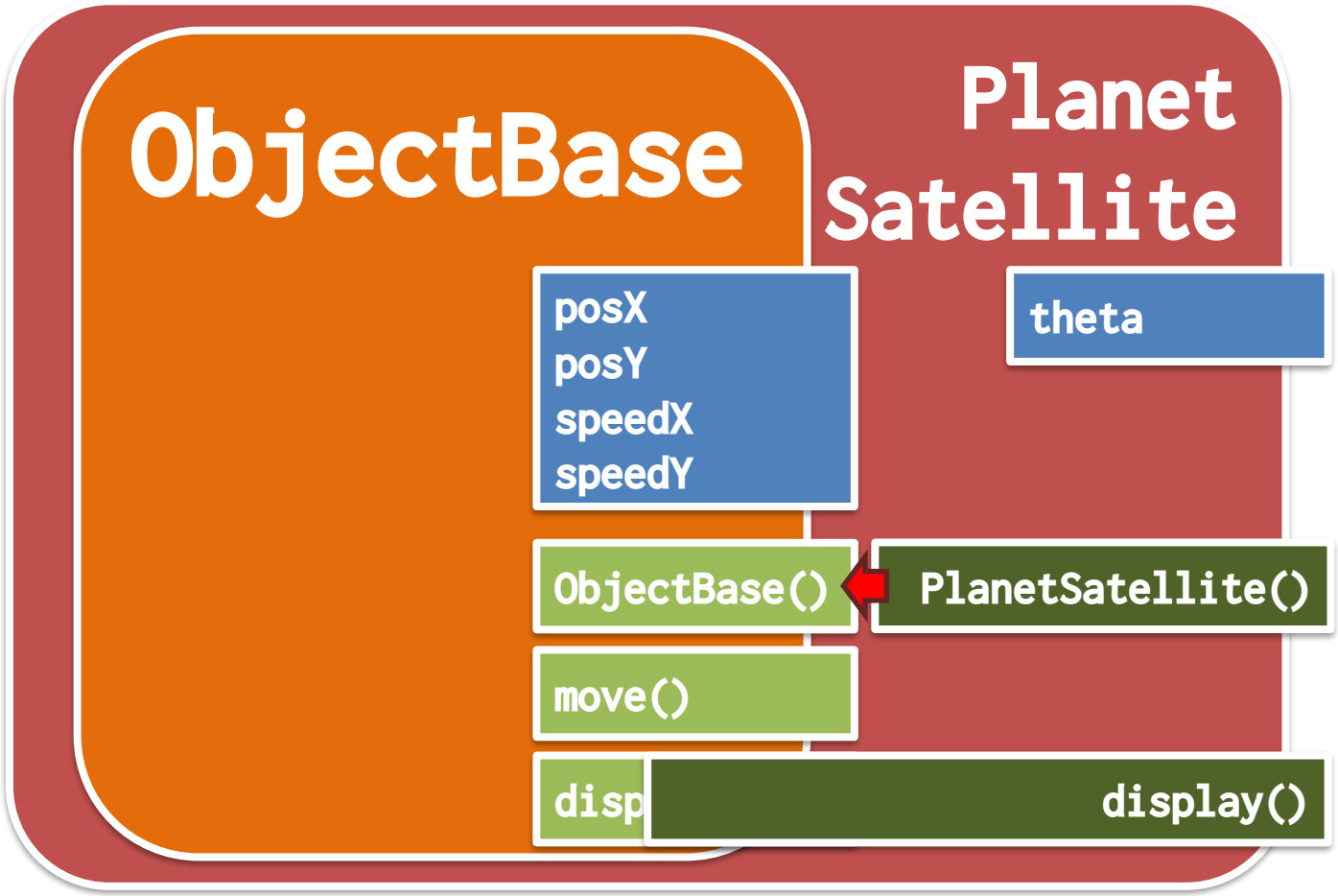
衛星の開始角を0～360度の任意の場所にしたい．どうするか？

- 考え方
  - コンストラクタで指定したらOK？

継承すると・・・

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

- コンストラクタで定義する
  - 勝手に親が呼び出されるので便利！（明示的に呼び出す場合は super() と書く）



やってみる

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

```
class PlanetSatellite extends ObjectBase {
  int theta;
  PlanetSatellite(){
    theta = (int)random(360);
  }
  void display(){
    fill(255, 0, 0);
    ellipse(posX, posY, 30, 30);
    // 回転させる
    theta = theta + 5;
    int sx = (int)(posX+50*sin(radians(theta)));
    int sy = (int)(posY+50*cos(radians(theta)));
    ellipse(sx, sy, 10, 10);
  }
}
```

