

学生のための上達 Java 練習問題解答

第 1 章 練習問題

1 (1)

```
//[1]面積，体積の計算(1)
package kihonproj;

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;

public class Sikakumenseki extends JApplet {
    //縦，横のデータの設定，面積の計算，結果の表示
    public void paint(Graphics g){
        int tate = 15;
        int yoko =200;
        int menseki = tate * yoko;
        g.drawString("縦 = 15(cm)，横 = 200(cm) の長方形の面積は", 50, 50);
        g.drawString(menseki +"(cm2) です", 50, 80);
    }
}
```

1 (2)

```
// [1]面積，体積の計算(2)
package kihonproj;

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;

public class Sankakumenseki extends JApplet {
    //変数 s (面積) の宣言
    double s;
    //底辺 a，高さ h の入力と面積 s の計算
    public void init(){
        String x, y;
        double a, h;
        x = JOptionPane.showInputDialog("三角形の底辺 a(cm)を入力してください");
        a = Double.parseDouble(x);
        y = JOptionPane.showInputDialog("三角形の高さ h(cm)を入力してください");
        h = Double.parseDouble(y);
        s = a*h/2;
    }
}
```

```

}
//面積 s の表示
public void paint(Graphics g){
    g.drawString("三角形の面積 s = " + s +"(cm2)", 50, 50);
}
}

```

1 (3)

// [1]面積，体積の計算(3)

```
package kihonproj;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.applet.*;
```

```
import javax.swing.*;
```

```
public class Enmenseki extends JApplet {
```

```
    //変数 c (円周), s (面積), v (体積) の宣言
```

```
    double c, s, v;
```

```
    //半径 r の入力と円周 c, 面積 s, 体積 v の計算
```

```
    public void init(){
```

```
        String x;
```

```
        double r;
```

```
        x = JOptionPane.showInputDialog("半径 r を入力してください");
```

```
        r = Double.parseDouble(x);
```

```
        c = 2*Math.PI*r;
```

```
        s = Math.PI*r*r;
```

```
        v = 4.0/3.0*Math.PI*r*r*r;
```

```
    }
```

```
    //円周 c, 面積 s, 体積 v の表示
```

```
    public void paint(Graphics g){
```

```
        g.drawString("円周 c = " + c, 50, 50);
```

```
        g.drawString("面積 s = " + s, 50, 80);
```

```
        g.drawString("体積 v = " + v, 50, 110);
```

```
    }
```

```
}
```

2 (1)

// [2]選択処理問題(1)

```
package kihonproj;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.applet.*;
```

```
import javax.swing.*;
```

```
public class Bigsmall extends JApplet {
```

```
    //変数 a,b の宣言
```

```
    int a,b;
```

```

//整数 a,b の入力
public void init(){
    String x,y;
    x = JOptionPane.showInputDialog("整数 a の値を入力してください");
    a = Integer.parseInt(x);
    y = JOptionPane.showInputDialog("整数 b の値を入力してください");
    b = Integer.parseInt(y);
    //メッセージペインへ a,b の値を表示
    System.out.println("a=" + a);
    System.out.println("b=" + b);
}
public void paint(Graphics g){
    //a,b の大きさの判定
    if(a>b){
        g.drawString("a =" + a + " ----- Big", 50, 50);
        g.drawString("b =" + b + " ----- Small", 50, 80);
    }
    else{
        g.drawString("a =" + a + " ----- Small", 50, 50);
        g.drawString("b =" + b + " ----- Big", 50, 80);
    }
}
}
}

```

2 (2)

// [2] 選択処理問題 (2)

```
package kihonproj;
```

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;
```

```
public class Kisetu extends JApplet {
```

```
    //変数 n の宣言
```

```
    int n;
```

```
    //1 から 12 の整数を入力
```

```
    public void init(){
```

```
        String x;
```

```
        x = JOptionPane.showInputDialog("1 から 12 の整数を入力してください\nn 季節を表示しま  
す");
```

```
        n = Integer.parseInt(x);
```

```
    }
```

```
    public void paint(Graphics g){
```

```
        //n の値を判定して場合分けする
```

```
        switch(n){
```

```
            case 3:
```

```
            case 4:
```

```
            case 5:g.drawString("春です!", 50, 50);
```

```
                break;
```

```

        case 6:
        case 7:
        case 8:g.drawString("夏です!", 50, 50);
            break;
        case 9:
        case 10:
        case 11:g.drawString("秋です!", 50, 50);
            break;
        case 12:
        case 1:
        case 2:g.drawString("冬です!", 50, 50);
            break;
        default:g.drawString("番号は1~12の整数を入力してください",50 ,50);
    }
}
}

```

2 (3)

// [2] 選択処理問題 (3)

```
package kihonproj;
```

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;
```

```
public class Seijin extends JApplet {
    //変数nの宣言
    int n;
    //年令nの入力
    public void init(){
        String x;
        x = JOptionPane.showInputDialog("あなたの年令を入力してください");
        n = Integer.parseInt(x);
    }
    public void paint(Graphics g){
        //年令が20才以上かの判定
        if(n>=20){
            g.drawString("喫煙が許されています.", 50, 50);
            g.drawString("健康に注意して喫煙しましょう.", 50, 80);
        }
        else{
            g.drawString("未成年です.", 50, 50);
            g.drawString("喫煙は許されていません.", 50, 80);
        }
    }
}

