



プログラミング演習2

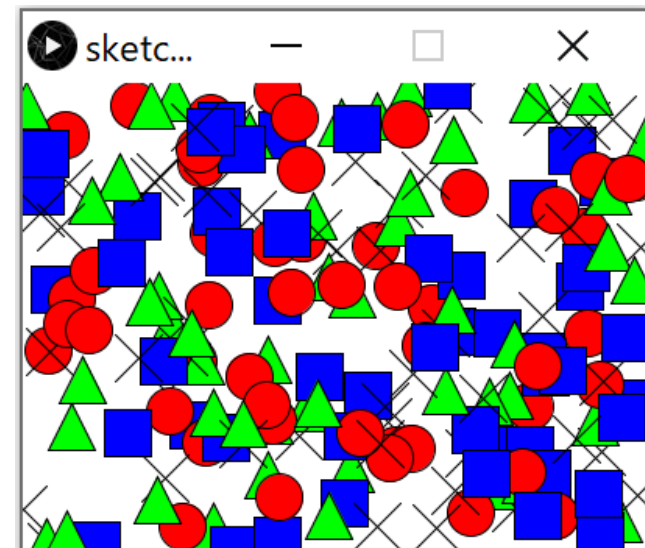
クラスと継承

中村、小林、辻野、橋本

課題3-1: basic_boundA114



- ObjectBaseクラスを継承し、赤色の円が動き回るCircleClass、青色の正方形が動き回るSquareClass、黒色のxが動き回るCrossClass、緑色の△が動き回るTriangleClassを作成せよ。
- またこれを利用して50個の○と、50個の正方形と、50個のxと、50個の△が動き回るプログラムを作成せよ。
- ただし、その速度はx、y方向それぞれ-5~5の実数値とし、○と□は壁で跳ね返り、xと△は跳ね返らずに反対側から出てくるようにせよ。
- なお、ArrayListを利用して実現せよ。



課題3-2: basic_JankenBattle



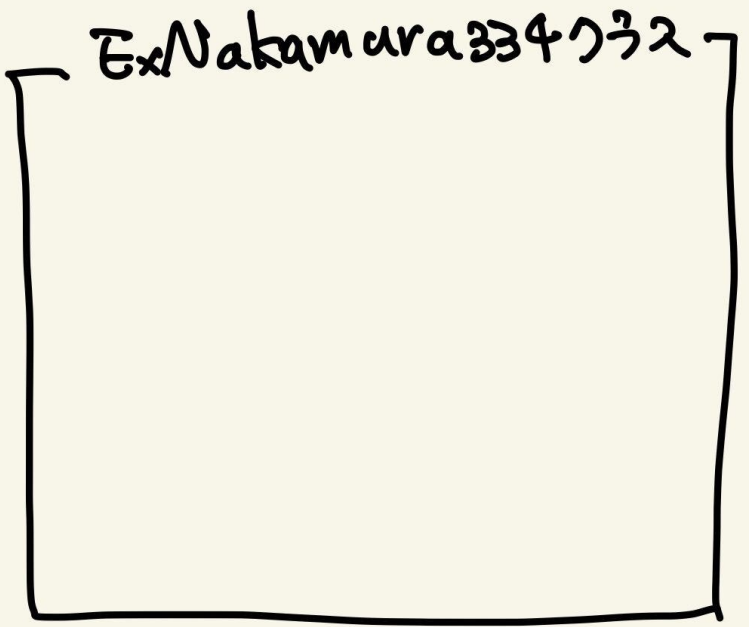
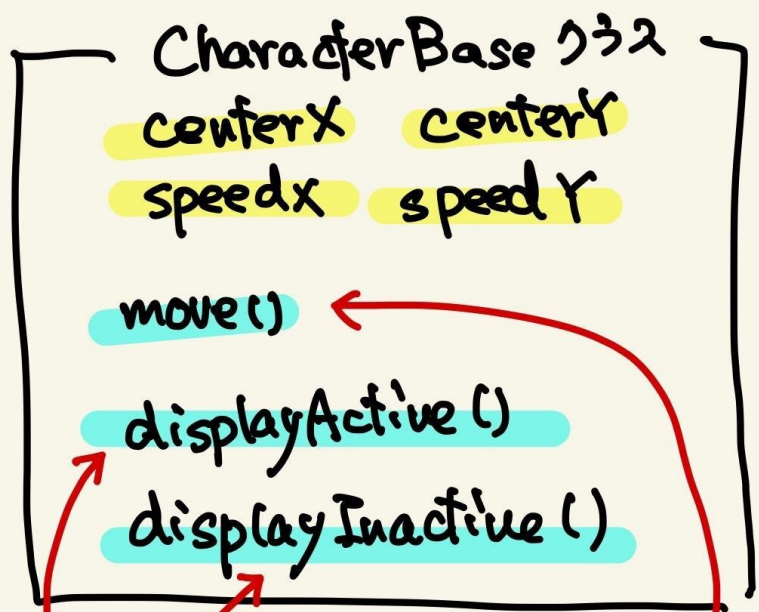
- Hand クラスを継承し、グーの役割を果たすRockクラスとチョキの役割を果たすScissorsクラス、パーの役割を果たすPaperクラスを作成せよ。
 - 各コンストラクタでtypeを指定せよ
 - displayクラスをオーバーライドすることで、isAliveがtrueのときに、グーチョキパーそれぞれのオブジェクトを表示するようにせよ（文字で表示しても良い）
 - 大きさは直径30ピクセル程度とせよ
 - グーとパーは壁で跳ね返り、チョキは壁にくると逆から出てくるようにせよ
- それぞれ30個ずつ画面にランダムに配置し、速度はxy方向それぞれ-2~+2の間でランダムにせよ
 - setDataで設定せよ