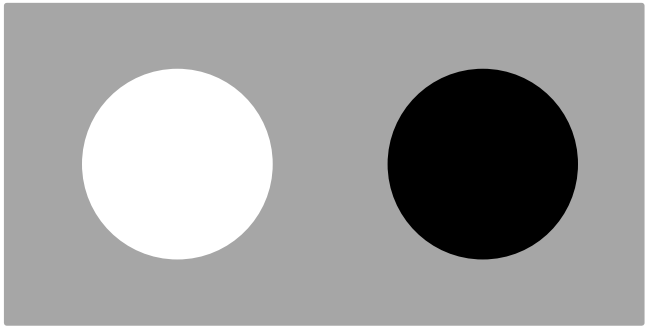
独立したボタン

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

2つのボタンをもつUIのパーツを作成したい

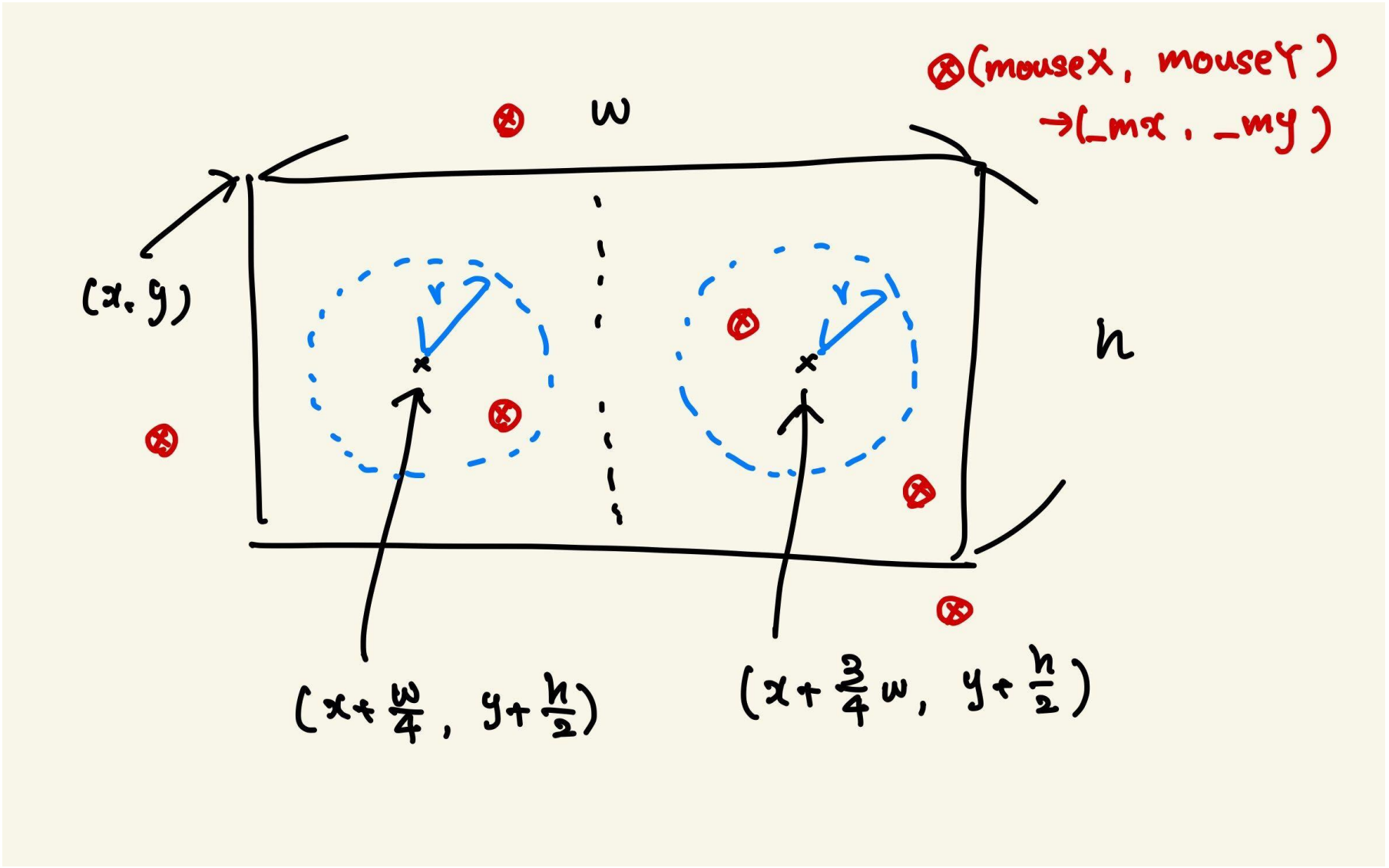
- UIのパーツは，左右に丸がボタンとして2つならんでおり，四角形により囲まれているものとする
- また，どちらかの丸をクリックすると，クリックされた丸が白→黒→白→黒と繰り返し色を変更するようにしたい
- このパーツをウインドウ内に縦に5つ並べよ

```
class TwoButtonUI {
  int x, y;
  int w, h, r;
  int leftButton;
  int rightButton;

  TwoButtonUI(int _x, int _y){
    x = _x;
    y = _y;
    w = 200;
    h = 100;
    r = 40;
    leftButton = 0;
    rightButton = 0;
  }

  void display(){
    fill(200, 200, 200);
    rect(x, y, w, h);
    if(leftButton == 1) fill(0);
    else fill(255);
    ellipse(x+w/4, y+h/2, r*2, r*2);
    if(rightButton == 1) fill(0);
    else fill(255);
    ellipse(x+w*3/4, y+h/2, r*2, r*2);
  }

  // (_mx, _my)の位置がクリックされた
  void click(int _mx, int _my){
    if(dist(_mx, _my, x+w/4, y+h/2)<=r){
      leftButton = 1 - leftButton;
    } else
    if(dist(_mx, _my, x+w*3/4, y+h/2)<=r){
      rightButton = 1 - rightButton;
    }
  }
}
```