```

2 (4)

// [2] 選択処理問題 (4)

```

package kihonproj;

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;

public class Eitango extends JApplet {
    //変数 n の宣言
    int n;
    //n の値として 1 から 6 を入力
    public void init(){
        String x;
        x = JOptionPane.showInputDialog("1 から 6 を入力してください\nn その月の英単語を表示します");
        n = Integer.parseInt(x);
    }
    public void paint(Graphics g){
        //n の値を判定して場合分けする
        switch(n){
            case 1:g.drawString("January", 50, 50);
                break;
            case 2:g.drawString("February", 50, 50);
                break;
            case 3:g.drawString("March", 50, 50);
                break;
            case 4:g.drawString("April", 50, 50);
                break;
            case 5:g.drawString("May", 50, 50);
                break;
            case 6:g.drawString("June", 50, 50);
                break;
            default:g.drawString("1~6 の整数を入力してください", 50, 50);
        }
    }
}

```

2 (5)

// [2] 選択処理問題 (5)

```

package kihonproj;

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;

public class Nijihouteisiki extends JApplet {
    //変数 a,b,c (2 次方程式の係数) の宣言
    double a,b,c;
    //a,b,c の入力

```

```

public void init(){
    String x,y,z;
    x = JOptionPane.showInputDialog("2次方程式の係数 a を入力してください");
    a = Double.parseDouble(x);
    y = JOptionPane.showInputDialog("2次方程式の係数 b を入力してください");
    b = Double.parseDouble(y);
    z = JOptionPane.showInputDialog("2次方程式の係数 c を入力してください");
    c = Integer.parseInt(z);
}
public void paint(Graphics g){
    //変数 d (判別式), x1, x2 (異なる2つの実数解), x (重解) の宣言
    double d, x1, x2, x;
    //判別式の計算
    d = b*b-4.0*a*c;
    //d > 0 の判定
    if(d>0){
        //異なる2つの実数解の計算と表示
        x1 = (-b+Math.sqrt(d))/(2.0*a);
        x2 = (-b-Math.sqrt(d))/(2.0*a);
        g.drawString("相異なる2つの実数解", 50, 50);
        g.drawString("x1 = " + x1, 50, 80);
        g.drawString("x2 = " + x2, 50, 110);
    }
    else if(d==0){
        //重解の計算と表示
        x= -b/(2.0*a);
        g.drawString("重解", 50, 50);
        g.drawString("x = " + x, 50, 80);
    }
    else{
        //虚数解のときの表示
        g.drawString("虚数解です.", 50, 50);
    }
}
}
}

```

3 (1)

// [3] 図形問題(1)

```

package kihonproj;

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;

public class Dousinen extends JApplet {
    //変数 n の宣言
    int n;
    //円の個数 n を入力

```

```

public void init(){
    String x;
    x = JOptionPane.showInputDialog("描画する円の個数を入力してください");
    n = Integer.parseInt(x);
}
public void paint(Graphics g){
    int x =20,y=20;
    //描画色の設定
    g.setColor(Color.blue);
    //左上隅の座標(x,y),幅を10,20,30...,高さを10,20,30...とする円を表示
    for(int i=1; i<=n; i++){
        g.drawOval(x,y,10*i, 10*i);
    }
}
}
}

```

3 (2)

// [3]図形問題(2)

```
package kihonproj;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.applet.*;
```

```
import javax.swing.*;
```

```
public class Sankaku extends JApplet {
```

```
    //変数hの宣言
```

```
    int h;
```

```
    //h(行数)の入力
```

```
    public void init(){
```

```
        String x,y;
```

```
        x = JOptionPane.showInputDialog("三角形の高さ方向の行数hの値を入力してください");
```

```
        h = Integer.parseInt(x);
```

```
    }
```

```
    public void paint(Graphics g){
```

```
        int x, y = 0;
```

```
        //1行目に"*"を1個,2行目に"*"を2個,...,h行目に"*"をh個を横方向に表示
```

```
        for(int i=1; i<=h; i++){
```

```
            x = 25;
```

```
            y += 15;
```

```
            for(int j=1; j<=i; j++){
```

```
                g.drawString("*", x, y);
```

```
                x += 7;
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

4 (1)

```
//[4]反復処理問題(1)
package kihonproj;

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;

public class Sum1_100 extends JApplet {
    //for 文を用いて1 から 100 までの和を計算
    public void paint(Graphics g){
        g.drawString("1 から 100 までの総和を計算します", 50, 50);
        int sum = 0;
        for(int i=1; i<=100; i++){
            sum = sum + i;
        }
        g.drawString("総和 = " + sum, 50 ,80);
    }
}
```

4 (2)

```
// [4]反復処理問題(2)
package kihonproj;

