

松山大学 経営学部

情報処理論（応用）



第7回 例外処理



講師 檀 裕也

<http://www.cc.matsuyama-u.ac.jp/~dan/application/>

2006年10月17日

出席確認

- 出席確認フォームから学籍番号および氏名を送信せよ。

出席確認フォーム

<http://www.cc.matsuyama-u.ac.jp/~dan/application/attendance.html>

前回の課題

割引券発行プログラム (ex_20061013)

- 商品の購入金額に応じて割引券を発行する。
 - 1,000円未満の場合...発行しない
 - 1,000円以上10,000円未満の場合...100円分の割引券
 - 10,000円以上の場合...1,000円分の割引券

割引券発行プログラム

商品の購入金額を入力してください。

500 入力

割引券は発行しません。

終了

割引券発行プログラム

商品の購入金額を入力してください。

1500 入力

100円分の割引券を発行します。

終了

割引券発行プログラム

商品の購入金額を入力してください。

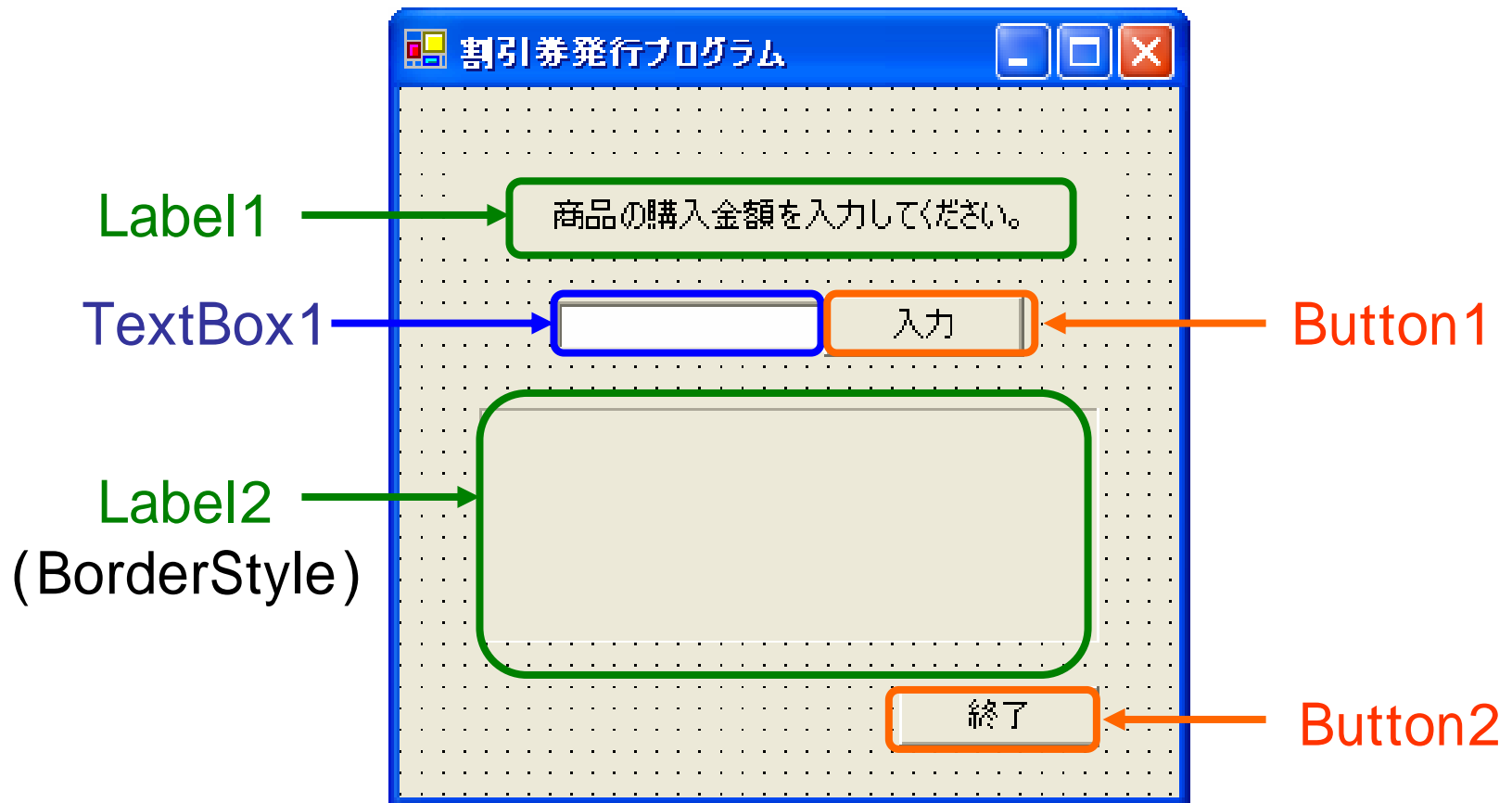
12000 入力

1000円分の割引券を発行します。

終了

解答例 (フォーム)

