ネットワークプログラミング応用演習

2132023番

内山　裕太

1. 目的

本講義で学んだ知識を再確認し，javaプログラミングの技術を向上させ，

実力確認試験などに対応できるようにする．

1. 課題

講義中に配布されたプログラムをもとに，

第7回講義資料（P107）中の☆1,2,3,4,6,7,9を実装した．

また，オリジナル拡張として「コマンドライン引数を用いた色設定」，

「自由な絵を描写できる機能」，「コマンドライン引数に応じて描写する図形を変更」を実装した．

1. 理論

特になし．

1. プログラム（ソースコード）

ソースコードは別途，zipファイルにて添付する．

今回用いた技術をソースコードの内容に合わせて記述する．

なお，配布プログラムの改良であるため，最後に変更点をまとめて記述する．

* 1. 静的変数（static変数）

☆1,2,3,6,9を実装するために用いた．

Paint2クラス15行目．　　Figureクラス6行目．である．

コマンドライン引数の1つ目に渡した値がstatic変数のiに渡される

（Paint2クラス20行目）．引数の値に応じて画面上に表示される図形の個数を変更することができる（Paint2クラス61,62,63行目）．

コマンドライン引数の2つ目に渡した値がstatic変数のkに渡される

（Paint2クラス21行目）．引数の値に応じて描画される図形が変化するようになっている（Paint2クラス53,54,55行目）．

また，Paint2クラス15行目にて宣言されているstatic変数のjは，

描画されている図形の個数を出力するために用いている．

* 1. 型変換（Integer.parseInt(str)）

☆2,7,9を実装するために用いた．

Paint2クラス20,21,23,24,25行目．

int型であるstatic変数にstr型であるコマンドライン引数を代入する場合は型変換を行う必要がある．そこで，IntegerクラスparseIntメソッドを

用いた．これにより，4-1の内容に加え，描画する図形の初期色を設定可能とした．コマンドライン引数の3,4,5つ目で設定することができる．

3つ目はR成分，4つ目はG成分，5つ目はB成分を指定する．

* 1. 乱数

☆7を実装するために用いた．

Paint2クラス77行目．

4-2では図形の初期色を設定することができたが，マウスカーソルを

ウィンドウの外側に移動させるたびに，色を構成する3成分がランダムで設定されるようになっている（Paint2クラス78,79,80行目）．

* 1. そのほかの変更点
     1. Paint2クラス

コマンドライン引数の2つ目の値に応じたクラスを呼び出すようになっている（53,54,55行目）．

55行目では，コマンドライン引数の2つ目の値が3の場合に，自由な絵を描写できるようになっている．そのため，76行目のmouseDraggedメソッドでは，マウスカーソルがドラッグされるたびに線を描写する処理を実行している．

* + 1. Figureクラス

色のRGB成分を決める3つの変数をstatic変数にすることで，Paint2

クラスから直接色の変更を可能とした．

* + 1. Circleクラス

Circleメソッドへ引数を渡し，その値によって図形のサイズを変更するようにした．

* + 1. Rectクラス

新たに四角形を描画するクラスとして追加した．Rectクラスへ引数を

渡し，その値によって図形のサイズを変更するようにした．

* + 1. Lineクラス

コマンドライン引数の2つ目に3を渡した場合に使用するクラスである．

始点と終点をマウスカーソルがある座標で統一した．

Graphics2Dクラスを用いて，線の太さを変更している．

1. 動作確認手法・環境

今回の動作環境として，eclipseを用いた．

以下，各課題の確認方法を記述する．

* 1. ☆1

アプリケーションを実行し，コンソールを用いて確認する．

マウスをクリックするたびに図形が描写されるため，クリックと同時にコンソールに数値が出力されれば動作していることとなる．

* 1. ☆2

アプリケーションを実行し，異なる座標に図形を5つほど描画したのちに，コマンドライン引数の1つ目で指定した個数まで図形を同じ座標へ描画する．その後，1つ描画するごとに最も古い図形から消える．

最初の5つ以降を同じ座標に描画することで，図形が消えるのを確認しやすくしている．なお，この課題は図形が消えたことを確認できた場合，動作していることとなる．

* 1. ☆3

アプリケーションを実行し，図形を描画するたびに図形のサイズが変動することを確認する．今回は2種類のサイズを用意しているため，コンソールに表示されている数値が奇数の場合と偶数の場合でサイズが異なることを確認できた場合，動作していることとなる．

* 1. ☆4

コマンドライン引数の3,4,5つ目で色を指定することで，描画される図形の色が変化することを確認する．描画する図形に任意の色がついていれば動作していることとなる．

* 1. ☆6

コマンドライン引数の2つ目の値によって，描画する図形が変化することを確認する．与える値が1の場合は円，2の場合は四角形，3の場合は

自由描画である．任意の値を渡し，対応する図形の描画がされれば動作していることとなる．

* 1. ☆7

マウスカーソルをウィンドウの外側にずらし，再度戻して描画すると

色が変化することを確認する．再度描画した際に，画面上の図形の色が

変化すれば動作していることとなる．

1. グラフィカル ユーザー インターフェイス が含まれている画像

   自動的に生成された説明グラフィカル ユーザー インターフェイス

   低い精度で自動的に生成された説明実行結果

上図は，5つ目までを上部に描画し，それ以降を1か所に描画した結果の

スクリーンショットである．描画した図形の個数がコンソールに出力されていることを確認できる．また，図形のサイズが変更されることや指定された個数に達した場合に古い図形から削除されることも確認できる（上手では画面上に20個まで表示するよう設定している）．

グラフ, 散布図

自動的に生成された説明グラフ, 散布図

自動的に生成された説明上図はコマンドライン引数のグラフ, 散布図

自動的に生成された説明3,4,5つ目の値を変更した結果である．

左から（255,0,0）（0,255,0）（0,0,255）で与えている．この結果より，

ロゴ, 会社名

自動的に生成された説明グラフ, 散布図

自動的に生成された説明グラフ, 散布図

自動的に生成された説明コマンドライン引数による初期色の設定を確認できる．

上図はコマンドライン引数の2つ目を変更した結果である．

グラフ

自動的に生成された説明左から（1）（2）（3）を与えている．与える値によって描画する図形が変更されることを確認できた．

アプリケーション が含まれている画像

自動的に生成された説明上図はマウスカーソルをウィンドウの外に移動させたのちに再度描画した結果である．mouseExitedメソッドが動作していることを確認できる．

1. まとめ

本課題を通して，配布プログラムをある程度理解することができた．最も

理解が深まったと思う点は，クラス間の継承や静的変数である．また，

レポートを書くにあたり，再度レポートの書き方について学んだこともあった．Overrideなど，まだ理解が足りない箇所も多く存在するため，実力確認試験に向けて再度復習を進めようと思う．本課題も，最初は何もわかっていなかったが，初回の講義資料から見直し，インターネットなどを用いて

用いるべきクラスなどを調べ，用いることでレポート作成ができたため，

わからないときは最初から見直してみることが大切であることを学んだ．