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;

public class Sum1_n extends JApplet {
    //変数 n の宣言と変数 sum の初期値設定
    int n;
    int sum=0;
    //n の入力と 1 から n までの和の計算
    public void init(){
        String x;
        x = JOptionPane.showInputDialog("n を入力してください");
        n = Integer.parseInt(x);
        for (int i=1; i<=n; i++){
            sum = sum + i;
        }
    }
    //結果の表示
    public void paint(Graphics g){
        g.drawString("1 から " + n + "までの総和 = " + sum, 50, 50);
    }
}
```

4 (3) -1

```
// [4]反復処理問題(3)while 文による 1 から 100 までの和の計算
```



```

package kihonproj;

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;

public class Sum1_100_while extends JApplet {
    public void paint(Graphics g){
        //変数 sum と i の初期値設定
        int sum = 0;
        int i = 1;
        //1 から 100 までの和の計算
        while(i<=100){
            sum = sum + i;
            i++;
        }
        //結果の表示
        g.drawString("1 から 100 までの総和 = " + sum, 50, 50);
    }
}

```

4 (3) -2

// [4]反復処理問題(3)while 文による 1 から n までの和の計算

```

package kihonproj;

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;

public class Sum1_n_while extends JApplet {
    //変数 n の宣言と変数 sum の初期値設定
    int n;
    int sum=0;
    public void init(){
        //n の入力
        String x;
        x = JOptionPane.showInputDialog("n を入力してください");
        n = Integer.parseInt(x);
        //1 から n までの和の計算
        int i = 1;
        while(i<=n){
            sum = sum + i;
            i++;
        }
    }
    //結果の表示
    public void paint(Graphics g){
        g.drawString("1 から " + n + "までの総和 = " + sum, 50, 50);
    }
}

```

```
}  
}
```

4 (4)

```
// [4]反復処理問題(4)
```

```
package kihonproj;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.applet.*;
```

```
import javax.swing.*;
```

```
public class kaijou extends JApplet {
```

```
    //変数nの宣言と変数kaiの初期値設定
```

```
    int n;
```

```
    long kai = 1;
```

```
    //nの入力と階乗の計算
```

```
    public void init(){
```

```
        String x;
```

```
        x = JOptionPane.showInputDialog("n!を計算します。nの値を入力してください");
```

```
        n = Integer.parseInt(x);
```

```
        for (int i=1; i<=n; i++){
```

```
            kai = kai*i;
```

```
        }
```

```
    }
```

```
    //結果の表示
```

```
    public void paint(Graphics g){
```

```
        g.drawString(n + "! = " + kai, 50, 80);
```

```
    }
```

```
}
```

4 (5)

```
// [4]反復処理問題(5)
```

```
package kihonproj;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.applet.*;
```

```
import javax.swing.*;
```

```
public class E extends JApplet {
```

```
    //変数nの宣言と変数kai, eの初期値設定
```

```
    int n;
```

```
    double kai = 1;
```

```
    double e = 1;
```

```
    //nの入力と式の計算
```

```
    public void init(){
```

```
        String x;
```

```
        x = JOptionPane.showInputDialog("自然対数の底eを計算します。ne=1+1/1!+1/2!+...+1/n!  
n!のnの値を入力してください");
```

```

    n = Integer.parseInt(x);
    for (int i=1; i<=n; i++){
        kai = kai*i;    //1!, 2!, ..., の計算
        e = e +1/kai; //1+1/1!+1/2!+...+1/n!の計算
    }
}
//結果の表示
public void paint(Graphics g){
    g.drawString("e = " + e, 50, 80);
}
}

```

5 (1)

```

// [5]配列問題(1)
package kihonproj;

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;

public class Goukei_heikin_bunsan extends JApplet {
    public void paint(Graphics g){
        //配列 a[]の初期化
        double a[] = {10.0,20.0,30.0,40.0,50.0};
        double sum = 0, s=0;
        //和 sum の計算
        for(int i=0; i<a.length; i++){
            sum +=a[i];
        }
        //平均 ave の計算
        double ave = sum/a.length;
        //分散 dis の計算
        for( int i=0; i<a.length; i++){
            double w = a[i] - ave;
            s += w * w;
        }
        double dis = s/a.length;
        g.drawString("合計 = " + sum, 50, 50);
        g.drawString("平均 = " + ave, 50, 80);
        g.drawString("分散 = " + dis, 50, 110);
    }
}

```

5 (2)

```

// [5]配列問題(2)
package kihonproj;

import java.awt.*;

```

```

import java.awt.event.*;
import java.applet.*;
import javax.swing.*;

public class Hairetu_wa extends JApplet {
    public void paint(Graphics g){
        //配列 a[],b[]の初期化
        int a[] = {10, 20, 30, 40, 50};
        int b[] = {3, 5, 7, 8, 9};
        //配列 c[]の宣言とメモリ割り当て
        int c[] = new int[10];
        //a[i]+b[i]の計算
        for(int i=0; i<a.length; i++){
            c[i] = a[i]+b[i];
        }
        int x = 10;
        int y = 40;
        //表題 a[i] b[i] a[i]+b[i]の表示
        g.drawString("a[i]", x, 20);
        g.drawString("b[i]", x+60, 20);
        g.drawString("a[i] + b[i]", x+120, 20);
        //配列データの表示
        for(int i=0; i<a.length; i++){
            g.drawString(""+a[i], x, y);
            g.drawString(""+b[i], x+60, y);
            g.drawString(""+c[i], x+120, y);
            y += 30;
        }
    }
}

```

5 (3)

