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
ArrayList, HashMap

中村，青山，小林，橋本

中間試験について

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

- 11月6日 4限目に203, 206教室にて実施
  - 3限目から部屋を使えるので3限目に来ましょう
  - 何らかの理由により欠席する場合は，【事前】にいずれかの先生に連絡すること．また，病欠の場合は医師の診断書を取ること．
  - 試験は4限開始なので，電車が止まったとかそういう理由は（1日じゅう止まらない限り）認めません
- 試験範囲
  - 前期，後期（ArrayListとHashMapを除く）分から
  - 小テスト（をいじったもの）からそれなりに出しますので小テストをチェック！

課題4-2 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

- 他者のクラスを使って遊ぶ
- 資料配布フォルダに、各自が作成したクラスファイルを置くため、それを利用して5つ以上のキャラクタが画面内を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 display1 を，押している間は display2 を利用するようにすると良い

課題4-3 BTR

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

- Objectクラスを継承し，
  - コンストラクタで塗りつぶし色がランダムに決定される○を描画するBallクラス
  - 青色の△を描画するTriangleクラス
  - 赤色の□を描画するRectangleクラス
  - を作成せよ．なお，Ballクラスは上下で跳ね返り（左右はワープ），Triangleクラスは左右で跳ね返り（上下はワープ），Rectangleクラスは上下左右すべてで跳ね返るようにせよ
  - また，上記クラスを用いて50個の○，50個の△，50個の□が動き回るプログラムを作成せよ

解答例


```
Ball [] balls = new Ball [50];
Triangle [] triangles = new Triangle [50];
Rectangle [] rects = new Rectangle [50];

void setup() {
  size( 400, 400 );
  for ( int i=0; i<50; i++ ) {
    rects[i] = new Rectangle();
    balls[i] = new Ball();
    triangles[i] = new Triangle();
  }
}

void draw() {
  background(255);
  for ( int i=0; i<50; i++ ) {
    rects[i].move();
    balls[i].move();
    triangles[i].move();
    rects[i].display();
    balls[i].display();
    triangles[i].display();
  }
}
```

なんかやっぱり
無駄な気がする

Array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

- Array型は同じ型のものを複数管理するクラス
  - `String [] students = new String [101];`
  - `int [][] lights = new int [100][100];`
  - `Ball [] balls = new Ball [50];`

- 違う形のオブジェクトをどう管理したら良い？
  - 先週やった色々な ExName 型（色々なキャラクタ）をまとめて扱いたい！

実を言うと

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

- 親クラスが一緒で呼び出すメソッドが同じならまとめて配列に！

```
ObjectChara [] charas
            = new ObjectChara [4];

void setup(){
  size(800,600);
  charas[0] = new ExKomatsu();
  charas[1] = new ExHashimoto();
  charas[2] = new ExKobayashi();
  charas[3] = new ExNakamura();
}

void draw(){
  background(255);
  for( int i=0; i<charas.length; i++ ){
    charas[i].move();
    if( mousePressed ){
      charas[i].display2();
    } else {
      charas[i].display1();
    }
  }
}
```

実を言うと

- 親クラスが一緒で呼び出すメソッドが同じならまとめて配列に！

```
Object [] objs = new Object [150];

void setup() {
  size( 400, 400 );
  for ( int i=0; i<50; i++ ) {
    objs[i] = new Ball();
  }
  for ( int i=0; i<50; i++ ) {
    objs[i+50] = new Rectangle();
  }
  for ( int i=0; i<50; i++ ) {
    objs[i+100] = new Triangle();
  }
}

void draw() {
  background(255);
  for ( int i=0; i<150; i++ ) {
    objs[i].move();
    objs[i].display();
  }
}
```

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

- 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各種のインスタンスは，それぞれ親クラスの型に代入することができる．呼び出されるのは，その継承されたクラスのメソッドとなるため，親クラスにそのメソッドがないとエラーになる
- 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，自動的にアップキャストされる
  - ダウンキャストは問題だけれど，アップキャストは基本的に問題なし（全部継承されるため）

数の定義面倒

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

- 配列の大きさを事前に指定しておくのは柔軟性がなくて面倒！
- もっと柔軟にオブジェクト集合を扱いたい！

```
ObjectChara [] charas
            = new ObjectChara [4];

void setup(){
  size(800,600);
  charas[0]=new ExKomatsu();
  charas[1]=new ExHashimoto();
  charas[2]=new ExKobayashi();
  charas[3]=new ExNakamura();
}

void draw(){
  background(255);
  for( int i=0; i<charas.length; i++ ){
    charas[i].move();
    if( mousePressed ){
      charas[i].display2();
    } else {
      charas[i].display1();
    }
  }
}
```