ボタンの当たり判定

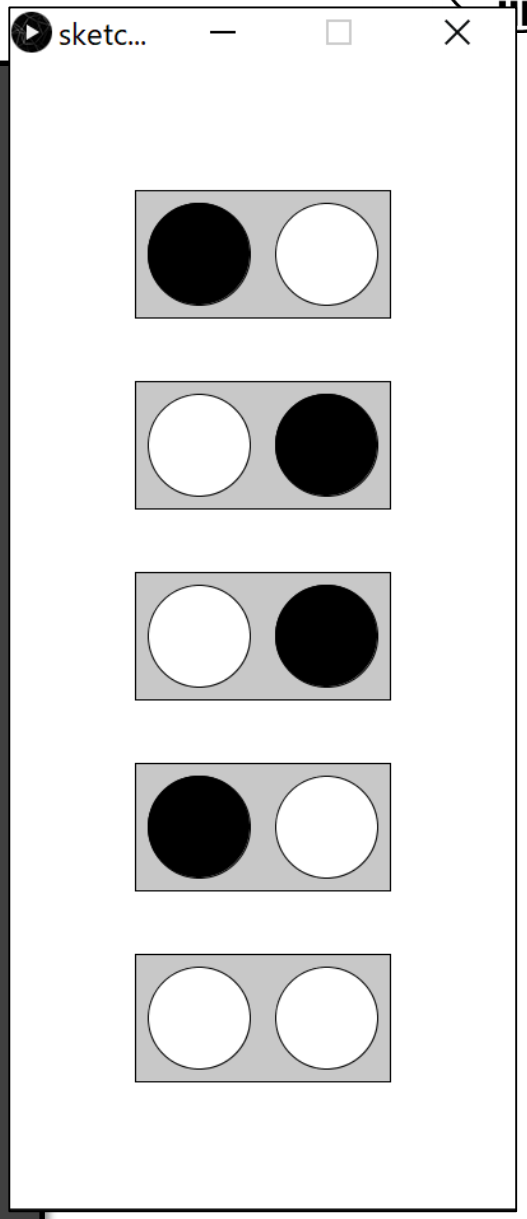
明治大学総合数理学部
先端メディアサイエンス学科
中村研究室



全部クリックする！

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

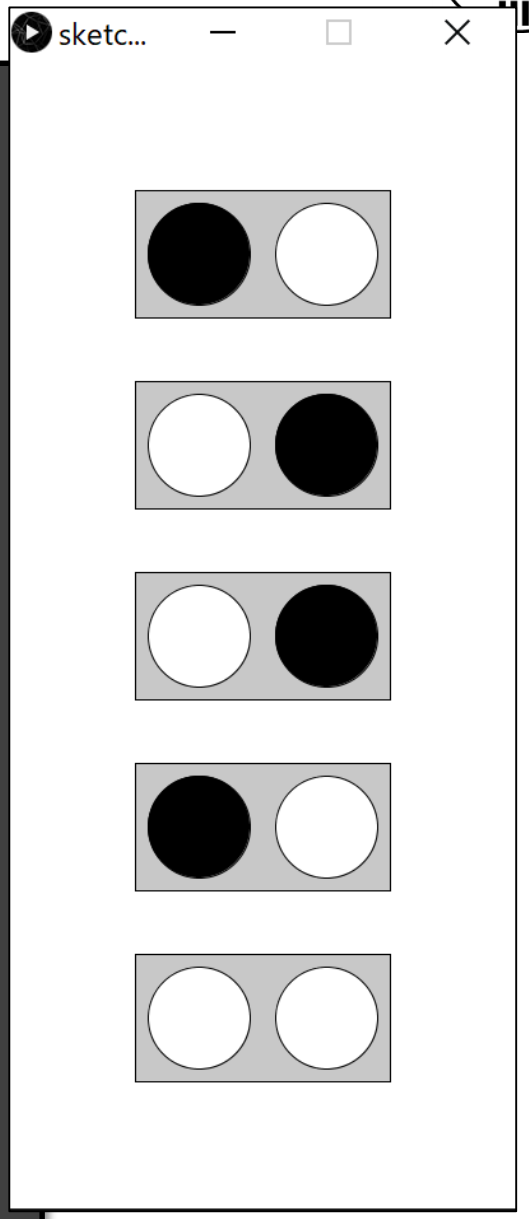
```
ArrayList<TwoButtonUI> UIs = new ArrayList<TwoButtonUI>();

void setup(){
  size(400, 900);
  UIs.add(new TwoButtonUI(100, 100));
  UIs.add(new TwoButtonUI(100, 250));
  UIs.add(new TwoButtonUI(100, 400));
  UIs.add(new TwoButtonUI(100, 550));
  UIs.add(new TwoButtonUI(100, 700));
}

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

void mousePressed(){
  for(int i=0; i<UIs.size(); i++){
    UIs.get(i).click(mouseX, mouseY);
  }
}
```



全部クリックする！

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

```
TwoButtonUI[] UIs = new TwoButtonUI[5];

void setup(){
  size(400, 900);
  UIs[0] = new TwoButtonUI(100, 100);
  UIs[1] = new TwoButtonUI(100, 250);
  UIs[2] = new TwoButtonUI(100, 400);
  UIs[3] = new TwoButtonUI(100, 550);
  UIs[4] = new TwoButtonUI(100, 700);
}

void draw(){
  background(255);
  for(int i=0; i<UIs.length; i++){
    UIs[i].display();
  }
}

void mousePressed(){
  for(int i=0; i<UIs.length; i++){
    UIs[i].click(mouseX, mouseY);
  }
}
```



