

例題でわかるC++Builder 問・練習問題 解答

解答は原則として、イベントハンドラだけを載せてある。網掛けの部分は読者が自分で入力する部分である。

第I部 C++Builderの基礎

第2章 問2.1

【プロパティの設定】

Label1 の Alignment	プロパティ	taLeftJustify に設定
Label2 の Alignment	プロパティ	taCenter に設定
Label3 の Alignment	プロパティ	aRightJustify に設定

【イベントハンドラ】

```
//-----  
void__fastcallTForm1::Button1Click(TObject*Sender)  
{  
    Label1->Caption="WelcometoC++Builder";  
    Label2->Caption="WelcometoC++Builder";  
    Label3->Caption="WelcometoC++Builder";  
}  
//-----  
void__fastcallTForm1::FormCreate(TObject*Sender)  
{  
    Label1->Caption="";  
    Label2->Caption="";  
    Label3->Caption="";  
}  
//-----
```

第2章 練習問題 1

【イベントハンドラ】

```
//-----  
void__fastcallTForm1::FormCreate(TObject*Sender)  
{  
    Label1->Caption="";  
    Label2->Caption="";  
    Label3->Caption="";  
}  
//-----  
void__fastcallTForm1::Button1Click(TObject *Sender)  
{  
    Label1->Caption="WelcometoC++Builder";  
    Label2->Caption="WelcometoC++Builder";  
    Label3->Caption="WelcometoC++Builder";  
}  
//-----
```

```
void__fastcallTForm1::Button2Click(TObject *Sender)
```

```
{
```

```
Label1->Caption="";
```

```
Label2->Caption="";
```

```
Label3->Caption="";
```

```
}
```

```
//-----
```

```
void__fastcallTForm1::Button3Click(TObject *Sender)
```

```
{
```

```
Close();
```

```
}
```

```
//-----
```

第2章 練習問題 2

【イベントハンドラ】

```
//-----
```

```
void__fastcallTForm1::FormCreate(TObject*Sender)
```

```
{
```

```
Label1->Caption="";
```

```
Label2->Caption="";
```

```
Label3->Caption="";
```

```
}
```

```
//-----
```

```
void__fastcallTForm1::Button1Click(TObject *Sender)
```

```
{
```

```
Label1->Caption="富士 一郎";
```

```
}
```

```
//-----
```

```
void__fastcallTForm1::Button2Click(TObject *Sender)
```

```
{
```

```
Label2->Caption="静岡県富士市山田町1 - 1 - 1";
```

```
}
```

```
//-----
```

```
void__fastcallTForm1::Button3Click(TObject *Sender)
```

```
{
```

```
Label3->Caption="0123-45-6789";
```

```
}
```

```
//-----
```

第3章 問3.1

【イベントハンドラ】

```
//-----
```

```
void__fastcallTForm1::FormCreate(TObject*Sender)
```

```
{
```

```
Edit1->Text="";
```

```
Edit2->Text="";
```

```
Edit3->Text="";
```

```
}
```

```
//-----
```

```

void__fastcallTForm1::Button1Click(TObject *Sender)
{
    Edit3->Text=FormatFloat("#.##", (StrToFloat(Edit1->Text)
        +StrToFloat(Edit2->Text)));
}
//-----
void__fastcallTForm1::Button3Click(TObject *Sender)
{
    Edit1->Text="";
    Edit2->Text="";
    Edit3->Text="";
}
//-----
void__fastcallTForm1::Button4Click(TObject *Sender)
{
    Close();
}
//-----
void__fastcallTForm1::Button2Click(TObject *Sender)
{
    Edit3->Text=FormatFloat("#.##", (StrToFloat(Edit1->Text)
        -StrToFloat(Edit2->Text)));
}
//-----

```

第3章 練習問題 1

【イベントハンドラ】

```

//-----
void__fastcallTForm1::FormCreate(TObject*Sender)
{
    Edit1->Text="0";
    Edit2->Text="0";
}
//-----

void__fastcallTForm1::BitBtn2Click(TObject *Sender)
{
    doubleippan_ryokin,gakusei_ryokin,gokei;
    ippan_ryokin = StrToFloat(Edit1->Text)*StrToFloat(Panel5->Caption);
    gakusei_ryokin=StrToFloat(Edit2->Text)*StrToFloat(Panel6->Caption);
    gokei=ippan_ryokin+ gakusei_ryokin;
    Panel3->Caption=FormatFloat("###,###",gokei);
}
//-----
void__fastcallTForm1::BitBtn3Click(TObject *Sender)
{
    Close();
}
//-----

```

```
void__fastcallTForm1::BitBtn1Click(TObject *Sender)
```

```
{  
    Edit1->Text="0";  
    Edit2->Text="0";  
    Panel3->Caption="0";  
}
```

```
}
```

```
//-----
```

第4章 練習問題 1

【イベントハンドラ】

```
//-----
```

```
void__fastcallTForm1::FormCreate(TObject*Sender)
```

```
{  
    Memo1->Text="0";  
    Initial_Value();  
}
```

```
}
```

```
//-----
```

```
void__fastcallTForm1::SpeedButton1Click(TObject *Sender)
```

```
{  
    TSpeedButton*btn;          //Sender を TSpeedButton として  
    btn= (TSpeedButton*)Sender; //扱えるように定義する  
  
    ButtonCheck(btn);          //計算前に記号ボタンを押したかどうかをチェックする関数  
  
    if(shori_flag2==1){        //不適当なボタンが押された場合の処理  
        shori_flag2= 0;  
        shori_flag1= 0;  
    }  
    else{  
        ButtonShori(btn);      //各ボタンに対する処理関数  
    }  
}
```

```
}
```

```
//-----
```

【ヘッダファイル】

```
//-----
```

```
#ifndef mon4_1Unit1_2H
```

```
#define mon4_1Unit1_2H
```

```
//-----
```

```
#include <Classes.hpp>
```

```
#include <Controls.hpp>
```

```
#include <StdCtrls.hpp>
```

```
#include <Forms.hpp>
```

```
#include <Buttons.hpp>
```

```
#include <ExtCtrls.hpp>
```

```
//-----
```

```
classTForm1:publicTForm
```

```
{  
    __published: //IDE 管理のコンポーネント  
    TPanel*Panel1;
```

```

TPanel*Panel2;
TMemo*Mem1;
TSpeedButton *SpeedButton1;
TSpeedButton *SpeedButton2;
TSpeedButton *SpeedButton3;
TSpeedButton *SpeedButton4;
TSpeedButton *SpeedButton5;
TSpeedButton *SpeedButton6;
TSpeedButton *SpeedButton7;
TSpeedButton *SpeedButton8;
TSpeedButton *SpeedButton9;
TSpeedButton *SpeedButton10;
TSpeedButton *SpeedButton11;
TSpeedButton *SpeedButton12;
TSpeedButton *SpeedButton13;
TSpeedButton *SpeedButton14;
TSpeedButton *SpeedButton15;
TSpeedButton *SpeedButton16;
TSpeedButton *SpeedButton17;
void__fastcallFormCreate(TObject *Sender);
void__fastcallSpeedButton1Click(TObject*Sender);
private:      // ユーザ宣言
floatkei;    //合計値
floatw;      //作業用
floatsign;   //符号の保存
intshori_flag1;//計算開始のチェック(初期値:0, 計算途中:1)
intshori_flag2;//不適当なボタンを押したかどうかのチェック(初期値:0, 押した:1)
AnsiStrings; //作業用
TSpeedButton*btn; //Sender を TSpeedButton として扱えるように定義する
public:      // ユーザ宣言
__fastcallTForm1(TComponent*Owner);
//-----
// 初期値設定関数
void__fastcallInitial_Value(void){
    shori_flag1=0;
    shori_flag2=0;
    sign=+1;
}
//-----
// ボタンチェック処理関数 計算開始時, 不適当なボタンを押した場合の処理
voidButtonCheck(TSpeedButton*btn)
{
    if(shori_flag1==0){ //計算開始の場合の処理
        kei=0;
        Mem1->Text=" "; //入力ボックスの初期化
        shori_flag1= 1; //shori_flag1 を 1 にセット
        switch(btn->Tag){ //数値の入力前に記号ボタンを押した場合の処理
            case11:
            case12:
            case13:

```

```

    case14:
    case16:
    case17:
        ShowMessage("不適當なボタンが押された!");
        shori_flag2=1;
        break;
    }
}
}
//-----
// ボタンの処理関数 各ボタンの処理
//
voidButtonShori(TSpeedButton*btn)
{
    switch(btn->Tag){
        case 1: Mem01->Text=Mem01->Text+"1";break; // 1 の処理
        case 2: Mem01->Text=Mem01->Text+"2";break; // 2 の処理
        case 3: Mem01->Text=Mem01->Text+"3";break; // 3 の処理
        case 4: Mem01->Text=Mem01->Text+"4";break; // 4 の処理
        case 5: Mem01->Text=Mem01->Text+"5";break; // 5 の処理
        case 6: Mem01->Text=Mem01->Text+"6";break; // 6 の処理
        case 7: Mem01->Text=Mem01->Text+"7";break; // 7 の処理
        case 8: Mem01->Text=Mem01->Text+"8";break; // 8 の処理
        case 9: Mem01->Text=Mem01->Text+"9";break; // 9 の処理
        case10: Mem01->Text=Mem01->Text+"0";break; // 0 の処理
        case11: kei=kei+sign*(StrToFloat(Mem01->Text)); // +の処理
                Mem01->Text=" ";
                sign = +1;
                break;
        case12: kei=kei+sign*(StrToFloat(Mem01->Text)); // -の処理
                Mem01->Text=" ";
                sign = -1;
                break;
        case13: // *の処理
        case14: // /の処理
        case15: Mem01->Text="0"; //AC(AllClear)の処理
                shori_flag1=0;
                break;
        case16: if(Mem01->Text!="") // =の処理
                w=kei+sign*(StrToFloat(Mem01->Text));
                else
                w=sign*kei;
                Mem01->Text=FormatFloat("#0.####0",w); //小数第5位まで表示
                shori_flag1=0;
                sign = +1;
                break;
        case17: Mem01->Text=Mem01->Text+".";break; // .の処理
    }
}
};

```

第4章 練習問題 2

関数化

【イベントハンドラ】

```
//-----  
// 加減乗除算 === 関数化  
//-----  
void__fastcallTForm1::FormCreate(TObject*Sender)  
{  
    Memo1->Text="0";  
}  
//-----  
floatkei=0;        //合計値  
floatjyoujo=1;    //乗除計算領域  
floatw;           //作業用  
intshori_flag1=0;//計算開始のチェック(初期値:0, 計算途中:1)  
intshori_flag2=0;//不適当なボタンを押したかどうかのチェック(初期値:0, 押した:1)  
AnsiStrings;     //作業用  
AnsiStringwkazu; //退避用  
charop;          //演算記号保存用  
//-----  
//  
// ButtonCheck : ボタンチェック関数  
voidButtonCheck(TSpeedButton*btn)  
{  
    if(shori_flag1==0){        //計算開始の場合の処理  
        kei=0;  
        Form1->Memo1->Text="";        //入力ボックスの初期化  
        wkazu=" ";  
        op="+";  
        jyoujo=1;  
        shori_flag1=1;            //shori_flag1 を1 にセット  
        switch(btn->Tag){        //数値の入力前に記号ボタンを押した場合の処理  
            case11:                // +  
            case12:                // -  
            case13:                // *  
            case14:                // /  
            case16:                // =  
            case17:                // .  
                ShowMessage("不適当なボタンが押された!");  
                shori_flag2=1;  
                break;  
        }  
    }  
}  
//-----  
// ボタン処理関数  
voidButtonShori(TSpeedButton*btn)  
{
```

```

if(shori_flag2==1){ //不適当なボタンが押された場合の処理
    shori_flag2= 0;
    shori_flag1= 0;
}
else{
    switch(btn->Tag){
        case 1:Form1->Memol->Text=Form1->Memol->Text+"1";break; //1の処理
        case 2:Form1->Memol->Text=Form1->Memol->Text+"2";break; //2の処理
        case 3:Form1->Memol->Text=Form1->Memol->Text+"3";break; //3の処理
        case 4:Form1->Memol->Text=Form1->Memol->Text+"4";break; //4の処理
        case 5:Form1->Memol->Text=Form1->Memol->Text+"5";break; //5の処理
        case 6:Form1->Memol->Text=Form1->Memol->Text+"6";break; //6の処理
        case 7:Form1->Memol->Text=Form1->Memol->Text+"7";break; //7の処理
        case 8:Form1->Memol->Text=Form1->Memol->Text+"8";break; //8の処理
        case 9:Form1->Memol->Text=Form1->Memol->Text+"9";break; //9の処理
        case10:Form1->Memol->Text=Form1->Memol->Text + "0";break; //0の処理
        case11:switch(op){ // +の処理
            case '*':jyoujo=jyoujo*StrToFloat(Form1->Memol->Text);
                kei=kei+jyoujo;
                break;
            case '/':jyoujo=jyoujo/StrToFloat(Form1->Memol->Text);
                kei=kei+jyoujo;
                break;
            case '+':kei=kei+StrToFloat(Form1->Memol->Text);
                break;
            case '-':kei=kei-StrToFloat(Form1->Memol->Text);
                break;
        }
        Form1->Memol->Text=" ";
        op='+';
        break;
    case12:switch(op){ // - の処理
        case '*':jyoujo=jyoujo*StrToFloat(Form1->Memol->Text);
            kei=kei+jyoujo;
            break;
        case '/':jyoujo=jyoujo/StrToFloat(Form1->Memol->Text);
            kei=kei+jyoujo;
            break;
        case '+':kei=kei+StrToFloat(Form1->Memol->Text);
            break;
        case '-':kei=kei-StrToFloat(Form1->Memol->Text);
            break;
    }
    Form1->Memol->Text=" ";
    op='-';
    break;
    case13:switch(op){ // *の処理
        case '*':jyoujo=jyoujo*StrToFloat(Form1->Memol->Text);
            break;
        case '/':jyoujo=jyoujo/StrToFloat(Form1->Memol->Text);

```



```

        break;
        case '+':jyoujo=StrToFloat(Form1->Memol->Text);
        break;
        case '-':jyoujo=-1*StrToFloat(Form1->Memol->Text);
        break;
    }
    Form1->Memol->Text=" ";
    op='*';
    break;
case14:switch(op){                // /の処理
    case '*':jyoujo=jyoujo*StrToFloat(Form1->Memol->Text);
        break;
    case '/':jyoujo=jyoujo/StrToFloat(Form1->Memol->Text);
        break;
    case '+':jyoujo=StrToFloat(Form1->Memol->Text);
        break;
    case '-':jyoujo=-1*StrToFloat(Form1->Memol->Text);
        break;
}
Form1->Memol->Text=" ";
op='/';
break;
case15:Form1->Memol->Text="0";      //AC(All Clear)の処理
shori_flag1=0;
break;
case16:switch(op){                // =の処理
    case '*':jyoujo=jyoujo*StrToFloat(Form1->Memol->Text);
        kei=kei+jyoujo;
        break;
    case '/':jyoujo=jyoujo/StrToFloat(Form1->Memol->Text);
        kei=kei+jyoujo;
        break;
    case '+':kei=kei+StrToFloat(Form1->Memol->Text);
        break;
    case '-':kei=kei-StrToFloat(Form1->Memol->Text);
        break;
}
Form1->Memol->Text=FormatFloat("#0.####0",kei); //小数第5位まで表示
shori_flag1=0;
break;
case17:Form1->Memol->Text=Form1->Memol->Text + ".";break; // .の処理
}
}
//-----
void__fastcallTForm1::SpeedButton1Click(TObject *Sender)
{
    TSpeedButton*btn;           //Sender を TSpeedButtonとして
    btn= (TSpeedButton*)Sender; //扱えるように定義する

```

```

ButtonCheck(btn);
ButtonShori(btn);
}

クラス化
【イベントハンドラ】
//-----
// 加減乗除算 === クラス化
//-----
void__fastcall TForm1::FormCreate(TObject*Sender)
{
    Mem0->Text="0";
    Initial_Value();
}
//-----
void__fastcall TForm1::SpeedButton1Click(TObject *Sender)
{
    TSpeedButton*btn;          //Sender を TSpeedButton として
    btn= (TSpeedButton*)Sender; //扱えるように定義する

    ButtonCheck(btn);

    if(shori_flag2==1){ //不適當なボタンが押された場合の処理
        shori_flag2= 0;
        shori_flag1= 0;
    }
    else{
        ButtonShori(btn);
    }
}

```

【ヘッダファイル】

```

//-----
#ifndef mon4_22Unit1H
#define mon4_22Unit1H
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <Buttons.hpp>
#include <ExtCtrls.hpp>
//-----
class TForm1:public TForm
{
__published: //IDE 管理のコンポーネント
    TPanel*Panel1;
    TPanel*Panel2;
    TMem0*Mem0;
    TSpeedButton *SpeedButton1;