```

// [5]配列問題(3)
package kihonproj;

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;

public class Naiseki extends JApplet {
    public void paint(Graphics g){
        //配列 a[],b[]の初期化
        int a[] = {2, 4, 6, 8, 10};
        int b[] = {1, 3, 5, 7, 9};
        //配列 c[]の宣言とメモリ割り当て
        int c[] = new int[10];
        //s=a[1]*b[1]+a[2]*b[2]+...+a[n]*b[n]の計算
        int s = 0;
        for(int i=0; i<a.length; i++){

```

```

        s += a[i]*b[i];
    }
    int x = 10;
    int y = 40;
    //表題 a[i] b[i]の表示
    g.drawString("a[i]", x, 20);
    g.drawString("b[i]", x+60, 20);
    //配列データの表示
    for(int i=0; i<a.length; i++){
        g.drawString(""+a[i], x, y);
        g.drawString(""+b[i], x+60, y);
        y += 20;
    }
    //s の表示
    g.drawString("ベクトルの内積 s = " + s, x, y+30);
}
}

```

第 2 章 練習問題

1 (1)

```

package classproj;

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;

//Gakuseil クラスの定義
class Gakuseil{
    //メンバ変数の宣言 (学生の情報)
    String bango; //学生番号
    String simei; //氏名
    int eigo; //英語の点数
    int kokugo; //国語の点数
    int rika;//理科の点数
} //クラス Gakuseil の終わり

//アプレットの定義
public class ClassTest1 extends JApplet { //クラス名:TestClass1
    Gakuseil obj=new Gakuseil(); //Gakuseil クラスのオブジェクト obj を生成

    public void paint(Graphics g){ //Paint()メソッドの定義
        //メンバ変数にデータを設定
        obj.bango="999";
        obj.simei="富士山 太郎";
        obj.eigo=100;
        obj.kokugo=70;
    }
}

```

```
obj.rika=85;
```

```
//(1)情報の表示
```

```
//System.out.println("学生番号" + obj.bango); //メッセージペインに表示  
g.drawString("学生番号 " + obj.bango, 50, 50); //アプレットに表示  
g.drawString("氏名 " + obj.simei, 50, 80);  
g.drawString("英語の点数 " + obj.eigo, 50, 110);  
g.drawString("国語の点数 " + obj.kokugo, 50, 140);  
g.drawString("理科の点数 " + obj.rika, 50, 170);
```

```
//(2)メッセージペインへ表示
```

```
System.out.println("学生番号 " + obj.bango);  
System.out.println("氏名 " + obj.simei);  
System.out.println("英語の点数 " + obj.eigo);  
System.out.println("国語の点数 " + obj.kokugo);  
System.out.println("理科の点数 " + obj.rika);
```

```
}
```

```
} //クラス ClassTest1 の終わり
```

2

```
package classproj;
```

```
import java.awt.*;  
import java.awt.event.*;  
import java.applet.*;  
import javax.swing.*;
```

```
//Gakusei2 クラスの定義
```

```
class Gakusei2{  
    //メンバ変数の宣言(学生の情報)  
    String bango; //学生番号  
    String simei; //氏名  
    int eigo; //英語の点数  
    int kokugo; //国語の点数  
    int rika; //理科の点数
```

```
//3 科目の平均点を計算するメソッド heikin() の定義
```

```
double heikin(){//メソッド名 heikin, 戻り値型(Double)  
    int gokei=eigo + kokugo + rika; //3 科目の合計  
    double heikinten = gokei/3.0; //3 科目の平均  
    //System.out.println("平均点 "+heikinten);  
    return(heikinten); //平均点の戻り値(heikinten)
```

```
}
```

```
} //クラスの終わり
```

```
//アプレットの定義
```

```
public class ClassTest2 extends JApplet {  
    Gakusei2 obj = new Gakusei2(); //Gakusei2 クラスのオブジェクトを生成
```

```

public void paint(Graphics g){ //paint()メソッドの定義
    obj.bango = "999"; //メンバ変数にデータを与えます
    obj.simei = "富士山 太郎";
    obj.eigo = 100;
    obj.kokugo = 70;
    obj.rika = 85;

    //(1)情報の表示
    //System.out.println("学生番号" + obj.bango);
    g.drawString("学生番号 " + obj.bango, 50, 50); //アプレットに表示
    g.drawString("氏名 " + obj.simei, 50, 80);
    g.drawString("英語の点数 " + obj.eigo, 50, 110);
    g.drawString("国語の点数 " + obj.kokugo, 50, 140);
    g.drawString("理科の点数 " + obj.rika, 50, 170);

    //メソッド heikin()を呼び出して、平均点を表示
    g.drawString("平均点 " + obj.heikin(), 50, 200); //アプレットに表示

    //(2)メッセージペインへ平均点を表示
    System.out.println("平均点 " + obj.heikin());
}
} //アプレットの終わり

```

3