課題4-1: basic_ManyChara

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

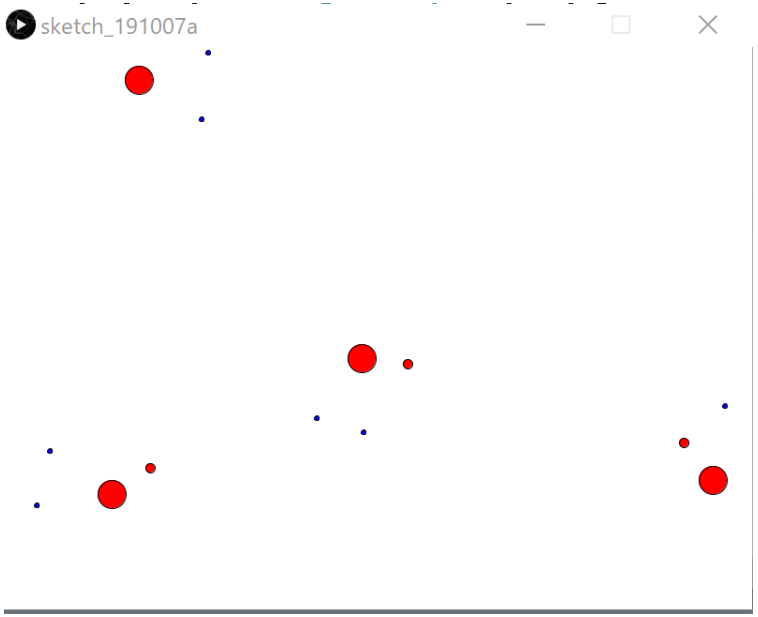
- 色々な人が作成した ExXXXXX クラスを利用して多くのキャラクタが勝手に動き回るプログラムを作成せよ
- またマウスのボタンを押すと，すべてのキャラクタが別の表示スタイルになるようにせよ
  - 押してない間は displayActive を，押している間は displayInactive を利用するようにすると良い
- そのために，各自が作成したキャラクタクラスが入ったファイルを研究室のDiscordにアップせよ
  - 同じ研究室のもの（再履修の場合は，再履修の学生のもの）を5つ以上利用して（同じ研究室で足りなければ他の研究室のを使ってもよい）、キャラクタが画面内を動き回るプログラムを作成せよ
  - 他の研究室のものを使用しつつかなりたくさんのキャラクタを入れてもよい

（ヒント） ArrayListに放り込むだけ！！

課題4-2: basic_PlanetSatellite

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

- PlanetSatellite クラスを継承し，惑星の中心から80ピクセル離れた位置に青色の小さな衛星が2つある（直径5ピクセルで毎フレーム3度ずつ動く）PlanetSatelliteEx クラスを作成せよ．また，そのクラスを用いて4個の惑星（3つの衛星がある）がウインドウ内を動きまわるようにせよ
  - 考え方
    - インスタンス変数を追加する
    - init メソッドと，display メソッドをオーバーライドする