課題3-3:basic_CharacClass



- 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，画面の外に出ないないようにせよ
- なお配布プログラムではCharacterBaseクラスのキャラクタ（□）と、ExNakamura334クラスのキャラクタを表示しているが、自分で作成したクラス（継承したクラス）のキャラクタをExNakamura334クラスのキャラクタに置き換えて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（来週配布します）
 - （例）X-3-34 中村の場合、クラス名を「ExNakamura334」とする

CharacterBaseの継承？

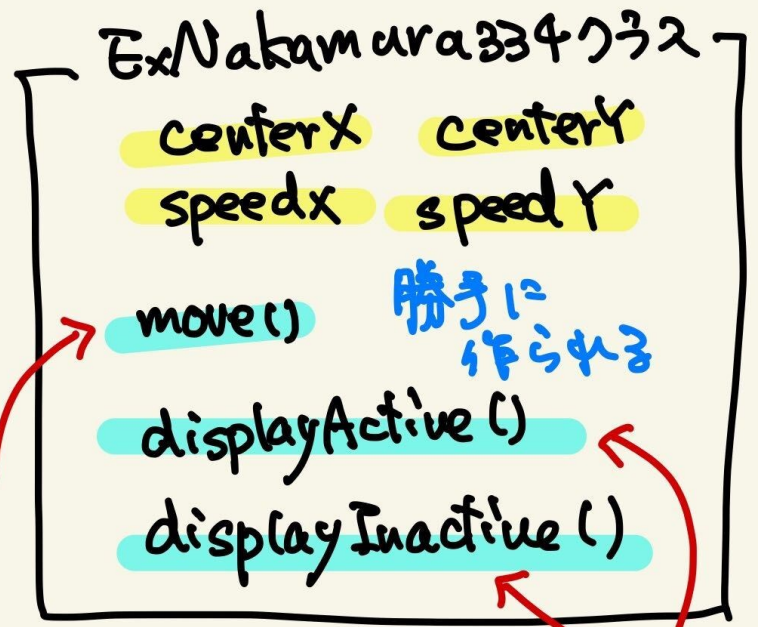
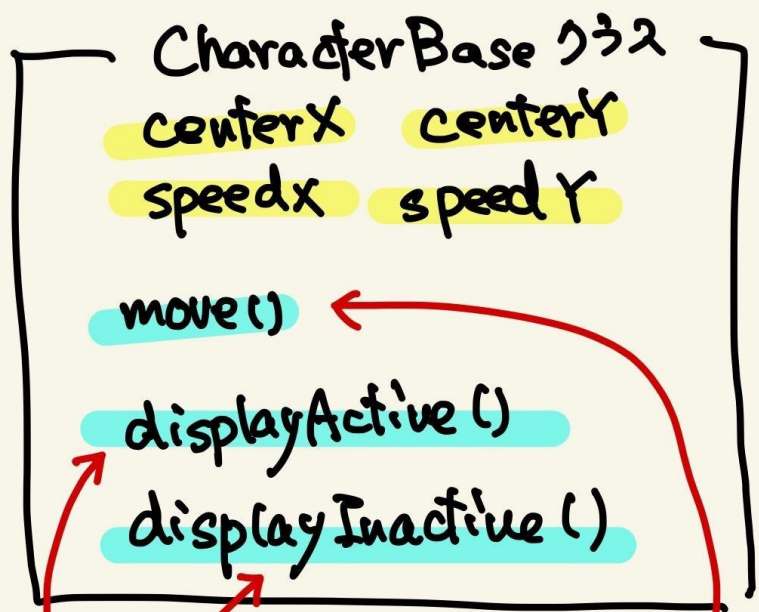


speedx, speedY を使って
centerX, centerY を動かす

アクティブなキャラクタを描画, (□ が描画できる)

アクティブじゃないキャラクタを描画 (■ が描画できる)