```

package classproj;

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;

//Gakusei3 クラスの定義
class Gakusei3 {
    //メンバ変数の宣言(学生の情報)
    String bango; //学生番号
    String simei; //氏名
    int eigo; //英語の点数
    int kokugo; //国語の点数
    int rika; //理科の点数

    //コンストラクタの定義
    Gakusei3(String b, String s, int e, int k, int r){
        //b,s,e,k,r は bango,simei,eigo,kougo,rika の意味
        bango = b;
        simei = s;
        eigo = e;
        kokugo = k;
        rika = r;
    }
}

```

```

//3 科目の平均点を計算するメソッド heikin()
double heikin(){
    int gokei = eigo + kokugo + rika;
    double heikinten = gokei/3.0;
    //System.out.println("");
    return(heikinten);
}
} //クラスの終わり

//アプレット
public class ClassTest3 extends JApplet {
    //コンストラクタ Gakusei3() に初期値 "999", "富士山 太郎", 100, 70, 85 を与えます
    Gakusei3 obj=new Gakusei3("999", "富士山 太郎", 100, 70, 85);
    public void paint(Graphics g){
        /*
        obj.bango="999"; //メンバ変数にデータを設定
        obj.simei="富士山 太郎";
        obj.eigo=100;
        obj.kokugo=70;
        //System.out.println("学生番号" + obj.bango);
        */
        //情報の表示
        g.drawString("学生番号 " + obj.bango, 50, 50); //アプレットに表示
        g.drawString("氏名 " + obj.simei, 50, 70);
        g.drawString("英語の点数 " + obj.eigo, 50, 90);
        g.drawString("国語の点数 " + obj.kokugo, 50, 110);
        g.drawString("理科の点数 " + obj.rika, 50, 130);

        //メソッド heikin() を呼び出して, 表示
        g.drawString("平均点 " + obj.heikin(), 50, 160); //アプレットに表示
    }
} //アプレットの終わり

```

4

```

package classproj;

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;

//Gakusei4 クラスの定義
//Gakusei4 という 1 つのクラスの中に同じ名前のコンストラクタを 3 つ定義してあります
class Gakusei4{
    //メンバ変数の宣言 ( 学生の情報 )
    String bango; //学生番号
    String simei; //氏名
    int eigo; //英語の点数

```



```

int kokugo; //国語の点数
int rika; //理科の点数
int hoken; //保健の点数
int goukei; //合計

//コンストラクタ1の定義
Gakusei4(String b, String s, int e, int k){ //引数b,s,e,kの4つ
    bango = b;
    simei = s;
    eigo = e;
    kokugo = k;
    goukei = eigo + kokugo; //英語,国語の合計点
}

//コンストラクタ2の定義
Gakusei4(String b, String s, int e, int k, int r){ //引数b,s,e,k,rの5つ
    bango = b;
    simei = s;
    eigo = e;
    kokugo = k;
    rika = r;
    goukei = eigo + kokugo + rika; //英語,国語,理科の合計点
}

//コンストラクタ3の定義
Gakusei4(String b, String s, int e, int k, int r, int h){
    //引数b,s,e,k,r,hの6つ
    bango = b;
    simei = s;
    eigo = e;
    kokugo = k;
    rika = r;
    hoken = h;
    goukei = eigo + kokugo + rika + hoken; //英語,国語,理科,保健の合計点
}
} //クラスGakusei4の終わり

//アプレットの定義
public class ClassTest4 extends JApplet {
    //オブジェクトobj1,obj2の生成,コンストラクタの初期化
    Gakusei4 obj1 = new Gakusei4("999"," 富士山 太郎",100, 70);
    Gakusei4 obj2 = new Gakusei4("999"," 富士山 太郎",100, 70, 85);
    Gakusei4 obj3 = new Gakusei4("999"," 富士山 太郎",100, 70, 85, 60);

    //学生番号,氏名,科目の点数,合計点の表示
    public void paint(Graphics g){
        //学生番号,氏名の表示
        g.drawString(obj1.bango + obj1.simei, 50, 50);

        //英語,国語の点数の表示と合計

```

```

g.drawString("英語 = " + obj1.eigo +
             " 国語 = " + obj1.kokugo, 50, 80);
g.drawString("2 科目(英語, 国語)の合計点 = " + obj1.goukei, 50,100);

//英語, 国語, 理科の点数の表示と合計
g.drawString("英語 = " + obj2.eigo
             + " 国語 = " + obj2.kokugo
             + " 理科 = " + obj2.rika, 50, 130);
g.drawString("3 科目(英語, 国語, 理科)の合計点 = " + obj2.goukei, 50,150);

//英語, 国語, 理科, 保健の点数の表示と合計
g.drawString("英語 = " + obj3.eigo
             + " 国語 = " + obj3.kokugo
             + " 理科 = " + obj3.rika
             + " 保健 = " + obj3.hoken,50,180);
g.drawString("4 科目(英語, 国語, 理科, 保健)の合計点 = " + obj3.goukei, 50,200);
}
} //アプレットの終わり

```

第3章 練習問題

1 解答例

- (1) JTextField()
- (2) JLabel()
- (3) setText(" 500")
- (4) setText(" 800")
- (5) getContentPane()
- (6) getContentPane()

2

JBUILDER によって自動生成されたコード一覧