```

```

TSpeedButton *SpeedButton2;
TSpeedButton *SpeedButton3;
TSpeedButton *SpeedButton4;
TSpeedButton *SpeedButton5;
TSpeedButton *SpeedButton6;
TSpeedButton *SpeedButton7;
TSpeedButton *SpeedButton8;
TSpeedButton *SpeedButton9;
TSpeedButton *SpeedButton10;
TSpeedButton *SpeedButton11;
TSpeedButton *SpeedButton12;
TSpeedButton *SpeedButton13;
TSpeedButton *SpeedButton14;
TSpeedButton *SpeedButton15;
TSpeedButton *SpeedButton16;
TSpeedButton *SpeedButton17;
void__fastcallFormCreate(TObject *Sender);
void__fastcallSpeedButton1Click(TObject*Sender);
private:          // ユーザ宣言
floatkei;        //合計値
floatjyoujo;     //乗除計算領域
floatw;          //作業用
intshori_flag1; //計算開始のチェック(初期値:0, 計算途中:1)
intshori_flag2; //不適当なボタンを押したかどうかのチェック(初期値:0, 押した:1)
AnsiStrings;    //作業用
AnsiStringwkazu; //退避用
charop;         //演算記号保存用
public:          //ユーザ宣言
fastcallTForm1(TComponent*Owner);
void__fastcallTForm1::Initial_Value(void); //初期値設定
void__fastcallButtonCheck(TSpeedButton*btn); //ボタンのチェック
void__fastcallButtonShori(TSpeedButton*btn); //ボタンの処理
};
//-----
externPACKAGE TForm1*Form1;
//-----
// 初期値設定
void__fastcallTForm1::Initial_Value(void)
{
    kei=0;
    jyoujo=1;
    shori_flag1=0;
    shori_flag2=0;
}
//-----
// ボタンのチェック
void__fastcallTForm1::ButtonCheck(TSpeedButton*btn)
{
    if(shori_flag1==0){ //計算開始の場合の処理
        kei=0;

```

```

Memo1->Text=" "; //入力ボックスの初期化
wkazu=" ";
op='+';
jyoujo=1;
shori_flag1= 1; //shori_flag1 を 1 にセット
switch(btn->Tag){ //数値の入力前に記号ボタンを押した場合の処理
    case11: // +
    case12: // -
    case13: // *
    case14: // /
    case16: // =
    case17: // .
        ShowMessage("不適当なボタンが押された!");
        shori_flag2=1;
        break;
}
}
}
}

```

//-----
// ボタンの処理

```

void__fastcall TForm1::ButtonShori(TSpeedButton*btn)
{
    switch(btn->Tag){
        case 1: Memo1->Text = Memo1->Text+"1";break; // 1の処理
        case 2: Memo1->Text = Memo1->Text+"2";break; // 2の処理
        case 3: Memo1->Text = Memo1->Text+"3";break; // 3の処理
        case 4: Memo1->Text = Memo1->Text+"4";break; // 4の処理
        case 5: Memo1->Text = Memo1->Text+"5";break; // 5の処理
        case 6: Memo1->Text = Memo1->Text+"6";break; // 6の処理
        case 7: Memo1->Text = Memo1->Text+"7";break; // 7の処理
        case 8: Memo1->Text = Memo1->Text+"8";break; // 8の処理
        case 9: Memo1->Text = Memo1->Text+"9";break; // 9の処理
        case10:Memo1->Text=Memo1->Text+"0";break; // 0の処理
        case11:switch(op){ // +の処理
            case '*': jyoujo=jyoujo*StrToFloat(Memo1->Text);
                kei=kei+jyoujo;
                break;
            case '/': jyoujo=jyoujo/StrToFloat(Memo1->Text);
                kei=kei+jyoujo;
                break;
            case '+': kei=kei+StrToFloat(Memo1->Text);
                break;
            case '-': kei=kei-StrToFloat(Memo1->Text);
                break;
        }
        Memo1->Text=" ";
        op='+';
        break;
        case12:switch(op){ // -の処理
            case '*': jyoujo=jyoujo*StrToFloat(Memo1->Text);

```

```

        kei=kei+jyoujo;
        break;
    case '/': jyoujo=jyoujo/StrToFloat(Memol->Text);
        kei=kei+jyoujo;
        break;
    case '+': kei=kei+StrToFloat(Memol->Text);
        break;
    case '-': kei=kei-StrToFloat(Memol->Text);
        break;
}
Memol->Text=" ";
op='-';
break;
case13:switch(op){
// *の処理
    case '*': jyoujo=jyoujo*StrToFloat(Memol->Text);
        break;
    case '/': jyoujo=jyoujo/StrToFloat(Memol->Text);
        break;
    case '+': jyoujo=StrToFloat(Memol->Text);
        break;
    case '-': jyoujo=-1*StrToFloat(Memol->Text);
        break;
}
Memol->Text=" ";
op='*';
break;
case14:switch(op){
// /の処理
    case '*': jyoujo=jyoujo*StrToFloat(Memol->Text);
        break;
    case '/': jyoujo=jyoujo/StrToFloat(Memol->Text);
        break;
    case '+': jyoujo=StrToFloat(Memol->Text);
        break;
    case '-': jyoujo=-1*StrToFloat(Memol->Text);
        break;
}
Memol->Text=" ";
op='/';
break;
case15:Memol->Text="0"; //AC(AllClear)の処理
shori_flag1=0;
break;
case16:switch(op){
// =の処理
    case '*': jyoujo=jyoujo*StrToFloat(Memol->Text);
        kei=kei+jyoujo;
        break;
    case '/': jyoujo=jyoujo/StrToFloat(Memol->Text);
        kei=kei+jyoujo;
        break;
    case '+': kei=kei+StrToFloat(Memol->Text);

```

```

        break;
        case '-': kei=kei-StrToFloat(Mem01->Text);
        break;
    }
    Mem01->Text=FormatFloat("#0.####0",kei); //小数第 5 位まで表示
    shori_flag1=0;
    break;
    case17:Mem01->Text=Mem01->Text+".";break; // .の処理
}
}

```

第5章 練習問題 1

【イベントハンドラ】

```

//-----
TDateTime t1, t2;
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    t1=Now();
    Edit1->Text=Now();
    Timer1->Enabled=true;
}
//-----
void __fastcall TForm1::Button2Click(TObject *Sender)
{
    t2=Now();
    Edit3->Text=t2-t1;
    Timer1->Enabled=false;
}
//-----
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    Edit2->Text=Now();
}
//-----
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    Timer1->Enabled=false;
    Edit1->Text="0";
    Edit2->Text="0";
    Edit3->Text="0";
}
//-----

```

第5章 練習問題 2

【イベントハンドラ】

```

//-----
// 現年月日の設定と NextYear, cdown, y, m, d オブジェクト
TDateTime SetDate(Now()), NextYear, cdown;
Wordy, m, d;

```

```

//-----
void __fastcall TForm1::Timer1Timer(TObject*Sender)
{
    WordHour, Min, Sec, MSec; //Hour, Min, Sec, MSec オブジェクト

    cdown=NextYear-Now(); //目標年月日までの日数計算
    DecodeTime(cdown, Hour, Min, Sec, MSec); //時間, 分, 秒の取り出し

    Label5->Caption=IntToStr(cdown); //日数表示
    Label7->Caption=IntToStr(Hour); //時間表示
    Label9->Caption=IntToStr(Min); //分表示
    Label11->Caption=IntToStr(Sec); //秒表示
}
//-----
void __fastcall TForm1::FormCreate(TObject*Sender)
{
    DecodeDate(Now(), y, m, d); //今年の取り出し
    y++; //今年に1年をカウントする
    NextYear=AnsiString(IntToStr(y)+"/1/1"); //翌年の1月1日を設定
    Label2->Caption=IntToStr(y)+"年まであと"; //目標年の表示
}
//-----

```

第II部 グラフィックス

第6章 練習問題 1

```

//-----
void __fastcall TForm1::h1Click(TObject *Sender)
{
    Canvas->MoveTo(50,100); //Tの作成
    Canvas->LineTo(150,100);
    Canvas->MoveTo(100,100);
    Canvas->LineTo(100,200);

    Canvas->MoveTo(200,100); //Dの作成
    Canvas->LineTo(200,200);
    Canvas->MoveTo(200,100);
    Canvas->LineTo(300,130);
    Canvas->MoveTo(200,200);
    Canvas->LineTo(300,170);
    Canvas->MoveTo(300,130);
    Canvas->LineTo(300,170);

    Canvas->MoveTo(350,100); //Uの作成
    Canvas->LineTo(350,200);
    Canvas->MoveTo(450,100);
    Canvas->LineTo(450,200);
    Canvas->MoveTo(350,200);
}

```

```

Canvas->LineTo(450,200);
}
//-----
void__fastcallTForm1::Exit1Click(TObject*Sender)
{
Close();
}
//-----

```

第6章 練習問題 2

```

//-----
void__fastcallTForm1::h1Click(TObject *Sender)
{
Canvas->Rectangle(10,10,200,100);
Canvas->Ellipse(50,30,80,60);
Canvas->Ellipse(120,30,150,60);
Canvas->MoveTo(60,80);
Canvas->LineTo(140,80);
}
//-----
void__fastcallTForm1::Exit1Click(TObject*Sender)
{
Close();
}
//-----