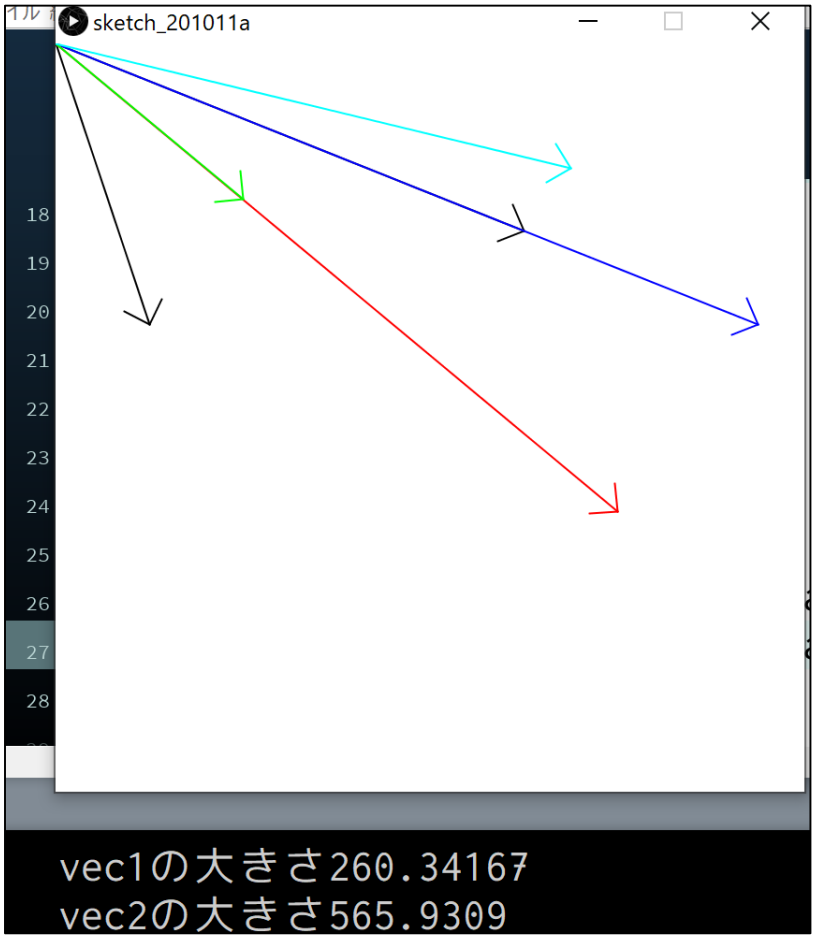


課題4-3: basic_MyVector

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

- 配布する basic_MyVector.zip の MyVectorクラスのベクトルの
  - 足し算 add
  - 引き算 subtract
  - 掛け算 multiply
  - 割り算 divide
  - 大きさ magnitude

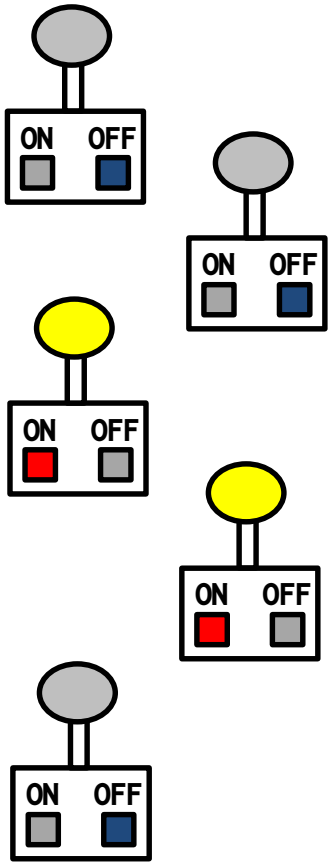
  の計算を行うメソッドを完成させ，右図のような出力になるようにせよ



