

概要

ITSS準拠の研修カリキュラム「プログラミングの基礎（B121）」のうち、「クラス」の実習として実際にクラスとそのインスタンスを生成して挙動を見る実習です。

課題

事前準備

特に準備はありませんが、Visual StudioでC#のWindows Formプロジェクトを作成してください。後述する課題1でクラスを作成しますが、そのインスタンスをNewして、メソッドやプロパティを呼ぶロジックを適当なボタン等に関連付けて書く必要があります。

課題1 銀行口座クラス作成

プロジェクトに新しく「クラス」を追加して、後述の「BankAccount」クラスのコードを張り付けて完成させてください。クラスの機能は大きく下記の4機能です。

- 口座名義をプロパティで指定可能。
- Depositメソッドで入金（戻り値無し）。
- Withdrawメソッドで出金（失敗ならFalseを返す）。
- 金額はAccountBalanceプロパティで確認可能。

口座名義を表すAccountName以外は宣言部分しか記述されていないので、中のロジックを作ってみてください。

作ったクラスは、プログラムのFormに適当なボタンを貼り付け、ボタンが押されると各メソッドが動くような仕掛けを追記して実際に動かしてみましょう（このあたりの具体的な方法が判らなければ講師が教えます）。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ClassSample
{
    class BankAccount
    {
        // 口座の内部情報
        private string accountName = ""; // 口座名義人名
        private int total = 0; // 口座残高

        /// <summary>
        /// 口座名義プロパティ。
        /// </summary>
        public string AccountName
```

```
{
    get
    {
        return accountName;
    }
    set
    {
        accountName = value;
        if (value == null)
        {
            accountName = "";
        }
    }
}

/// <summary>
/// 口座残高設定/取得用プロパティ。
/// </summary>
public int AccountBalance
{
    get
    {
    }
    set
    {
    }
}

/// <summary>
/// 口座への預け入れ
/// </summary>
/// <remarks>
/// 引数で指定した金額を口座に追加する。
/// </remarks>
/// <returns></returns>
public void Deposit(int kingaku)
{
}

/// <summary>
/// 口座引き落とし
/// </summary>
/// <remarks>
/// 引数で指定した金額を口座から引き落とす。
/// </remarks>
/// <param name="kingaku">引き落とし金額。</param>
/// <returns>追加成功ならTrue、失敗ならFalse。</returns>
public Boolean Withdraw(int kingaku)
{
}
```

```
}  
}
```

課題2 口座操作GUI追加

課題1で作ったクラスに入る情報を画面から操作できるようにします。フォームに下記の部品を用意し、金額を指定しての入出金を行えるようにしてください。

- 入出金テキストボックス
- 入金ボタン（入金後の総額をメッセージで表示）
- 出金ボタン（出金後の総額をメッセージで表示）

課題3 残高照会機能追加

課題2の内容を改造し、入出金のたびに表示されるメッセージを取りやめ、金額は残高照会でテキストボックスに表示されるようにします。下記の機能を追加してください。

- 入出金でのメッセージ表示をやめる。
- 残高照会ボタンを追加し、照会ボタンで残高がテキストボックスに表示されるようにする。

課題4 複数人口座管理機能追加

銀行らしく、複数人の口座情報を管理できるようにします。ここまでの課題で作ったクラスの「インスタンス」を新人の人数分用意し、それぞれの名前を口座名義に持つ口座情報を人数分用意してください。ここまですで作ったボタン等を改造し、各口座について入出金、残高照会を行えるようにしてください。

課題5 残高不足対応

引き出し処理で口座残高を超える引き出しが行われそうになった場合、警告メッセージを表示して引き出しを中止する処理を盛り込んでください。

課題6 コンストラクター作成

BankAccountクラスにコンストラクターを作り、新しく作ったBankAccountインスタンスの預金残高が必ず1,000円になるようにしてください。

課題7 引数有りコンストラクターの作成

BankAccountクラスのコンストラクターを改造し、引数「string accountName」を追加、口座名義人の情報をコンストラクターで初期設定しなければならないようにしてください。併せて、BankAccountを使っている機能の方も、BankAccountインスタンス作成時の処理で引数を指定するように変更してください。

課題8 引数の異なるコンストラクターオーバーロードの作成

BankAccountクラスのコンストラクターを増やし、引数をそれぞれ変えて下記の3種類にしてください。

- 引数無し …… 各プロパティの初期値は設定されない。
- 口座名義人を初期設定できる（課題7で作った物と同じ）
- 口座名義人と口座残高の初期値を引数で指定できる（引数を2個にする）