ラベル・ボタン・テキストボックスを配置する



解答例 (コード)

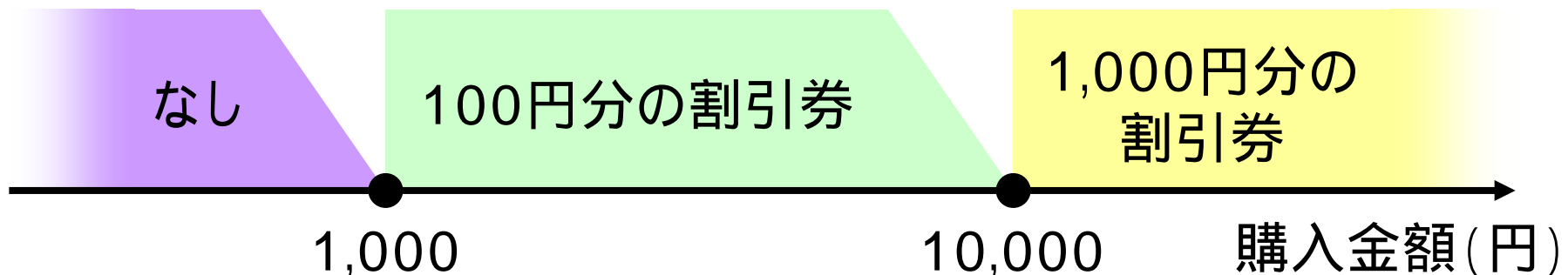
- 購入金額に応じて割引券の発行を処理する

```
Form1 (Declarations)
Public Class Form1
    Inherits System.Windows.Forms.Form

    Windows フォーム デザイナで生成されたコード

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        If TextBox1.Text >= 10000 Then
            Label2.Text = "1000円分の割引券を発行します。"
        ElseIf TextBox1.Text >= 1000 Then
            Label2.Text = "100円分の割引券を発行します。"
        Else
            Label2.Text = "割引券は発行しません。"
        End If
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        Me.Dispose()
    End Sub
End Class
```