```

第6章 練習問題 3

```

//-----
void__fastcallTForm1::Exit1Click(TObject*Sender)
{
Close();
}
//-----
void__fastcallTForm1::N1Click(TObject *Sender)
{
Canvas->MoveTo(10,10);
Canvas->LineTo(200,200);
}
//-----
void__fastcallTForm1::N2Click(TObject *Sender)
{
Canvas->Rectangle(30,30,200,150);
}
//-----
void__fastcallTForm1::N3Click(TObject *Sender)
{
Canvas->Ellipse(10,10,100,100);
}
//-----

```


第6章 練習問題 4

```
//-----  
void__fastcallTForm1::Exit1Click(TObject*Sender)  
{  
    Close();  
}  
//-----  
void__fastcallTForm1::N1Click(TObject *Sender)  
{  
    Canvas->MoveTo(10,10);  
    Canvas->LineTo(200,200);  
}  
//-----  
void__fastcallTForm1::N2Click(TObject *Sender)  
{  
    Canvas->Rectangle(30,30,200,150);  
}  
//-----  
void__fastcallTForm1::N3Click(TObject *Sender)  
{  
    Canvas->Ellipse(10,10,100,100);  
}  
//-----  
void__fastcallTForm1::N5Click(TObject *Sender)  
{  
    Canvas->Pen->Color=clRed;  
}  
//-----  
void__fastcallTForm1::N6Click(TObject *Sender)  
{  
    Canvas->Pen->Color=clBlue;  
}  
//-----  
void__fastcallTForm1::N7Click(TObject *Sender)  
{  
    Canvas->Pen->Color=clLime;  
}  
//-----  
void__fastcallTForm1::N8Click(TObject *Sender)  
{  
    Canvas->Pen->Color=0x0098f289;  
}  
//-----  
void__fastcallTForm1::N9Click(TObject *Sender)  
{  
    Canvas->Pen->Color=0x009ff4219;  
}  
//-----
```

第6章 練習問題 5

```
//-----  
void__fastcallTForm1::h1Click(TObject *Sender)  
{  
    Canvas->Brush->Style=bsSolid;  
    Canvas->Brush->Color=clFuchsia;  
    Canvas->Rectangle(50,50,450,300);  
  
    Canvas->Brush->Color=clTeal;  
    Canvas->Rectangle(100,100,400,250);  
  
    Canvas->Brush->Color=clMaroon;  
    Canvas->Rectangle(150,150,350,200);  
}  
//-----  
void__fastcallTForm1::Exit1Click(TObject*Sender)  
{  
    Close();  
}  
//-----
```

第6章 練習問題 6

```
//-----  
void__fastcallTForm1::h1Click(TObject *Sender)  
{  
    Canvas->Pen->Width=10;  
    Canvas->Ellipse(50,50,300,300);  
  
    Canvas->Pen->Width=5;  
    Canvas->Ellipse(100,100,250,250);  
  
    Canvas->Pen->Width=2;  
    Canvas->Ellipse(150,150,200,200);  
}  
//-----  
void__fastcallTForm1::Exit1Click(TObject*Sender)  
{  
    Close();  
}  
//-----
```

第6章 練習問題 7

```
//-----  
void__fastcallTForm1::h1Click(TObject *Sender)  
{  
    Canvas->MoveTo(0,250);  
    Canvas->LineTo(600,250);  
}
```

```

Canvas->Brush->Color=c1Lime;

Canvas->Brush->Style=bsSolid;
Canvas->Rectangle(50,125,100,250);
Canvas->TextOut(75,270," 4月");
Canvas->TextOut(75,30,"125");

Canvas->Brush->Style=bsBDiagonal;
Canvas->Rectangle(150,112,200,250);
Canvas->TextOut(175,270," 5月");
Canvas->TextOut(175,30,"138");

Canvas->Brush->Style=bsFDiagonal;
Canvas->Rectangle(250,65,300,250);
Canvas->TextOut(275,270," 6月");
Canvas->TextOut(275,30,"185");

Canvas->Brush->Style=bsCross;
Canvas->Rectangle(350,124,400,250);
Canvas->TextOut(375,270," 7月");
Canvas->TextOut(375,30,"126");

Canvas->Brush->Style=bsDiagCross;
Canvas->Rectangle(450,102,500,250);
Canvas->TextOut(475,270," 8月");
Canvas->TextOut(475,30,"148");

Canvas->Brush->Style=bsVertical;
Canvas->Rectangle(550,120,600,250);
Canvas->TextOut(575,270," 9月");
Canvas->TextOut(575,30,"130");
}
//-----
void__fastcallTForm1::Exit1Click(TObject*Sender)
{
    Close();
}
//-----

```

第6章 練習問題 8

```

//-----
void__fastcallTForm1::Exit1Click(TObject*Sender)
{
    Close();
}
//-----
void__fastcallTForm1::N5Click(TObject *Sender)
{
    Canvas->Pen->Color=c1Red;
}

```

```

}
//-----
void__fastcallTForm1::N6Click(TObject *Sender)
{
Canvas->Pen->Color=clBlue;
}
//-----
void__fastcallTForm1::N7Click(TObject *Sender)
{
Canvas->Pen->Color=clLime;
}
//-----
void__fastcallTForm1::N8Click(TObject *Sender)
{
Canvas->Pen->Color=0x0098f289;
}
//-----
void__fastcallTForm1::N9Click(TObject *Sender)
{
Canvas->Pen->Color=0x009ff4219;
}
//-----
void__fastcallTForm1::h1Click(TObject *Sender)
{
Canvas->Rectangle(50,50,200,200);
}
//-----
void__fastcallTForm1::N14Click(TObject*Sender)
{
Canvas->Brush->Color=clPurple;
}
//-----
void__fastcallTForm1::N15Click(TObject*Sender)
{
Canvas->Brush->Color=clSilver;
}
//-----
void__fastcallTForm1::N16Click(TObject*Sender)
{
Canvas->Brush->Color=clGray;
}
//-----
void__fastcallTForm1::N17Click(TObject*Sender)
{
Canvas->Brush->Color=0x00f8d3c5;
}
//-----
void__fastcallTForm1::N18Click(TObject*Sender)
{
Canvas->Brush->Color=0x001f331f;
}

```

```

}
//-----
void__fastcallTForm1::N2Click(TObject *Sender)
{
    Canvas->Brush->Style=bsSolid;
}
//-----
void__fastcallTForm1::N3Click(TObject *Sender)
{
    Canvas->Brush->Style=bsClear;
}
//-----
void__fastcallTForm1::N10Click(TObject*Sender)
{
    Canvas->Brush->Style=bsHorizontal;
}
//-----
void__fastcallTForm1::N11Click(TObject*Sender)
{
    Canvas->Brush->Style=bsVertical;
}
//-----
void__fastcallTForm1::N12Click(TObject*Sender)
{
    Canvas->Brush->Style=bsCross;
}
//-----

```

第6章 練習問題 9

```

//-----
void__fastcallTForm1::Exit1Click(TObject*Sender)
{
    Close();
}
//-----
void__fastcallTForm1::N5Click(TObject *Sender)
{
    Canvas->Font->Color=clRed;
}
//-----
void__fastcallTForm1::N6Click(TObject *Sender)
{
    Canvas->Font->Color=clBlue;
}
//-----
void__fastcallTForm1::N7Click(TObject *Sender)
{
    Canvas->Font->Color=clLime;
}

```

```

//-----
void__fastcallTForm1::N8Click(TObject *Sender)
{
    Canvas->Font->Color=0x0098f289;
}
//-----
void__fastcallTForm1::N9Click(TObject *Sender)
{
    Canvas->Font->Color=0x009ff4219;
}
//-----
void__fastcallTForm1::h1Click(TObject *Sender)
{
    Canvas->Font->Size=40;
    Canvas->TextOut(50,50," T D U");
}
//-----
void__fastcallTForm1::N14Click(TObject *Sender)
{
    Canvas->Brush->Color=clPurple;
}
//-----
void__fastcallTForm1::N15Click(TObject *Sender)
{
    Canvas->Brush->Color=clSilver;
}
//-----
void__fastcallTForm1::N16Click(TObject *Sender)
{
    Canvas->Brush->Color=clGray;
}
//-----
void__fastcallTForm1::N17Click(TObject *Sender)
{
    Canvas->Brush->Color=0x00f8d3c5;
}
//-----
void__fastcallTForm1::N18Click(TObject *Sender)
{
    Canvas->Brush->Color=0x001f331f;
}
//-----

