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，高橋，小林，橋本

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Slack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_ClickRandom

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

- 800x600のウイ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-5～5の実数値）、○は生成されたときにランダムに5～60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-5～5の実数値）、□は生成されたときにランダムに10～40の間で一辺の長さが決まるようにせよ

課題4-3: basic_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．なおAAAAAAはAAAを2回とカウントせよ
    - 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

課題4-3: basic_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"));
}
```

課題4-3: basic_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4-1: hw_ClickGame

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□、同じく独特の動きをする5個の×をクリックにより倒すゲームを作成せよ（ただしそれぞれの動きは違うものとせよ）
- なお、すべての敵が消滅すると画面上にクリアと表示するようにせよ

課題4-2: 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

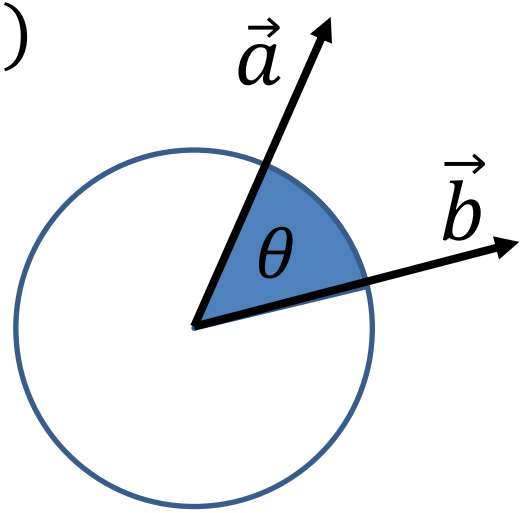
```
void setup() {
  VectorOperation vo = new VectorOperation();
  float[] vectorA = {10, 0, 1, 9};
  float[] vectorB = {8, 1, 1, 4};
  vo.setVectorA(vectorA);
  vo.setVectorB(vectorB);

  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());
}
```

コサイン類似度

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 $\vec{a}$ と $\vec{b}$ のなす角度を利用した類似度

$$\vec{a} = (a_1, a_2) \text{ と } \vec{b} = (b_1, b_2)$$



- ベクトルの内積の計算は...

$$\vec{a} \cdot \vec{b} = |\vec{a}||\vec{b}|cos\theta$$

$$cos\theta = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|} = \frac{a_1 \cdot b_1 + a_2 \cdot b_2}{\sqrt{a_1^2 + a_2^2} \cdot \sqrt{b_1^2 + b_2^2}}$$

コサイン類似度：3次元の場合

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 $\vec{a}$ と $\vec{b}$ のなす角度を利用した類似度

$$\vec{a} = (a_1, a_2, a_3), \vec{b} = (b_1, b_2, b_3)$$

$$cos\theta = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|} = \frac{a_1 \cdot b_1 + a_2 \cdot b_2 + a_3 \cdot b_3}{\sqrt{a_1^2 + a_2^2 + a_3^2} \cdot \sqrt{b_1^2 + b_2^2 + b_3^2}}$$

コサイン類似度：N次元の場合

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 $\vec{a}$ と $\vec{b}$ のなす角度を利用した類似度

$$\vec{a} = (a_1, a_2, a_3 \ldots, a_N), \vec{b} = (b_1, b_2, b_3 \ldots, b_N)$$

$$cos\theta = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|} = \frac{\sum_{i=1}^{N} a_i \cdot b_i}{\sqrt{\sum_{i=1}^{N} a_i^2} \cdot \sqrt{\sum_{i=1}^{N} b_i^2}}$$

三角形の面積

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 $\vec{a}$ と $\vec{b}$ のからなる三角形の面積Sは下記の式で求めることができる

$$S = \frac{1}{2}\sqrt{|\vec{a}|^2\left|\vec{b}\right|^2 - (\vec{a}\cdot\vec{b})^2}$$

  - 三次元の場合はこんな感じ

$$\vec{a} = (a_1, a_2, a_3), \vec{b} = (b_1, b_2, b_3)$$

$$S = \frac{1}{2}\sqrt{|\vec{a}|^2\left|\vec{b}\right|^2 - (\vec{a}\cdot\vec{b})^2}$$

$$= \frac{1}{2}\sqrt{(a_1^2 + a_2^2 + a_3^2)(b_1^2 + b_2^2 + b_3^2) - (a_1\cdot b_1 + a_2\cdot b_2 + a_3\cdot b_3)^2}$$

宿題4-3: hw_ImageProc

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

- PImage クラスを使って画像処理をしよう
  - 適当な画像をダウンロードし，その画像に対してフィルタを掛けてみましょう！
  - 左上にオリジナル画像，右上にTHRESHOLDで二値化したもの，左下にBLURでぼかしたもの，右下にINVERTでネガポジ反転したものを表示しよう