CharacterBaseの継承？

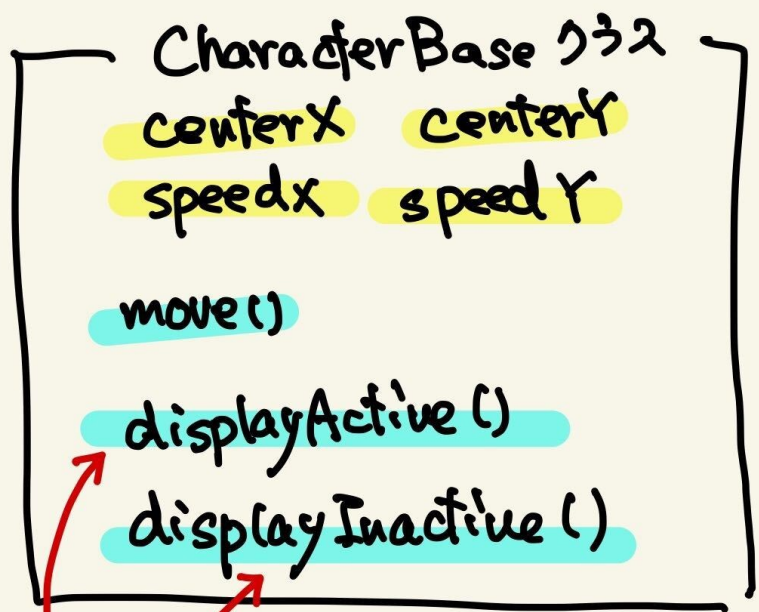


speedx, speedY を使って
centerX, centerY を動かす

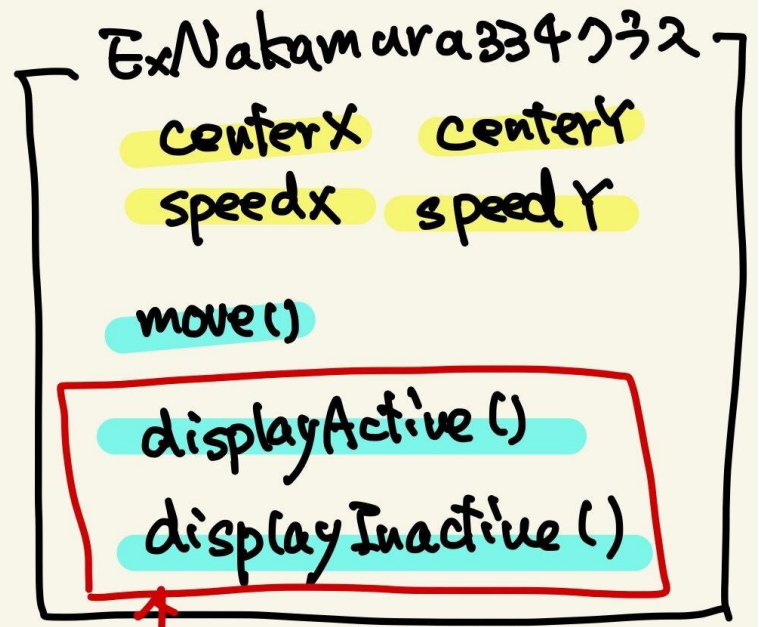
アクティブなキャラクタを描画, (□ が描画される)

アクティブじゃないキャラクタを描画 (■ が描画される)

CharacterBaseの継承？



継承



こいつはオーバーライドして独自のものを作ります

□ が描画はできる
■ が描画はできない

□ と ■ がは困る

CharacterBaseの継承？



CharacterBase のクラス

- centerX centerY
- speedx speedY
- move()
- displayActive()
- displayInactive()

継承

ExNakamura334 のクラス

- centerX centerY
- speedx speedY
- move()
- displayActive()
- displayInactive()

- が 描画 します
- が 描画 します

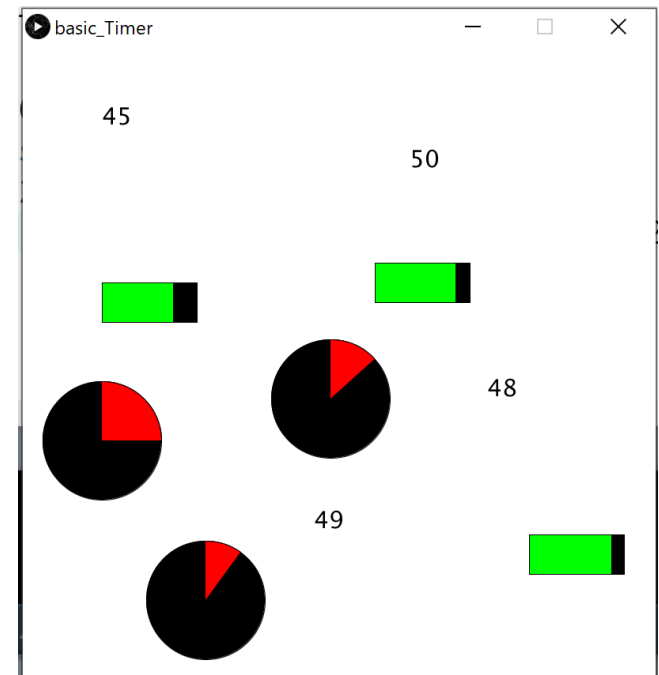
こっちはマシがえやば
よいの？

継承の意味がない!!

宿題3-1: hw_Timer



- 配布する hw_Timer.zip のTimerCoreクラスを継承し、TextTimerやProgressTimerを参考にしつつタイマーをクリックしてから60秒の経過を計測する、新しいタイマークラスを作成せよ。なお新しいタイマーは2つ以上実現し、それぞれ3つずつ以上表示せよ
- なお、タイマー内をマウスクリックしたときにそれぞれのタイマーがスタートするようにせよ



宿題3-2: hw_JankenGuriko



- 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_VendingMachine



- 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3-3: hw_VendingMachine



- 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3-3: 動作チェック



- 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