```

第6章 練習問題 10 省略

第7章 練習問題 1

```

//-----
void__fastcallTForm1::Show1Click(TObject *Sender)
{

```

```

int x;

// グラフを描く
Canvas->Pen->Color=clRed;
Canvas->Pen->Width=2;

for(x = -5; x <= 4; x++){
    Canvas->MoveTo(50+50*(x+5), 300-10*x*x);
    Canvas->LineTo(50+50*(x+6), 300-10*(x+1)*(x+1));
}
}

//-----
void __fastcall TForm1::Exit1Click(TObject*Sender)
{
    Close();
}

//-----
void __fastcall TForm1::FormPaint(TObject*Sender)
{
    inti;

    Canvas->Pen->Width=2;
    Canvas->MoveTo(0, 300);
    Canvas->LineTo(600, 300); // x 軸を引く

    Canvas->MoveTo(300, 0);
    Canvas->LineTo(300, 350); // y 軸を引く

    Canvas->Font->Size=10;
    Canvas->TextOut(550, 310, " X ");
    Canvas->TextOut(20, 20, " Y ");

    // 目盛りをつける
    Canvas->Pen->Width=1;
    for(i = 0; i <= 10; i++) {
        Canvas->MoveTo(50+50*i, 0);
        Canvas->LineTo(50+50*i, 300);
    }

    for(i = 0; i <= 2; i++) {
        Canvas->MoveTo(50, 100*i);
        Canvas->LineTo(550, 100*i);
    }

    Canvas->TextOut(300, 310, " 0 ");

    for(i = 1; i <= 3; i++) {
        Canvas->TextOut(30, 300-100*i, i*10);
    }
}
}

```

```

//-----
第7章 練習問題 2
//-----
void__fastcallTForm1::Show1Click(TObject*Sender)
{
    intwcos(int );
    intx1,x2,y1,y2;

    Canvas->Pen->Width=2;
    Canvas->Pen->Color=clRed;

    for(x1=0;x1<=60;x1++){
        y1=wcos(x1);
        x2=x1+1;
        y2=wcos(x2);

        Canvas->MoveTo(x1+50,y1);
        Canvas->LineTo(x2+50,y2);
    }
}

intwcos(intx)
{
    floatatai;
    inty;
    atai=cos(3.14/180*x);
    y=150-100*atai;
    return(y);
}

//-----
void__fastcallTForm1::Exit1Click(TObject*Sender)
{
    Close();
}

//-----
void__fastcallTForm1::FormPaint(TObject*Sender)
{
    inti;

    Canvas->Pen->Width=2;
    Canvas->MoveTo(30,150);
    Canvas->LineTo(410,150); // x 軸を引く

    Canvas->MoveTo(50,50);
    Canvas->LineTo(50,250); // y 軸を引く

    Canvas->Font->Size=10;
    Canvas->TextOut(420,130,"X");
}

```



```

Canvas->TextOut(20,30," Y ");

//目盛りを付ける
Canvas->Pen->Width=1;

Canvas->MoveTo(50,50);
Canvas->LineTo(410,50);

Canvas->MoveTo(50,100);
Canvas->LineTo(410,100);

Canvas->MoveTo(50,200);
Canvas->LineTo(410,200);

Canvas->MoveTo(50,250);
Canvas->LineTo(410,250);

for(i = 1; i <=4; i++) {
    Canvas->MoveTo(50+90*i,50);
    Canvas->LineTo(50+90*i,250);
}

Canvas->TextOut(30,45,"1");
Canvas->TextOut(20,95,"0.5");
Canvas->TextOut(20,145,"0");
Canvas->TextOut(20,195,"-0.5");
Canvas->TextOut(30,245,"-1");

for(i = 1; i <=4; i++) {
    Canvas->TextOut(55+90*i,160,i*90);
}
}
//-----

```

第7章 練習問題 3

```

//-----
void __fastcall TForm1::Show1Click(TObject*Sender)
{
    int wcos(int),wsin(int);
    int i,j,x1,x2,y1,y2;

    Canvas->Pen->Width=2;
    Canvas->Pen->Color=clRed;

    for(i = 0; i <360; i++){

        x1=wsin(i);
        y1=wcos(i);

```

```

    j = i +1;
    x2=wsin(j);
    y2=wcoss(j);

    Canvas->MoveTo(x1,y1);
    Canvas->LineTo(x2,y2);
}
}

intwsin(intx)
{
    floatatai;
    inty;
    atai=sin(3.14/180*x);
    y=150+100*atai;
    return(y);
}

intwcos(intx)
{
    floatatai;
    inty;
    atai=cos(3.14/180*x);
    y=150-100*atai;
    return(y);
}

//-----
void__fastcallTForm1::Exit1Click(TObject*Sender)
{
    Close();
}

//-----
void__fastcallTForm1::FormPaint(TObject*Sender)
{
    Canvas->Pen->Width=2;
    Canvas->MoveTo(50,150);
    Canvas->LineTo(250,150); // x 軸を引く

    Canvas->MoveTo(150,50);
    Canvas->LineTo(150,250); // y 軸を引く

    Canvas->Font->Size=10;
    Canvas->TextOut(210,155," X ");
    Canvas->TextOut(155,70," Y ");

    //目盛りを付ける
    Canvas->Pen->Width=1;

    Canvas->MoveTo(50,50);
    Canvas->LineTo(250,50);

```

```

Canvas->MoveTo(50,100);
Canvas->LineTo(250,100);

Canvas->MoveTo(50,200);
Canvas->LineTo(250,200);

Canvas->MoveTo(50,250);
Canvas->LineTo(250,250);

Canvas->MoveTo(50,50);
Canvas->LineTo(50,250);

Canvas->MoveTo(100,50);
Canvas->LineTo(100,250);

Canvas->MoveTo(200,50);
Canvas->LineTo(200,250);

Canvas->MoveTo(250,50);
Canvas->LineTo(250,250);

Canvas->TextOut(20,155,"-1");
Canvas->TextOut(155,155,"0");
Canvas->TextOut(255,155,"1");

Canvas->TextOut(155,35,"1");
Canvas->TextOut(155,255,"-1");
}
//-----

```

第7章 練習問題 4

```

//-----
void __fastcall TForm1::Show1Click(TObject*Sender)
{
    int x;

    // グラフを描く
    Canvas->Pen->Color=clRed;
    Canvas->Pen->Width=2;

    for(x = -5; x <= 4; x++) {
        Canvas->MoveTo(50+50*(x+5), 150-(x*x*x));
        Canvas->LineTo(50+50*(x+6), 150-((x+1)*(x+1)*(x+1)));
    }
}
//-----
void __fastcall TForm1::Exit1Click(TObject*Sender)
{

```

```

Close();
}
//-----
void__fastcallTForm1::FormPaint(TObject*Sender)
{
    inti;

    Canvas->Pen->Width=2;
    Canvas->MoveTo(50,150);
    Canvas->LineTo(550,150); //x軸を引く

    Canvas->MoveTo(300,0);
    Canvas->LineTo(300,300); //y軸を引く

    Canvas->Font->Size=10;
    Canvas->TextOut(530,170,"X");
    Canvas->TextOut(330,20,"Y");

    //目盛りを付ける
    Canvas->Pen->Width=1;
    for(i = 0; i<=10;i++) {
        Canvas->MoveTo(50+50*i,0);
        Canvas->LineTo(50+50*i,300);
    }

    for(i = 0; i<=6;i++) {
        Canvas->MoveTo(50,50*i);
        Canvas->LineTo(550,50*i);
    }

    Canvas->Font->Size=8;
    for(i = 0; i<=6;i++) {
        Canvas->TextOut(310,305-50*i,(i-3)*50);
    }
}
//-----

```

第7章 練習問題 5

```

//-----
void__fastcallTForm1::Show1Click(TObject*Sender)
{
    intx1,y1,x2,y2;
    floatx,y,t;
    //グラフを描く
    Canvas->Pen->Color=clRed;
    Canvas->Pen->Width=2;

    y=196;
    y1=196;

```

```

x1=0;
t=0;
while(y >= 0){
    y=196-(98/2*t*t);
    x=120*t;
    x2=x;
    y2=y;

    Canvas->MoveTo(50+x1,200-y1);
    Canvas->LineTo(50+x2,200-y2);

    x1=x2;
    y1=y2;
    t=t+0.1;
}
}
//-----
void __fastcall TForm1::Exit1Click(TObject*Sender)
{
    Close();
}
//-----
void __fastcall TForm1::FormPaint(TObject*Sender)
{
    inti;

    Canvas->Pen->Width=2;
    Canvas->MoveTo(50,200);
    Canvas->LineTo(300,200); // x 軸を引く

    Canvas->MoveTo(50,0);
    Canvas->LineTo(50,200); // y 軸を引く

    Canvas->Font->Size=10;
    Canvas->TextOut(280,210," X ");
    Canvas->TextOut(30,20," Y ");

    //目盛りを付ける
    Canvas->Pen->Width=1;
    for(i = 1; i<=5;i++) {
        Canvas->MoveTo(50+50*i,0);
        Canvas->LineTo(50+50*i,200);
    }

    for(i = 0; i<=4;i++) {
        Canvas->MoveTo(50,50*i);
        Canvas->LineTo(300,50*i);
    }

    Canvas->Font->Size=8;
}