```

1: package guiproj;
2:
3: import java.awt.*;
4: import java.awt.event.*;
5: import java.applet.*;
6: import javax.swing.*;
7:
8: public class GUITest2_1 extends JApplet {
9:     JPanel jPanel1 = new JPanel();
10:    JLabel jLabel1 = new JLabel();
11:    JTextField jTextField1 = new JTextField();
12:    JLabel jLabel2 = new JLabel();
13:    JTextField jTextField2 = new JTextField();
14:    JLabel jLabel3 = new JLabel();
15:    JTextField jTextField3 = new JTextField();
16:

```

```

17: public GUITest2_1() {
18:     try {
19:         jbInit();
20:     }
21:     catch(Exception e) {
22:         e.printStackTrace();
23:     }
24: }
25: private void jbInit() throws Exception {
26:     jLabel1.setText("学生番号");
27:     jTextField1.setText(" 120");
28:     jLabel2.setText(" 氏名");
29:     jTextField2.setText("鈴木 一太郎");
30:     jLabel3.setText(" 英語");
31:     jTextField3.setText(" 80");
32:     this.getContentPane().add(jPanel1, BorderLayout.CENTER);
33:     jPanel1.add(jLabel1, null);
34:     jPanel1.add(jTextField1, null);
35:     jPanel1.add(jLabel2, null);
36:     jPanel1.add(jTextField2, null);
37:     jPanel1.add(jLabel3, null);
38:     jPanel1.add(jTextField3, null);
39: }
40: }

```

解答例

8~40行：クラス GUITest2_1 の定義

9~15行：new 演算子によってオブジェクトを生成

17~24行：コンストラクタの定義，jbInit()メソッドを呼び出す

25~39行：jbInit()メソッドの定義

26,27,28,29,30,31行：コンポーネントの初期化

32行：getContentPane()メソッドで取得したコンテナオブジェクト（描画領域）内にadd()メソッドを用いて jPanel1 を配置

33~38行：jPanel1 にadd()メソッドを用いて jLabel1 , jTextField1 , jLabel2 , jTextField2 , jLabel3 , jTextField3 のオブジェクトを配置

第4章 練習問題

1 解答例

インスペクタ画面の中で次のプロパティを設定します。

jLabel1 の Background を Yellow , Foreground を Blue , opaque を true

jLabel2 の Background を Cyan , Foreground を Red , opaque を true

```

void jButton1_actionPerformed(ActionEvent e) {
    jLabel1.setText("ようこそJavaの世界へ!");
    jLabel2.setText("Javaはオブジェクト指向言語です!");
}

```

2 解答例

```
void jButton1_actionPerformed(ActionEvent e) {
    String s = jTextField1.getText(); //jTextField1 に入力された文字列を変数 s に代入
    for(int i=1; i<=10; i++){
        jTextArea1.append(s + "\n"); //変数 s の内容を jTextArea1 に追加, 表示
    }
}
```

3 解答例

//ボタングループにチェックボックスを追加

```
buttonGroup1.add(jCheckBox1);
buttonGroup1.add(jCheckBox2);
buttonGroup1.add(jCheckBox3);
buttonGroup1.add(jCheckBox4);
```

```
void jCheckBox1_itemStateChanged(ItemEvent e) {
    flag = 1;
}
```

```
void jCheckBox2_itemStateChanged(ItemEvent e) {
    flag = 2;
}
```

```
void jCheckBox3_actionPerformed(ActionEvent e) {
    flag = 3;
}
```

```
void jCheckBox4_actionPerformed(ActionEvent e) {
    flag = 4;
}
```

```
int flag;
```

```
JCheckBox jCheckBox4 = new JCheckBox();
```

```
void jButton1_actionPerformed(ActionEvent e) {
    switch(flag){
        case 1:jLabel1.setText("Spring");
            break;

        case 2:jLabel1.setText("Summer");
            break;

        case 3:jLabel1.setText("Autumn");
            break;

        case 4:jLabel1.setText("Winter");
            break;
    }
}
```

4 解答例

- (1) 設計画面で jPanel1 の幅と高さを変えてみてください。
- (2) jPanel1 の幅と高さの値を調べて、小さい方の値の 1/2 をスクロールボックスの maximum プロパティの値にします。

```
jPanel1.setBounds(new Rectangle(x, y, 幅, 高さ))
```

- (3) unitIncrement プロパティを 2 に変更します。
- (4) value プロパティの値を maximum プロパティの値の 1/2 にします。

5 解答例

jScrollBar1 のプロパティの設定

```
maximum = 20
orientation = HORIZONTAL
unitIncrement = 2
visibleAmount = 0
```

```
void jScrollBar1_adjustmentValueChanged(AdjustmentEvent e) {
    //jPanel1 のサイズオブジェクト d を取得
    Dimension d = jPanel1.getSize();

    //jPanel1 のグラフィックオブジェクト g を取得
    Graphics g = jPanel1.getGraphics();

