

3. 実験で使用するプログラミング手法およびクラス

実験では課題を解くプログラムを Java により作成する。本実験の内容は、基本的には四則演算と繰り返し処理および計算結果の表示についてプログラムが作成できれば良い。以下に、本実験で使用するおもなクラスを挙げる。詳細は JavaDoc を参照のこと。

- ・ Scanner クラス

キーボードから入力された文字を読み取るために使用する。Scanner オブジェクトの `nextDouble()` メソッドで実数値、`nextInt()` メソッドで整数値として読み込む。

(以下、検討事項で使用するクラス)

- ・ FileWriter クラス

新しいファイルを作成したり、既に存在するファイルへテキストデータを追加するために使用する。

- ・ BufferedWriter クラス

文字をバッファリングすることによって、文字、配列、または文字列を効率良く文字型出力ストリームに書き込むために使用する。

- ・ PrintWriter クラス

オブジェクトの書式付き表現をテキスト出力ストリームに出力するために使用する。

FileWriter クラス、BufferedWriter クラス、PrintWriter クラスを合わせて使用し、数値計算の結果をテキストファイルに出力することができる。

4. 実験課題

以下の実験課題について、プログラムを入力または作成しなさい。プログラムを実行し、その結果を確認して記録すること。

4 - 1. 実数値の入出力

実数値の入出力を行うプログラム (EX4_1.java) を以下に示す。これを入力し実行結果を確認しなさい。

```
/* Ex4_1.java 実数値の入力と出力 */
import java.util.Scanner;
public class Ex4_1{
    public static void main(String[] args){
        System.out.print("実数値を入力してください:");
        Scanner scan = new Scanner(System.in); // キーボード入力の準備
        double num = scan.nextDouble(); // キーボードから実数値を入力
        System.out.println("入力した実数値は" + num + "です。");
    }
}
```

4 - 2. 数値計算における丸め誤差

実数値 0.1 を 10 回加算し、その結果を表示するプログラム (Ex4_2.java) を以下に示す。これを入力し実行結果を確認しなさい。

```
/* Ex4_2.java 数値計算における丸め誤差 */
public class Ex4_2{
    public static void main(String[] args){
        int i;
        double sum;
        System.out.println("丸め誤差の確認");
        sum = 0.0;
        for(i=1; i<=10; i++){
            sum = sum + 0.1;
            System.out.println("i:" + i + " sum:" + sum);
        }
    }
}
```

4 - 3. 区分求積法による積分

区分求積法を用い、以下の定積分の結果 S を求め表示するプログラムを作成しなさい。
刻み幅 h は 0.1 とする。

$$S = \int_0^1 3x^2 dx$$

区分求積法のプログラム (Ex4_3.java) の一部を以下に示す。これを基にプログラムを作成すること。

```

/*****
/*   Ex4_3. java   区分求積法プログラム (未完成)   */
*****/

public class Ex4_3{
    public static void main(String[] args) {
        double x, y;
        double h;
        double ds , s;

        h = 0.1;
        x = 0.0;
        s = 0.0;

        for(int i=0; i<(int) (1/h); i++) { //h が 0.1 のとき 10 個の矩形を計算
            y = ??; // y=3x^2 を計算 (^2 は 2 乗を意味する)
            ds = ??; //一つの矩形領域の面積を計算
            s = ??; //定積分値に矩形面積を加える
            x = ??; //x を刻み幅の分だけ増やす
        }
        System.out.println(s); //面積の出力
    }
}

```

5. 検討事項

5-1. 刻み幅と数値計算誤差の確認

4-3節で作成したプログラム (Ex4_3.java) の刻み幅 h を 0.01 としたプログラム (Ex5_1.java) を作成し、プログラム (Ex4_3.java) との誤差の比較を行いなさい。それぞれの誤差は、理論値と数値計算結果の差から求めることができる。

5-2. 台形公式による積分

台形公式を用い、4-3節に示した式の定積分の結果 S を求め表示するプログラム (Ex5_2.java) を作成しなさい。刻み幅 h は 0.1 とする。また、4-3節でのプログラム (Ex4_3.java) と誤差の比較を行いなさい。それぞれの誤差は、理論値と数値計算結果の差から求めることができる。

5-3. 運動方程式の解法とシミュレーション (オイラー法)

オイラー法を用いて物体の自由落下のシミュレーションを行うプログラム (Ex5_3.java) を作成しなさい。自由落下のシミュレーションには、2-3で述べたアルゴリズムを使用すること。このとき時刻の刻み幅 $h=0.01$ 、重力加速度 $g=9.80665$ とし、初速度 v_0 、初期高度 (位置) x_0 は任意の値をプログラム実行時に入力し、その結果を確認できるようにすること。シミュレーションでは、時刻 t 、高度 x および速度 v を求める。時刻 t は、繰り返し処理の中で $t=t+h$ のように計算すれば求められる。繰り返し処理の条件は、 $x>0$ 、すなわち物体が地面に接するまでとする。