宿題4-1：hw_SwitchLight

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

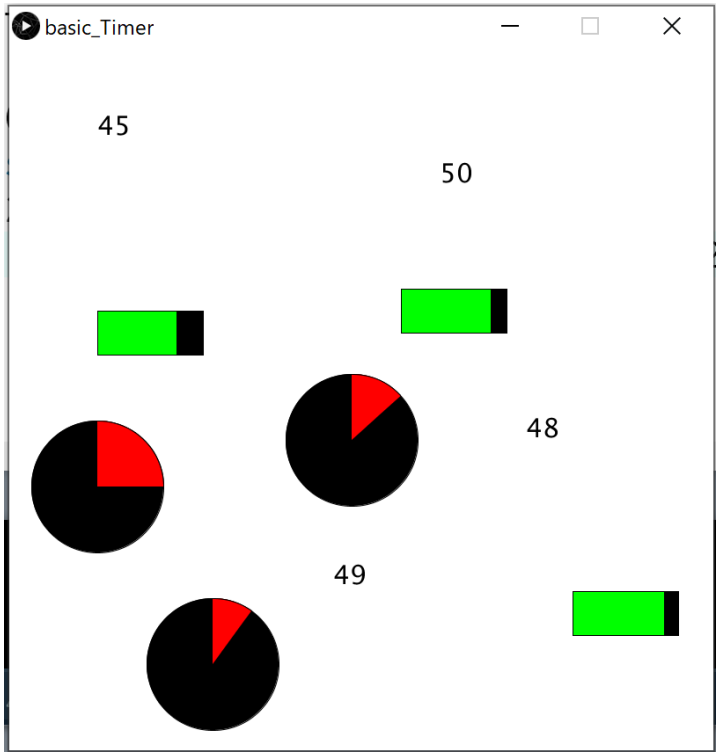
- 2つのボタンと1つのライトからなる照明を作成する．この照明をSwitchLightクラスとして作成せよ（形状については任せる）
- ボタンは左右に配置し，左側のボタンをクリックするとライトが光り，右側のボタンをクリックするとライトが消えるようにせよ
- また，SwitchLightクラスを利用して，5つの照明を画面に表示するようにせよ



