



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

課題4-1: basic_ManyChara



- 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ー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ーが別の表示スタイルになるようにせよ
 - 押していない間は `displayInactive` を, 押している間は `displayActive` 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ークラスが入ったファイルを研究室のSlackにアップせよ
 - 同じ研究室のものを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)、キャラクター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て、もっと多くのキャラクターを入れてもよい

(ヒント) ArrayListに放り込むだけ!!

課題4-2: basic_ClickRandom



- 800x600のウィンドウ内をクリックするたびに、そのクリックした場所に、赤色の○か青色の□のどちらかが生成され、○と□が動き回るプログラムを作成せよ
 - ただし、コンストラクタの中ではmouseX, mouseYという変数を使わないようにせよ
 - ○は上下左右で跳ね返るようにし（速度は-5以上5未満の実数値）、○は生成されたときにランダムに5以上60未満の間で直径が決まるようにせよ
 - □は端に来ると反対から出てくるようにし（速度は-5以上5未満の実数値）、□は生成されたときにランダムに10以上40未満の間で一辺の長さが決まるようにせよ

課題4-3: basic_Celestial



- basic_Celestialを利用し、天体の動きっぽいものを作成せよ。配布プログラムの中には、CelestialBodyクラスを継承したSunクラス、Planetクラスがある
- SunやPlanetクラスを真似しつつ、CelestialBodyクラスを継承し、Satelliteクラスを作成せよ
 - Satelliteクラスは、コンストラクタの引数としてその衛星の惑星Planetに該当するもののみを取るようにせよ。
 - コンストラクタ内で、distanceを30以上50未満の値で、angleSpeedを0.02以上0.08未満の値でそれぞれランダムに設定せよ。
 - 半径は固定で2、bodyColorとして白色を設定せよ。
- 作成したプログラムを利用し、下記を描画せよ
 - Sunから150の距離に半径10の惑星（速度は0.01）と1つの衛星
 - Sunから250の距離に半径10の惑星（速度は0.008）と2つの衛星
 - Sunから350の距離に半径20の惑星（速度は0.006）と72個の衛星

宿題4-1: hw_Matrix3x3



- 3x3の行列を2次元の配列として管理し、行列の値をdisplayメソッドで表示するクラスMatrix3x3がある。
- この関数を拡張し、行列式を求める関数determinant()、対角和を求める関数trace()、ユークリッドノルム（平方和の平方根）を求める関数norm()を作成せよ。なお、いずれの関数も返り値の型はfloatとせよ。

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

の行列式 (determinant) は、次のように求めることができます：

$$\det(A) = a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31})$$

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

この行列Aの主対角和 (trace) は次のように求められます：

$$\text{Trace}(A) = a_{11} + a_{22} + a_{33}$$

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}$$

協力ChatGPTさん

Matrix:

```
2.0 -1.0 0.0  
-1.0 2.0 -1.0  
0.0 -1.0 -2.0
```

行列式 (det) = -8.0

対角和 (trace) = 2.0

ユークリッドノルム (norm) = 4.0

宿題4-2: hw_JankenBattle



- 過去に作ったbasic_JankenBattleを改良し、800x600のウィンドウ内をクリックするたびに、そのクリックした場所に応じて、グーの役割を果たすRockクラス、チョキの役割を果たすScissorsクラス、パーの役割を果たすPaperクラスのインスタンスを作成して表示せよ。
 - なお、ウィンドウの上部3分の1でクリックされたらグー、中央の3分の1でクリックされたらチョキ、下部3分の1でクリックされたらパーとせよ
 - 各コンストラクタでm_typeを指定せよ
 - 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
 - グーの速度はxy方向それぞれ-2以上2未満、チョキの速度はxy方向それぞれ-5以上5未満、パーの速度はxy方向それぞれ-10以上10未満のランダムな値とせよ

宿題4-3: 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4-3: 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4-3: 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