

MyLib (2015年度)

平成 27 年 11 月 27 日

1 全体構成

「モデリングとシミュレーション実験」では、各課題で共通的に必要な機能を MyLib というライブラリにまとめ、配布しています。この文書は、その全体を説明することを目的とします。

MyLib は、3 個のパッケージで構成されています。

- `myLib.utils`: リストの生成、ファイルの利用などの汎用的な機能を `static` メソッドとして実装しています。 `static` メソッドですから、クラスインスタンスを生成せずに利用できます。
- `myLib.rungeKutta`: 連立常微分方程式の数値的解法である 4 次の Runge Kutta 法と、連立常微分方程式で記述される力学システムのテンプレートクラスを格納しています。
- `myLib.gui`: Java Swing を使った表示のためのテンプレートクラスを格納しています。

2 `myLib.utils` パッケージ

2.1 `Utils` クラス

リストの生成など汎用的に利用できる機能を `static` メソッドとして実装しています。各メソッドは表 1 に示します。

表 1: Utils クラスのメソッド。なお T、K 及び V はクラステンプレート

修飾子と型	メソッドと説明
static <T> List<T>	createList() クラス T を格納する空のリストを生成する。
static <K, V> Map<K, V>	createMap() クラス K から V への空の写像を生成する。
static int[]	createRandomNumberList(int limit, int num) 整数 (値が 0 から limit-1) をでたために並べ替え、そのうち num 個からなる配列を生成する。
static int[]	createRandomNumberList(int limit) 整数 (値が 0 から limit-1) をでたために並べ替えた配列を生成する。
static List<Integer>	createRandomNumberList(int limit) 整数 (値が 0 から limit-1) をでたために並べ替えたリストを生成する。
static <T> Set<T>	createSet() クラス T を格納する空の集合を生成する。
static String	Object2String(char sep, Object... objects) objects で指定される任意の個数のオブジェクトを文字列に変換し、区切り文字 sep で連結した文字列を生成する。オブジェクトを文字列に変換する際には、そのクラスで定義された toString() メソッドを用いる。

2.2 FileIO クラス

ファイルへの書き込みなどの機能を static メソッドとして実装しています。各メソッドは表 2 に示します。

表 2: FileIO クラスのメソッド

修飾子と型	メソッドと説明
static BufferedReader	openReader(String filename) filename で指定されたファイルを開き BufferedReader を返す。ファイルが開けない場合には IOException をスローする。
static BufferedWriter	openWriter(String filename) filename で指定されたファイルを開き BufferedWriter を返す。ファイルが開けない場合には IOException をスローする。
static String[]	readCSV(BufferedReader in) in から一行読みだし、カンマで区切って、結果を文字列配列で返す。読みだせなかった場合には IOException をスローする。
static String[]	readSSV(BufferedReader in) in から一行読みだし、スペースで区切って、結果を文字列配列で返す。読みだせなかった場合には IOException をスローする。
static String[]	readVars(BufferedReader in, String sep) in から一行読みだし、正規表現で表された sep で区切って、結果を文字列配列で返す。読みだせなかった場合には IOException をスローする。
static void	writeCSV(BufferedWriter out, Object... objects) out へ向けて、objects で指定される任意の個数のオブジェクトを文字列に変換し、カンマで区切って出力する。書けなかった場合には IOException をスローする。
static void	writeSSV(BufferedWriter out, Object... objects) out へ向けて、objects で指定される任意の個数のオブジェクトを文字列に変換し、スペースで区切って出力する。書けなかった場合には IOException をスローする。
static void	writeSSV(BufferedWriter out, char sep, Object... objects) out へ向けて、objects で指定される任意の個数のオブジェクトを文字列に変換し、区切り文字 sep で区切って出力する。書けなかった場合には IOException をスローする。

3 myLib.rungeKutta パッケージ

3.1 DifferentialEquation インターフェイス

一階連立微分方程式に対応した関数型のインターフェイスです。

表 3: DifferentialEquation クラスのメソッド

修飾子と型	メソッドと説明
double[]	rhs(double t, double y[]) 独立変数 t と従属変数 y[] を与えて、その時の従属変数の一階微分を戻り値とする。

3.2 RungeKutta クラス

一階連立微分方程式を解く 4 次の Runge Kutta 法に対応する機能を static メソッドで実装したクラスです。

表 4: RungeKutta クラスのメソッド

修飾子と型	メソッドと説明
static double[]	rk4(double x, double y[], double h, DifferentialEquation eq) 一階連立微分方程式 eq を独立変数 x とその時の従属変数 y[] を与えて、h だけ進んだ時の従属変数の値を返す。
static double[][]	rkdumb(double vstart[], double x1, double x2, int nstep, DifferentialEquation eq) 一階連立微分方程式 eq を、従属変数の初期値 vstart[] を与え、独立変数を x1 から x2 まで、nstep に分割して解く。その間の従属変数の値 nstep 個を返す。戻り値配列の第一インデックスが変数インデックス、第二インデックスは値 nstep を表す。

3.3 Dynamics クラス

Runge-Kutta 法で連立微分方程式を解く場合の、力学系のクラスです。

表 5: Dynamics クラスのコンストラクタ

コンストラクタと説明
Dynamics(double... initials) 従属変数を initials に与えて初期化する。独立変数は 0 となる。

表 6: Dynamics クラスのメソッド

修飾子と型	メソッドと説明
List<Point2D.Double>	evolution(double h, int nstep, int vNum) 独立変数を h だけ、nstep に分割して進める。戻り値は独立変数と vNum で指定される従属変数を Point2D.Double で組にしたリストである。
List<Point2D.Double>	evolution(double h, int nstep) 独立変数を h だけ、nstep に分割して進める。戻り値は独立変数と 0 で指定される従属変数を Point2D.Double で組にしたリストである。
List<Point2D.Double>	evolution(double h, int nstep, int xNum, int yNum) 独立変数を h だけ、nstep に分割して進める。戻り値は xNum と yNum で指定される従属変数を Point2D.Double で組にしたリストである。
double[]	initialize() 全ての従属変数を初期値に戻す。
void	setEquation(DifferentialEquation equation) 連立微分方程式 equation を設定する。
void	setT(double t) 独立変数 t の値を設定する。
double[]	update(double dt) 独立変数を dt だけ進める。

4 myLib.gui パッケージ

4.1 DrawBase クラス

描画を行うための抽象クラスです。javax.swing.JPanel の拡張クラスであるとともに、Runnable を実装しています。

表 7: DrawBase クラスのコンストラクタ

コンストラクタと説明
DrawBase() javax.swing.JPanel のコンストラクタを呼び、初期化する。

表 8: DrawBase クラスのメソッド

修飾子と型	メソッドと説明
protected void	clearImage() BufferedImage のインスタンス image を背景色で塗りつぶす。
protected void	initImage() パネルの大きさを計測し、イメージを初期化する。
boolean	isRunning() Runnable に対応し、動作中かを返します。
abstract void	mkImage() BufferedImage のインスタンス image に、実際に図を描くための抽象メソッドです。実装が必要です。
void	paint(Graphics g) パネルに描画されたイメージを背景色で塗りつぶした後、g にイメージ image を描きます。
void	run() Runnable の必須メソッドです。イメージをクリアし、イメージを作成し、描画します。描画後 sleep だけ休止します。
void	saveImage(String filename) filename で指定されたファイルに BufferedImage のインスタンス image を png ファイルとして保存します。書き込めない場合には、IOException をスローします。
void	start() Runnable に対応し、起動可能とします。この後、本クラスのインスタンスを Thread に渡します。
void	stop() Runnable に対応し、停止します。