宿題4-2: hw_Timer

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

- 配布する hw_Timer.zip を利用し，生成されてから60秒経過すると自動的に消滅する（何も描画されなくなる）タイマーを，TimerCoreクラスを継承して，TextTimerやProgressTimerを参考にしつつ2つ以上実現せよ
- また，マウスクリックしたときにそれぞれのタイマーがランダムに生成されるようにせよ



宿題4-3: hw_VendingMachine

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

- 自動販売機（VendingMachineクラス）では日本円の10円・50円・100円の3種類の硬貨を扱い、販売される商品は全て200円以下で、10円～200円で10円刻みとなっています。
- 自動販売機には投入金額と購入金額からおつりを計算し、おつりを返却するか、商品を購入できないことを知らせる機能が必要です。これらの機能は、以下のルールに従って動作します。
  - お釣りがない場合は「ありがとうございました。おつりはありません」と表示する。
  - お釣りがある場合は「ありがとうございました。おつりは10円3枚と50円1枚です」と表示し、おつりを返却する。
  - 投入金額が足りないまたはおつりを返却できない場合は「商品を購入できません。10円0枚、50円1枚、100円1枚を返却します」と商品を購入できないことを知らせるとともに投入金額を返却する。
  - なお投入されたお金は、おつりとしても利用することが可能です。

宿題4-3: hw_VendingMachine

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

- ヒント
  - VendingMachineクラスには，内部に10円玉，50円玉，100円玉が何枚あるかを管理する変数を用意
  - 初期の枚数をセットするメソッドを作成

    initialize(10円の数, 50円の数, 100円の数) メソッド
  - お金を投入するメソッドを作成

    insert(10円の数, 50円の数, 100円の数) メソッド
  - 購入する商品を指定するメソッドを作成

    buy(値段) メソッド
    - buy メソッドは，標準出力で結果を返すようにせよ．ただし，
    - おつりは「投入金額 - 購入金額」で計算されます。
    - おつりは自動販売機の内部にある硬貨から枚数が最も少なくなるように選んだ硬貨の組合せで返却せよ

宿題4-3: 動作チェック

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

- 色々なパターンを用意して問題ないかを確認しよう

```
vMachine.initialize( 5, 5, 5 );
vMachine.insert( 0, 1, 1 );
vMachine.buy( 130 );
vMachine.insert( 0, 0, 2 );
vMachine.buy( 110 );
vMachine.insert( 0, 0, 2 );
vMachine.buy( 140 );

vMachine.initilize( 9, 8, 7 );
vMachine.insert( 0, 0, 2 );
vMachine.buy( 110 );
vMachine.insert( 0, 0, 2 );
vMachine.buy( 120 );
vMachine.insert( 0, 0, 2 );
vMachine.buy( 130 );
vMachine.insert( 2, 0, 2 );
vMachine.buy( 180 );
```

```
ありがとうございました。おつりは10円2枚と50円0枚と100円0枚です
商品を購入できません。10円0枚、50円0枚、100円2枚を返却します
ありがとうございました。おつりは10円1枚と50円1枚と100円0枚です
ありがとうございました。おつりは10円4枚と50円1枚と100円0枚です
ありがとうございました。おつりは10円3枚と50円1枚と100円0枚です
ありがとうございました。おつりは10円2枚と50円1枚と100円0枚です
商品を購入できません。10円2枚、50円0枚、100円2枚を返却します
```

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

DANIEL SHIFFMAN

- 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！
  - 休みの期間に一通り入力してみると色々学びがあっていいと思います！
  - 英語も難しくないから学習に良いよ！
  - https://natureofcode.com/book/