    //スクロールボックスの現在値 sv を取得
    int sv = jScrollBar1.getValue();
    jTextField1.setText(Integer.toString(sv));

    //パネル全体を緑で描画
    g.setColor(Color.blue);
    g.fillRect(0, 0, d.width, d.height);

    //パネルの中央に赤い円を描画
    //g.setColor(Color.red);
    //g.fillOval(d.width/2-sv, d.height/2-sv, 2*sv, 2*sv);
    int w = (int)d.width*sv/20; //描画する四角形の幅の計算
    int h = (int)d.height*sv/20; //描画する四角形の高さの計算
    g.setColor(Color.yellow); //色の指定
    g.fillRect(0, 0, w, h); //四角形の描画
}
```

6 解答例

```
private void jbInit() throws Exception {
    .
    .
    //コンボボックスの項目リストの追加
    jComboBox1.addItem("北海道 "); //コンボボックス 1 に表示する項目リスト
    jComboBox1.addItem("青森県 ");
```

```

jComboBox1.addItem("宮城県 ");
jComboBox1.addItem("東京都 ");
jComboBox1.addItem("宮城県 ");
jComboBox2.addItem("青梅市 "); // コンボボックス 2 に表示する項目リスト
jComboBox2.addItem("日南市 ");
jComboBox2.addItem("札幌市 ");
jComboBox2.addItem("仙台市 ");
jComboBox2.addItem("十和田市 ");

jComboBox3.addItem("ミヤギノハギ"); // コンボボックス 3 に表示する項目リスト
jComboBox3.addItem("ハマナス");
jComboBox3.addItem("ハマユウ");
jComboBox3.addItem("リンゴノハナ");
jComboBox3.addItem("ソメイヨシノ");
}

```

第 5 章 練習問題

1 解答例

```

// 集計用変数
int n1 = 0;
int n2 = 0;
int n3 = 0;
void jButton1_actionPerformed(ActionEvent e) {
    String outlist="";
    outlist += " 野球"+" \t"+n1+"\n"+" サッカー"+" \t"+n2+"\n"+" テニス"+
        "\t"+n3+"\n";
    jTextArea1.setText(outlist);

    // チェックマークを消す
    jCheckBox1.setSelected(false);
    jCheckBox2.setSelected(false);
    jCheckBox3.setSelected(false);
}

```

```

void jButton2_actionPerformed(ActionEvent e) {
    // テキストエリアをクリア
    jTextArea1.setText("");
    n1=0;
    n2=0;
    n3=0;
}

```

```

void jCheckBox1_actionPerformed(ActionEvent e) {
    // チェックされたら n1 に 1 を加える
    n1=n1+1;
}

```

```
void jCheckBox2_actionPerformed(ActionEvent e) {
    //チェックされたら n2 に 1 を加える
    n2=n2+1;
}
```

```
void jCheckBox3_actionPerformed(ActionEvent e) {
    //チェックされたら n3 に 1 を加える
    n3=n3+1;
}
```

2 解答例

```
void jButton1_actionPerformed(ActionEvent e) {
    //半径の入力
    String s = jTextField1.getText();
    double r = Double.parseDouble(s);

    //円周の計算, 表示
    double l = 2.0*Math.PI*r;
    jTextField2.setText(Double.toString(l));

    //面積の計算
    double m = Math.PI*r*r;
    jTextField3.setText(Double.toString(m));
}
```

3 解答例

```
void jButton1_actionPerformed(ActionEvent e) {
    double hi = Double.parseDouble(jTextField1.getText());
    double we = Double.parseDouble(jTextField2.getText());

    double sta = (hi-100)*0.9;
    double ws = we/sta;
    if(ws>1.1)
        jLabel4.setText("太りすぎです!");
    else if(ws>=0.9 && ws<=1.1)
        jLabel4.setText("標準です!");
    else
        jLabel4.setText("やせています!");
}

void jButton2_actionPerformed(ActionEvent e) {
    jTextField1.setText("");
    jTextField2.setText("");
    jLabel4.setText("");
}

void jButton3_actionPerformed(ActionEvent e) {
    System.exit(0);
}
```

```
}
```

4 解答例

```
void jButton1_actionPerformed(ActionEvent e) {  
    for(int i=1; i<=9; i++){  
        for(int j=1; j<=9; j++){  
            int x = i*j;  
            jTextArea1.append(x+"\t");  
        }  
        jTextArea1.append("\n");  
    }  
}
```

5 解答例