また、初速度 $v_0=0(\text{m/s})$ 、初期高度 $x_0=100(\text{m})$ としたとき、物体の速さが $20(\text{m/s})$ に達した時刻 $t(\text{s})$ を求め、実験結果として記録しなさい。

計算結果である double 型変数について表示桁数を指定する場合 (小数点以下何桁にするかなど) は、`System.out.printf()` や `String.format()` を利用するとよい。

例 変数 x の値を小数点以下 3 桁で示す場合 (改行あり)。

```
System.out.printf("%.3f\n", x);  
System.out.println(String.format("%.3f", x));
```

5-4. シミュレーション結果の可視化

5-3節で作成したプログラムをもとに、計算結果をテキストファイルに出力するプログラム (Ex5_4.java) を作成しなさい。また、出力したテキストファイルを Excel によってグラフ化 (可視化) しなさい。初速度 $v_0=0(\text{m/s})$ 、初期高度 $x_0=100(\text{m})$ としたときのグラフを作成し、実験結果としてレポートに添付すること。

- ・プログラム内でファイルを出力する方法

以下のプログラム (FileTest.java) は、1～10 および 2～20 の整数値を実行ファイルのあるフォルダに data.txt として出力するものである。ファイル操作のために java.io.BufferedWriter、java.io.FileWriter、java.io.PrintWriter を、例外処理のために java.io.IOException をインポートしている。以下のプログラムを実行し、その動作を確認してから利用しなさい。

```

/*****
/*   FileTest.java   ファイル出力（テキストファイル）                               */
*****/

import java.io. BufferedWriter; //ファイルの書き出しに必要
import java.io. FileWriter; //ファイルの書き出しに必要
import java.io. PrintWriter; //ファイルの書き出しに必要
import java.io. IOException; //例外処理に必要

public class FileTest{
    public static void main(String[] args) {
        try {
            //FileWriter オブジェクトの生成。出力ファイル名は data.txt。
            FileWriter fw = new FileWriter("data.txt");
            //BufferedWriter オブジェクトの生成
            BufferedWriter bw = new BufferedWriter(fw);
            //PrintWriter オブジェクトの生成
            PrintWriter pw = new PrintWriter(bw);

            //ファイル書き出し
            for(int i=1; i<=10; i++){
                pw.println(i + "¥t" + (i * 2));
                //このようにも記述できる。
                //pw.printf("%d¥t%d¥n", i, (i * 2));
            }
            System.out.println("ファイルを出力しました。");
            pw.close(); //PrintWriter オブジェクトを閉じる。
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}

```

5-5. 運動方程式の解法とシミュレーション（ルンゲ・クッタ法）

微分方程式の数値計算的解法として、ルンゲ・クッタ法も有名な手法である。この手法を用いて、物体の自由落下のシミュレーションを行うプログラム（Ex5_5.java）を作成しなさい。また、初速度 $v_0=0(\text{m/s})$ 、初期高度 $x_0=100(\text{m})$ としたときのグラフを作成し、実験結果としてレポートに添付しなさい。（グラフ化できない場合は、計算結果の一部を示すこと）

6. 実験結果

作成（入力）したプログラムとともに、それぞれのプログラムの実行結果の数値を見やすいように実験報告書に記載しなさい。

7. 実験報告書に関して

実験報告書は、以下の項目について Word などで作成し提出しなさい。実験指導書にある内容を記述する場合は、自らの理解を補足し自らの言葉に直して短くまとめて記述をする。（ソースプログラムも実験報告書中に記載すること）

提出ファイルは pdf 形式とし、ファイル名は「提出者の学籍番号_prog3.pdf」とする。pdf ファイルに変換できない場合は、メールで相談すること。

（実験報告書に必要な項目）

- ・実験のタイトル、レポート作成者の学籍番号および氏名、実験日
- ・目的
- ・基本的事項
- ・実験課題と検討事項について、
 - ー入力・作成したすべてのプログラムのソースコード
 - ーすべてのプログラムの実行結果（数値、グラフ）
 - ー実験結果からわかること。考察
- ・まとめ（実験で理解したこと、求められたことをまとめる）
- ・参考文献（実験やレポート作成時に参考にした書籍、Web サイトなどについて明記する。Web サイトは公的機関の発表しているものを参考にしましょう）

参考文献の書き方：

著者名：「本のタイトル」，出版社，pp. 1－5（参考にしたページ。ネット上の情報は URL を示す）

引用についての注意：

図などを参考文献からそのまま引用するときは、参考文献に番号を付けて引用箇所を明記すること。