ArrayList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

- ArrayList型は，可変で色々なものを格納できるクラス

  （例） ArrayList list = new ArrayList(); で定義

  - ArrayListの要素数を取得
    - list.size();
  - ArrayListに対するaddで要素を追加
    - list.add(value); // valueを追加
  - ArrayListに対するremoveで要素を削除
    - list.remove(index); // index番目を削除
  - ArrayListに対するgetで要素を取得
    - list.get(index); // index番目のオブジェクトを取得

ArrayList + Object

- ArrayList型の定義
- add()でリストに追加
- size()でリストのアイテム数を取得
- get()でリストからアイテム(オブジェクト)を取得

```
ArrayList list = new ArrayList();
void setup(){
  size(800,600);
  list.add( new ExKomatsu() );
  list.add( new ExHashimoto() );
  list.add( new ExKobayashi() );
  list.add( new ExNakamura() );
}
void draw(){
  background(255);
  for( int i=0; i<list.size(); i++ ){
    ObjectChara chara;
    chara = (ObjectChara)list.get(i);
    chara.move();
    if( mousePressed ){
      chara.display2();
    } else {
      chara.display1();
    }
  }
}
```

ArrayList + Object

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 list = new ArrayList();

void setup() {
  size( 400, 400 );
  for ( int i=0; i<50; i++ ) {
    list.add( new Rectangle() );
    list.add( new Ball() );
    list.add( new Triangle() );
  }
}

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

```
for ( int i=0; i<50; i++ ) {
  Rectangle rt = new Rectangle();
  list.add( rt );
  Ball ball = new Ball();
  list.add( ball );
  Triangle tri = new Triangle();
  list.add( tri );
}
```

list.size() で数を取得

list.get(i) でi番目を取得

(Object) で Object 型と明示

HashMap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

- HashMap型は，何かをキーとして値を格納するものであり，キーを利用して値を取り出せる

  HashMap<キーの型, 格納する値の型> dic;

  dic = new HashMap<キーの型, 格納する値の型>();

  - HashMapの要素数を取得
    - dic.size();
  - HashMapに対する要素を追加
    - dic.put(key, value); // keyでvalueを追加
  - HashMapに対するgetで要素を取得
    - dic.get(key); // keyに該当する値を取得

HashMap

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

- 辞書みたいなもの
- 何かのキーから値を取得する

```
HashMap<String, String> aadic
      = new HashMap<String, String>();

void setup() {
  aadic.put( "haa", "(°Д°)ﾊｧ?" );
  aadic.put( "ase", "( ; ´Д｀ )" );
  aadic.put( "kita", "ｷﾀｰ(°∀°)ｰ!" );
}

void draw() {
  background(255);
  println( aadic.get("haa") );
  println( aadic.get("ase") );
  println( aadic.get("kita") );
}
```

HashM

```
HashMap<String, RootChara> dict;
dict = new HashMap<String, ObjectChara>();

void setup() {
  dict.put( "hashimoto", new ExHashimoto() );
  dict.put( "komatsu", new ExKomatsu() );
  dict.put( "kobayashi", new ExKobayashi() );
  dict.put( "nakamura", new ExNakamura() );
}

void draw() {
  background(255);
  ObjectChara chara;
  chara = (ObjectChara)dict.get( "hashimoto" );
  chara.display();
  chara = (ObjectChara)dict.get( "komatsu" );
  chara.display();
  chara = (ObjectChara)dict.get( "kobayashi" );
  chara.display();
}
```

課題

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

- 課題5-1: ArrayChara
  - 先週のキャラクタ課題（他人のEx***を使う）を単純な配列で実現せよ．ただしキャラクタは5個以上とする
- 課題5-2: ArrayListChara
  - 同じく上記プログラムをArrayListを使って実現せよ．ただしキャラクタは10個以上とする
- 課題5-3: ArrayObject
  - 先週のObjectクラスを継承して〇、△、□を動かすプログラムについて、〇は50個、△は40個、□は30個とし、×も20個表示して動き回るプログラムを、ArrayListまたは配列で実現せよ。なお、×の跳ね返りなどについては自由でOK。