```

```

for(i = 1; i <= 5; i++) {
    Canvas->TextOut(50+50*i, 205, i*5);
}

for(i = 1; i <= 4; i++) {
    Canvas->TextOut(30, 205-50*i, i*5);
}
}
//-----

```

第8章 練習問題 1

```

//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1*Form1;
int iro=1;
//-----
void __fastcall TForm1::Exit1Click(TObject*Sender)
{
    Close();
}
//-----
void __fastcall TForm1::Show1Click(TObject*Sender)
{
    Timer1->Enabled=true;
}
//-----
void __fastcall TForm1::Timer1Timer(TObject*Sender)
{
    switch(iro){
        case1: Shape2->Brush->Color=clBlue;
                Shape3->Brush->Color=clWhite;
                Shape4->Brush->Color=clWhite;
                break;

        case2: Shape2->Brush->Color=clWhite;
                Shape3->Brush->Color=clYellow;
                Shape4->Brush->Color=clWhite;
                break;

        case3: Shape2->Brush->Color=clWhite;
                Shape3->Brush->Color=clWhite;
                Shape4->Brush->Color=clRed;
                break;
    }

    iro++;
    if(iro > 3) {
        iro=1;
    }
}

```

```

}
}
//-----

```

第8章 練習問題 2

```

//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1*Form1;
intsw=1;
//-----
void__fastcallTForm1::FormMouseMove(TObject*Sender,TShiftStateShift,
    intX,intY)
{
    Shap1->Left=X-25;
    Shap1->Top=Y-25;
}
//-----
void__fastcallTForm1::Exit1Click(TObject*Sender)
{
    Close();
}
//-----

```

第8章 練習問題 3

```

//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1*Form1;
intsw=1;
//-----
void__fastcallTForm1::FormMouseMove(TObject*Sender,TShiftStateShift,
    intX,intY)
{
    Shap1->Left=X-25;
    Shap1->Top=Y-25;
}
//-----
void__fastcallTForm1::Exit1Click(TObject*Sender)
{
    Close();
}
//-----
void__fastcallTForm1::Shap1MouseDown(TObject*Sender,
    TMouseButtonButton,TShiftStateShift,intX,intY)
{
    if(Button==mbLeft){
        Shap1->Visible=true;
    }
}

```

```

    if(Button==mbRight){
        Shapel->Visible=false;
    }
}
//-----
void__fastcallTForm1::FormMouseDown(TObject*Sender,TMouseButtonButton,
    TShiftStateShift,intX,intY)
{
    if(Button==mbLeft){
        Shapel->Visible=true;
    }

    if(Button==mbRight){
        Shapel->Visible=false;
    }
}
//-----

```

第8章 練習問題 4

```

//-----
#pragma package(smart_init)
#pragma resource"*.dfm"
TForm1*Form1;
floaty=0,t=0;
//-----
void__fastcallTForm1::FormPaint(TObject*Sender)
{
    inti;

    Canvas->Pen->Width=2;
    Canvas->MoveTo(50,250);
    Canvas->LineTo(300,250); //x軸を引く

    Canvas->MoveTo(50,50);
    Canvas->LineTo(50,250); //y軸を引く

    Canvas->Font->Size=10;
    Canvas->TextOut(290,260,"X");
    Canvas->TextOut(30,20,"Y");
    Canvas->TextOut(40,260,"0");

    //目盛り
    Canvas->Pen->Width=1;
    for(i=1;i<=5;i++) {
        Canvas->MoveTo(50+50*i,50);
        Canvas->LineTo(50+50*i,250);
    }
}

```



```

for(i = 0; i<=3;i++) {
    Canvas->MoveTo(50,50+50*i);
    Canvas->LineTo(300,50+50*i);
}

Canvas->Font->Size=8;
for(i=1;i<=4; i++) {
    Canvas->TextOut(20,250-50*i,i*5);
}

for(i=1;i<=5; i++) {
    Canvas->TextOut(50+50*i,260,i*5);
}
}
//-----
void__fastcallTForm1::Exit1Click(TObject*Sender)
{
    Close();
}
//-----
void__fastcallTForm1::Show1Click(TObject*Sender)
{
    Timer1->Enabled=true;
}
//-----
void__fastcallTForm1::Timer1Timer(TObject*Sender)
{
    intx1,y1;
    floatx;

    y = (98/2)*t*t;
    x = 120*t;

    x1=x;
    y1=y;

    Shape2->Left=50+x1-5;
    Shape2->Top= 50+y1-5;

    t = t + 0.1;

    if(y> 196){
        y = 0;
        t = 0;
    }
}
//-----

```

第 部 やさしい数値計算入門

第9章 練習問題 1

- (1) `PaintBox1->Canvas->Pen->Color=clGreen;`
- (2) `PaintBox1->Canvas->Pixels[xg][yg]=clBlack;`
- (3) たて線 `doubledx=PBW/20;`
よこ線 `doubledy=PBH/20;`
- (4) `PaintBox1->Canvas->TextOut(PBW-10,y0+5,"x");`
`PaintBox1->Canvas->TextOut(PBW/2.0-10,5,"y");`
- (5) たて線, よこ線を 20 等分したときは次のように cx, cy を直して
`//x,y 軸の 1 目盛り当たりの値を cx,cy とする`
`// (グラフ描画の係数 cx,cy)`
`double cx=PBW/20.0,cy=PBH/20.0;`

次の for 文の中の x の範囲を $x=-10, x \leq 10$ に直し, それぞれの関数の式を $y=x*x$; の式の位置に記述して実行する.

```
//関数の計算とグラフ表示
for(x=-10;x<=10;x=x+0.01){
//x を-5 から 5 まで 0.01 きざみで変化して y の値を求める
    y=-x*x;
    xg=x0+cx*x;    //x をグラフ描画座標 xg に変換
    yg=y0-cy*y;    //y をグラフ描画座標 yg に変換
    PaintBox1->Canvas->Pixels[xg][yg]=clRed;
    //xg , yg の位置に赤で点を描く
    PaintBox1->Canvas->Pixels[xg][yg]=clBlack;
    //xg , yg の位置に赤で点を描く
}
```

式の書き方

```
y=-x*x;
y=x*x+3;
y=x*x-3;
y=(x+1)*(x-3);
y=(x+4)*(x+1);
y=x+3;
y=x-3;
y=x*x*x;
y=-x*x*x;
y=(x-1)*(x-2)*(x-3);
```

第9章 練習問題 2

- (1) 中心は $(0, 0)$, 半径が 3 の円の式は媒介変数表示では

$$\begin{cases} x=3\cos(t) \\ y=3\sin(t) \end{cases}$$

中心が $(-3, -1)$, 半径が 2 の円は媒介変数表示では

$$\begin{cases} x=2\cos(t)-3 \\ y=2\sin(t)-1 \end{cases}$$

中心が $(-2, 1)$, 長径が 3 , 短径が 1 の楕円の媒介変数表示は

$$\begin{cases} x=3\cos(t)-2 \\ y=\sin(t)+1 \end{cases}$$

関数の計算とグラフ描画式の表示部分を次のように修正する .

```
//関数の計算とグラフ描画
double pai=3.14159;
for(double t=0;t<=2*pi;t=t+0.01){
    switch(fg)
    {
        case1:x=3*cos(t); //fg=1 のときの処理
              y=3*sin(t);
              break;

        case2:x=2*cos(t)-3;
              y=2*sin(t)-1;
              break;

        case3:x=3*cos(t)-2;
              y=1*sin(t)+1;
              break;
    }
}
```

式の表示の部分を次のように修正する .

```
void__fastcallTForm1::RadioButton1Click(TObject*Sender)
{
    //式の表示: 中心が(0,0) , 半径 3 の円
    fg=1;
    Label2->Caption="式:x=3cos(t) y=3sin(t)";
}
//-----
void__fastcallTForm1::RadioButton2Click(TObject*Sender)
{
    //式の表示: 中心が(-3,-1) , 半径 2 の円
    fg=2;
    Label2->Caption="式:x=2cos(t)-3y=2sin(t)-1";
}
//-----
void__fastcallTForm1::RadioButton3Click(TObject*Sender)
{
    //式の表示: 中心が(-2,1)長径 3 , 短径 1 の楕円
    fg=3;
    Label2->Caption="式:x=3cos(t)-2y=sin(t)+1";
}
```

RadioButton1 , RadioButton2 , RadioButton3 の Caption を次のように修正する .