```
//略語リスト  
jComboBox1.addItem("DVD");  
jComboBox1.addItem("CPU");  
jComboBox1.addItem("bit");  
jComboBox1.addItem("byte");  
jComboBox1.addItem("IC");  
  
void jComboBox1_actionPerformed(ActionEvent e) {  
    int i = jComboBox1.getSelectedIndex();  
    String outlist0 = "【digital videodisk】デジタル-ビデオディスク . CD と同じ直径 12cm  
の光ディスクに映像・音声情報を記録する . ";  
    String outlist1 = "【 CPU 】[central processing unit] 中央処理装置 . コンピュータ  
ーの中核となる装置で , 命令の解釈と実行の制御を行う . ";  
    String outlist2 = "【bit】【binary digit の略】(1)2 進法で基礎とする数字の 0 また  
は 1 . (2) 情報量を示す単位 . ";  
    String outlist3 = "【byte】情報量を示す単位 . 普通 , 1 バイトは 8 ビットである . ";  
    String outlist4 = "【 IC 】[integrated circuit] 集積回路のことで数ミリメートル四方  
のシリコンまたはガリウムヒ素などの半導体を加工して , 多数の回路素子を組み込み電子回路として機能  
するようにしたもので多くの電気製品に用いられている . ";  
  
    if(i == 0)  
        jTextArea1.setText(outlist0);  
    else if(i ==1)  
        jTextArea1.setText(outlist1);  
    else if(i ==2)  
        jTextArea1.setText(outlist2);  
    else if(i ==3)  
        jTextArea1.setText(outlist3);  
    else  
        jTextArea1.setText(outlist4);  
}
```

6


```

void jButton2_actionPerformed(ActionEvent e) {
    DecimalFormat precision = new DecimalFormat("0.00"); //平均値の表示形式を指定
    //表題
    JTextArea1.append("==== 縦横の計算 ==== " + "\n");
    JTextArea1.append("番号"+"\\t"+"国語"+"\\t"+"数学"+"\\t"+"英語"+"\\t"+
        "個人平均"+"\\n");

    //データの表示と横の計算
    for(int i=0; i<ten.length; i++){
        JTextArea1.append(Integer.toString(i+1)+"\\t"); //番号の表示
        int kei=0;
        for(int j=0; j<ten[i].length; j++){
            kei += ten[i][j]; //3 科目の合計を計算
            JTextArea1.append(ten[i][j] + "\\t"); //3 科目の点数を表示
        }

        //平均値の表示
        JTextArea1.append(precision.format(kei/3.0)+ "\\n"); //平均値を表示
    }

    //縦の計算と表示
    //int i=0;
    for(int j=0; j<ten[j].length; j++){
        double kei=0;
        for( int i=0; i<ten.length; i++){
            kei += ten[i][j]; //科目の合計
        }
        //平均値の表示
        JTextArea1.append("\\t" + precision.format(kei/ten.length)); //平均値を表示
    }
}

```

7

- (1) 43 行: blue を red に書きかえる .
- (2) 40 行の文を次のように書きかえる .

$y = -x * x$ $y = -x * x + 5$ $y = x + 1$ $y = x * x * x$ $y = (x + 1) * (x - 2)$

8 解答例

```

void jButton1_actionPerformed(ActionEvent e) {
    Dimension d = jPanel1.getSize();
    Graphics g = jPanel1.getGraphics();

    double x0 = d.width/2; //jPanel1 の幅の中央
    double y0 = d.height/2; //jPanel1 の高さの中央
    double r = d.height/2; //jPanel1 の高さの中央

    g.setColor(Color.white);
    g.fillRect(0, 0, d.width, d.height); //jPanel1 を白で塗りつぶす
}

```

```

g.setColor(Color.black); //軸の描画色
g.drawLine(0, (int)y0, (int)d.width, (int)y0); //x軸
g.drawLine((int)x0, 0, (int)x0, (int)d.height); //y軸

double x1 = r*Math.sin(m*0.0); //描画開始点の計算
double y1 = r*Math.sin(n*0.0); //描画開始点の計算
double xg1 = x0 + x1; //描画座標に変換
double yg1 = y0 - y1; //描画座標に変換
for(double s = 0; s<=360.0; s+=0.05){//0度~360まで0.05きざみで繰り返す
    double rad = Math.toRadians(s); //角度sをラジアンに変換
    double x2 = r*Math.sin(m*rad); //式の計算
    double y2 = r*Math.sin(n*rad); //式の計算
    double xg2 = x0+(int)x2; //描画座標に変換
    double yg2 = y0-(int)y2; //描画座標に変換
    g.setColor(Color.red); //描画色の指定
    g.drawLine((int)xg1, (int)yg1, (int)xg2, (int)yg2); //リサージュ図形の描画
    xg1 = xg2; //xg2をxg1へ代入
    yg1 = yg2; //yg2をyg1へ代入
}
}
int m, n;
void jButton1_actionPerformed(ActionEvent e) {
    m = 1;
}

void jButton2_actionPerformed(ActionEvent e) {
    m = 2;
}

void jButton3_actionPerformed(ActionEvent e) {
    m = 3;
}

void jButton4_actionPerformed(ActionEvent e) {
    m = 4;
}

void jButton5_actionPerformed(ActionEvent e) {
    m = 5;
}

void jButton6_actionPerformed(ActionEvent e) {
    n = 1;
}

void jButton7_actionPerformed(ActionEvent e) {
    n = 2;
}

void jButton8_actionPerformed(ActionEvent e) {

```

```
    n = 3;
}

void jButton9_actionPerformed(ActionEvent e) {
    n = 4;
}

void jButton10_actionPerformed(ActionEvent e) {
    n = 5;
}

void jButton3_actionPerformed(ActionEvent e) {
    System.exit(0);
}
}
```