RadioButton1 の Caption を「中心が $(0, 0)$, 半径 3 の円」
 RadioButton2 の Caption を「中心が $(-3, -1)$, 半径 2 の円」
 RadioButton3 の Caption を「中心が $(-2, 1)$, 長径 3 , 短径 1 の楕円」

(2) サイクロイドの式

$$\begin{cases} x=at-\sin(t) \\ y=1-\cos(t) \end{cases}$$

において $a=0.1, 0.2, 0.3, 0.4, 0.5$ とおいた式を次の部分に追加する。
 t は, $t=0$ から $t=6*\text{pai}$ 変化させる。

```
//関数の計算とグラフ描画
double pai=3.14159;
for(double t=0;t<=6*pai;t=t+0.01){
    switch(fg)
    {
        case1:x=3*cos(t);    //fg=1 のときの処理
            y=3*sin(t);
            break;

        case2:x=2*cos(t)-3;
            y=2*sin(t)-1;
            break;

        case3:x=3*cos(t)-2;
            y=1*sin(t)+1;
            break;
        case4:x=0.1*t-sin(t);
            y=1-cos(t);
            break;
    }
}
```

RadioButton4 をフォーム上に配置して, Caption をサイクロイドとする。

RadioButton4 をダブルクリックして, RadioButton4Click イベントハンドラに以下のコードを記述する。

```
void __fastcall TForm1::RadioButton4Click(TObject*Sender)
{
    //式の表示:サイクロイド
    fg=4;
    Label2->Caption="式:x=0.1t-sin(t) y=1-cos(t)";
}
}
```

第9章 練習問題 3

(1) 正葉形曲線のユーザ関数の n の値を 2, 3, 4, 5, 6, 7, 8, 9, 10 と修正する。

```
//ユーザ関数 func1()の定義 (正葉形)
double __fastcall TForm1::func1(double t)
{
    int n=2;
    return(5.0*sin(n*t)*cos(t));
}
}
```

```
//ユーザ関数 func2()の定義
double __fastcall TForm1::func2(double t)
{
    int n=2;
    return(5.0*sin(n*t)*sin(t));
}
```

式を表示する部分も修正する .

```
void __fastcall TForm1::RadioButon1Click(TObject*Sender)
{
    //フラグ(fg)のセットと式の表示
    fg=1;
    Label2->Caption="式:r=sin(n ),n=2";
}
```

- (2) RadioButon1 の Checked プロパティを true にすると初めに RadioButon1 がチェック状態になるのでチェック忘れが防げて操作性がよくなる .

第9章 練習問題 4

グラフ描画の部分の y の式を $y=(x-a)*(x-a)$, $y=x*x-a$;に修正する .

```
//グラフ描画
PaintBox1->Canvas->Pen->Color=clRed; //Pen を赤色に設定
Memo1->Lines->Clear(); //Memo1 の内容をクリア
for(x=-5;x<=5;x+=0.1){ //x を-5 から 5 まで 0.01 ずつ変えて y の値を計算
    y=(x-a)*(x-a);
    xg=x0+cx*x; //x をグラフ座標 xg に変換
    yg=y0-cy*y; //y をグラフ座標 yg に変換
    PaintBox1->Canvas->LineTo(xg,yg); //関数のグラフを描く

    //x,y を Memo コンポーネントに表示する
    Memo1->Lines->Add(FormatFloat("###0.0",x)+" "+FormatFloat("###0.0",y));
}
```

フォーム上の式を表示する部分も修正する .

グラフ描画の部分の y の式を $y=(x-a)*(x-a)$ としたとき Label4 , Label5 の Caption を修正する .

Label4 の Caption を x , Label5 の Caption を y

グラフ描画の部分の y の式を $y=x*x-a$ としたとき Label4 , Label5 の Caption を修正する .

Label4 の Caption を x , Label6 を左へ移動して x^2 とする . Label5 を左へ移動して Caption を $-a$ とする .

第10章 練習問題 1

- (1) $x_1=2, x_2=3$
- (2) $x=6$
- (3) $1 \pm 2i$

第10章 練習問題 2

フォーム上に配置した Label11 の Caption の部分を次のように修正する .

「何人かの学生の体重データを入力して平均値と最大値，最小値を求める」

コードの部分の変数名 `sintyo` を全て `taijyu` に変更する。

変更の方法は個々に変更してもよいが，変更する部分が多いときはエディタの検索(F)→置換(R)を使うと早く，間違いなくできる。

第10章 練習問題 3

(1) `g(x)=2x`

`g(x)=2x-1`

`g(x)=2x-2`

`g(x)=4x-5`

`g(x)-2x+3`

`g(x)=2x+1`

`g(x)=2x-2`

`g(x)=3x2`

`g(x)=3x2+4x+1`

`g(x)=3x2-8x+1`

(2) フォーム上の `Label2` の `Caption` を「方程式:`y=x-4=0`」と変更する。

ユーザ関数の定義部分を次のように修正する。

//ユーザ関数 `f(x)` の定義 (メンバ関数の定義)

```
double __fastcall TForm1::f(double x)
```

```
{
```

```
    return(x*x-4);
```

```
}
```

//ユーザ関数 `g(x)` の定義 (メンバ関数の定義)

```
double __fastcall TForm1::g(double x)
```

```
{
```

```
    return(2*x);
```

```
}
```

初期値 `-4` のとき，近似解 `x=-2.00000`，初期値 `3` のとき，近似解 `x=2.00000`

~ についても同様に修正して，それぞれの初期値を与えて近似解を求める。

問題	初期値	近似解 x	初期値	近似解 x
	3	2.00000	-2	-1.00000
	-2	-1.00000	4	3.00000
	-1	-0.83197	3	2.68614
	-4	-2.61803	1	-0.18614
	0.5	1.56155	-2	-2.56155
	-3	-0.41421	4	2.41421
	2	1.00000		
	1	0.46557		
	4	3.00000	1	2.00000
	-2	-1.00000		

(3)	EPS	n	近似解 x
	0.001	4	-2.00000
	0.0001	5	-2.00000
	0.00001	5	-2.00000

から について同様に調べる .

第10章 練習問題 4

- (1)
- 1.73145 , 1.73145
 - 0.99976 , 1.99976
 - 0.99976 , 2.99976
 - 0.18652 , 2.68652
 - 2.61816 , -0.38184
 - 2.50152 , 1.56152
 - 0.41504 , 2.41504
 - 1.00024
 - 0.46582
 - 99976 , 1.99956 , 3.00024

(2)	EPS	n	近似解
	0.001	10	-1.73145
	0.0001	14	-1.73199
	0.00001	17	-1.73205

EPS の値を小さく設定すると , 繰り返し回数は増え真値に近い近似解が得られる . から について同様に調べる .

第10章 練習問題 5

- (1) 積分関数の定義部分を次のように修正する .

```
//関数 f(x)=x*x の定義 (メンバ関数の定義)
double __fastcall TForm1::f(double x)
{
    return(x*x);
}
```

フォーム上の Label2 の Caption を「関数 $f(x)=x$ の積分」に修正し , 「関数 $f(x)=x^2$ の積分」とする .

分割数 n=10 のとき , 積分値=2.68000
 分割数 n=100 のとき , 積分値=2.66680
 分割数 n=1000 のとき , 積分値=2.66667

積分関数 `return(x*x);`
 分割数 n=10 のとき , 積分値=2.33500
 分割数 n=100 のとき , 積分値=2.33335
 分割数 n=1000 のとき , 積分値=2.33333

積分関数 `return(x*x+2*x+1);`
分割数 n=10 のとき, 積分値=3.04500
分割数 n=100 のとき, 積分値=3.00045
分割数 n=1000 のとき, 積分値=3.00000

積分関数 `return(x*x+2*x+1);`
分割数 n=10 のとき, 積分値=2.33500
分割数 n=100 のとき, 積分値=2.33335
分割数 n=1000 のとき, 積分値=2.33333

(2) 定積分値

8/3
7/3
3
7/3

第10章 練習問題 6

(1) 積分関数の定義部分を次のように修正する.

```
//積分関数の定義(メンバ関数の定義)
double __fastcall TForm1::f(double x)
{
    return(x*x);
}
```

フォーム上の Label2 の Caption を「関数 $f(x)=x$ の積分」に修正し, Label3 を左に移動して「関数 $f(x)=x^2$ の積分」とする.

分割数 n=10 のとき, 積分値=2.66667
分割数 n=100 のとき, 積分値=2.66667
分割数 n=1000 のとき, 積分値=2.66667

積分関数 `return(x*x);`
分割数 n=10 のとき, 積分値=2.3333
分割数 n=100 のとき, 積分値=2.33333
分割数 n=1000 のとき, 積分値=2.33333

積分関数 `return(x*x+2*x+1);`
分割数 n=10 のとき, 積分値=3.00000
分割数 n=100 のとき, 積分値=3.00000
分割数 n=1000 のとき, 積分値=3.00000

積分関数 `return(x*x+2*x+1);`
分割数 n=10 のとき, 積分値=2.3333
分割数 n=100 のとき, 積分値=2.33333
分割数 n=1000 のとき, 積分値=2.33333

(2) 定積分値

8/3

7/3
3
7/3
7/3

第10章 練習問題 7

(1) void __fastcall TForm1::Button2Click(TObject*Sender)

```
{
    //区分求積法による数値積分
    double a,b,h,s,x,y;
    int n,i;

    //積分区間始点 a , 積分区間終点 b , 分割数 n の入力
    a=StrToFloat(Edit1->Text);
    b=StrToFloat(Edit2->Text);
    n=StrToInt(Edit3->Text);

    //分割幅の計算
    h=(b-a)/n;

    s=0;
    for(i=0;i<=n-1;i++){
        x=a+i*h;
        s=s+f(x);
    }
    s=s*h;

    //積分値の表示
    Edit4->Text=FormatFloat("####0.00000",s); //Edit4 に計算結果の表示
    //-----
    //関数 f(x)=x*x の定義 (メンバ関数の定義)
    double __fastcall TForm1::f(double x)
    {
        return(x*x);
    }
}
```

(2) An の計算

```
for(i=0;i<=n-1;i++){
    x=a+i*h;
    s=s+f(x);
}
s=s*h;
```

Bn の計算

```
for(i=1;i<=n;i++){
    x=a+i*h;
    s=s+f(x);
}
s=s*h;
```

	An	Bn
n=10	0.28500	0.38500
n=100	0.32835	0.33835
n=1000	0.33283	0.33383

第10章 練習問題 8

コードの記述例

```
//-----  
#include <vcl.h>  
#pragmahdrstop  
  
#include "Unit1.h"  
//-----  
#include <math.h> //sin(),cos()で使用  
#pragmapackage(smart_init)  
#pragmaresource "*.dfm"  
TForm1 *Form1;  
//-----  
__fastcall TForm1::TForm1(TComponent *Owner)  
    : TForm(Owner)  
{  
}  
//-----  
  
void __fastcall TForm1::Button1Click(TObject *Sender)  
{  
    //原点 x0,y0  
    int x0=PaintBox1->Width/2;  
    int y0=PaintBox1->Height/2;  
  
    //図形描画領域の表示  
    PaintBox1->Canvas->Brush->Color=clBackground;  
    PaintBox1->Canvas->Rectangle(0,0,Width,Height);  
  
    //x 軸  
    PaintBox1->Canvas->MoveTo(0,PaintBox1->Height/2);  
    PaintBox1->Canvas->LineTo(PaintBox1->Width,PaintBox1->Height/2);  
  
    //y 軸  
    PaintBox1->Canvas->MoveTo(PaintBox1->Width/2,0);  
    PaintBox1->Canvas->LineTo(PaintBox1->Width/2,PaintBox1->Height);  
  
    //リサージュ図形の描画  
    double r=100,x,y,t;  
    double c=3.14159/180.0;  
    for(double s=0;s<=360;s+=0.1){  
        t=s*c;  
        x=r*sin(m*t);  
        y=r*cos(n*t);  
        PaintBox1->Canvas->Pixels[x0+x][y0-y]=clRed;  
    }  
}  
//-----
```

```

void__fastcallTForm1::RadioGroup1Click(TObject*Sender)
{
    //mの設定
    switch(RadioGroup1->ItemIndex){
        case0:m=1;break;
        case1:m=2;break;
        case2:m=3;break;
        case3:m=4;break;
        case4:m=5;break;
        case5:m=6;break;
        case6:m=7;break;
        case7:m=8;break;
        case8:m=9;break;
        case9:m=10;break;
    }
}
//-----

void__fastcallTForm1::RadioGroup2Click(TObject*Sender)
{
    //nの設定
    switch(RadioGroup2->ItemIndex){
        case0:n=1;break;
        case1:n=2;break;
        case2:n=3;break;
        case3:n=4;break;
        case4:n=5;break;
        case5:n=6;break;
        case6:n=7;break;
        case7:n=8;break;
        case8:n=9;break;
        case9:n=10;break;
    }
}
//-----

void__fastcallTForm1::FormPaint(TObject*Sender)
{
    //図形描画領域の設定
    PaintBox1->Top=160;
    PaintBox1->Left=30;
    PaintBox1->Width=315;
    PaintBox1->Height=260;
    PaintBox1->Canvas->Brush->Color=clBackground;
    PaintBox1->Canvas->Rectangle(0,0,Width,Height);

    //x軸
    PaintBox1->Canvas->MoveTo(0,PaintBox1->Height/2);
    PaintBox1->Canvas->LineTo(PaintBox1->Width,PaintBox1->Height/2);
}

```

```

//y 軸
PaintBox1->Canvas->MoveTo(PaintBox1->Width/2,0);
PaintBox1->Canvas->LineTo(PaintBox1->Width/2,PaintBox1->Height);

//m,n 初期値設定
RadioGroup1->ItemIndex=0;
RadioGroup2->ItemIndex=0;
}
//-----

```

【ヘッダファイル】

```

//-----
private:    //ユーザ宣言
    int m,n;
public:    //ユーザ宣言
    __fastcall TForm1(TComponent*Owner);
};
//-----
extern PACKAGE TForm1 *Form1;
//-----
#endif

```

第11章 練習問題 1

$$(1) \begin{bmatrix} 3 & 5 \\ 7 & 9 \end{bmatrix}$$

$$(2) \begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix}$$

$$(3) \begin{bmatrix} 8 & 10 & 12 \\ 14 & 10 & 12 \\ 20 & 22 & 24 \end{bmatrix}$$

$$(4) \begin{bmatrix} -2 & -2 & -2 \\ -2 & -2 & -2 \\ -2 & -2 & -2 \end{bmatrix}$$

$$(5) \begin{bmatrix} 15 & 15 & 15 & 15 \\ 15 & 15 & 15 & 15 \\ 14 & 14 & 14 & 14 \\ 14 & 14 & 14 & 14 \end{bmatrix}$$

$$(6) \begin{bmatrix} 5 & 3 & 1 & -1 \\ -3 & -5 & -7 & -9 \\ -12 & -14 & -16 & -18 \\ -20 & -22 & -24 & -26 \end{bmatrix}$$

第11章 練習問題 2

$$(1) \begin{bmatrix} 220 & 140 \\ 390 & 300 \end{bmatrix}$$

$$(2) \begin{bmatrix} 56 & 42 & 29 \\ 47 & 32 & 22 \\ 78 & 62 & 47 \end{bmatrix}$$

$$(3) \begin{bmatrix} 97 & 93 & 120 & 61 \\ 73 & 76 & 89 & 39 \\ 33 & 33 & 32 & 13 \end{bmatrix}$$

第11章 練習問題 3

Y 形の図形データを配列 a に格納する .

```
a[0][0]=0;
a[1][0]=-0.5;
a[0][1]=2.5;
a[1][1]=-0.5;
a[0][2]=4;
a[1][2]=-2;
a[0][3]=4;
a[1][3]=-1;
a[0][4]=3;
a[1][4]=0;
a[0][5]=4;
a[1][5]=1;
a[0][6]=4;
a[1][6]=2;
a[0][7]=2.5;
a[1][7]=0.5;
a[0][8]=0;
a[1][8]=0.5;
```

作図のための繰り返し範囲を修正する .

```
for(int i=1;i<=8;i++){ //回転後の図形の各点の座標を計算
```

第11章 練習問題 4

StringGrid のプロパティの設定例

StringGrid	ColCount	3 (3 列表示)	
	RowCount	6 (6 行表示)	
	DefaultColWidth	60 (セルの幅)	
	DefaultRowHeight	30 (セルの高さ)	
	FixedCols	0 (固定行を消す)	
	FixedRows	0 (固定行を消す)	
	-Options	goEditing	true (入力可とする)
		goAlwaysShowEditor	true (常に編集状態にする)
ScrollBars	ssNone (スクロールバーを表示しない)		

コードの記述例

```
void __fastcall TForm1::Button1Click(TObject* Sender)
{
    // 配列の宣言
    int a[6][3], b[6][3], c[6][3];

    // A店のデータを配列 a に格納
    for(int i=0; i<=5; i++){
        for(int j=0; j<=2; j++){
            a[i][j]=StrToInt(StringGrid1->Cells[j][i]);
        }
    }

    // B店のデータを配列 b に格納
    for(int i=0; i<=5; i++){
        for(int j=0; j<=2; j++){
            b[i][j]=StrToInt(StringGrid2->Cells[j][i]);
        }
    }

    // 合計の計算
    for(int i=0; i<=5; i++){
        for(int j=0; j<=2; j++){
            c[i][j]=a[i][j]+b[i][j];
        }
    }

    // 結果の表示
    for(int i=0; i<=5; i++){
        for(int j=0; j<=2; j++){
            StringGrid3->Cells[j][i]=IntToStr(c[i][j]);
        }
    }
}
//-----
```