今回の予定

- 例外処理
 - プログラム動作中のエラーに対処する
- 到達目標
 - Try ~ Catch ステートメントを使って、プログラム動作中のエラーに対処できる

例外処理

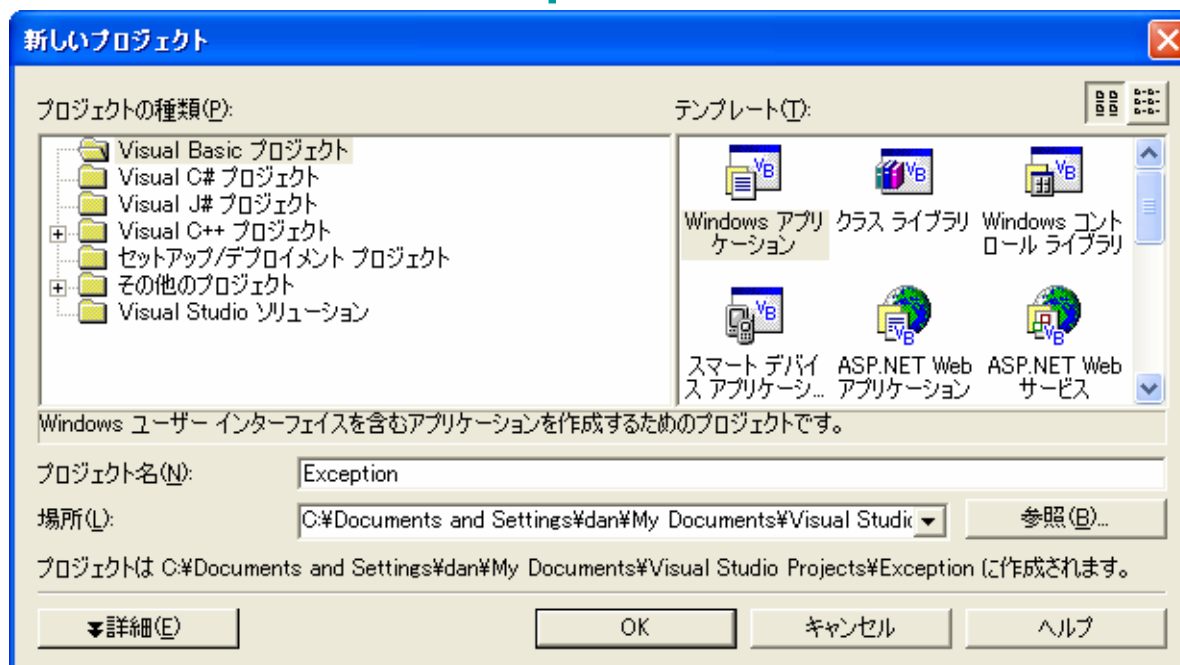
- プログラムの実行中にエラーが発生して処理が停止する事態を防止する
- 「例外」の例
 - 数値入力用のテキストボックスにユーザが数字ではない文字を入力した
 - ファイルを開こうとしたら、そのファイルが存在しなかった
- プログラムおよびデータの安定性のためには例外処理が**絶対不可欠**である

例題 6

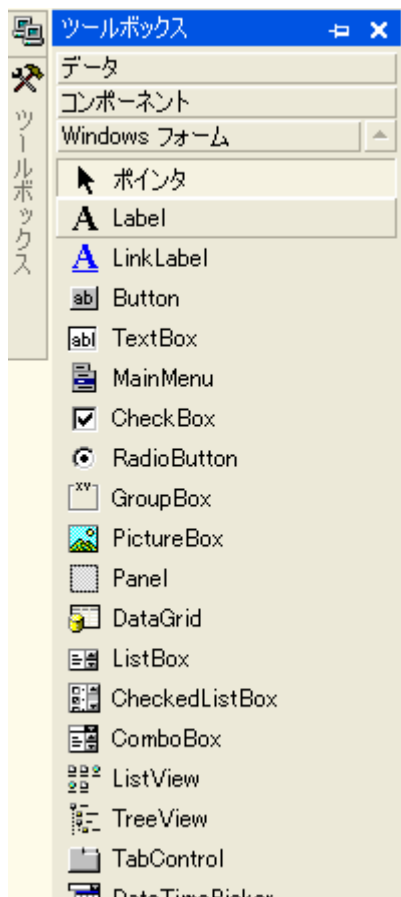
- 入力値が数字ではない場合、エラーを出すことなく数字の入力を促すメッセージを表示する
- 手順：
 - プロジェクトの新規作成 (Exception)
 - 画面レイアウト (ラベルとボタンなどの配置)
 - イベントハンドラの記述
 - 実行および動作確認

新しいプロジェクト

- プロジェクトの種類「Visual Basicプロジェクト」
- テンプレート「Windowsアプリケーション」
- プロジェクト名「Exception」



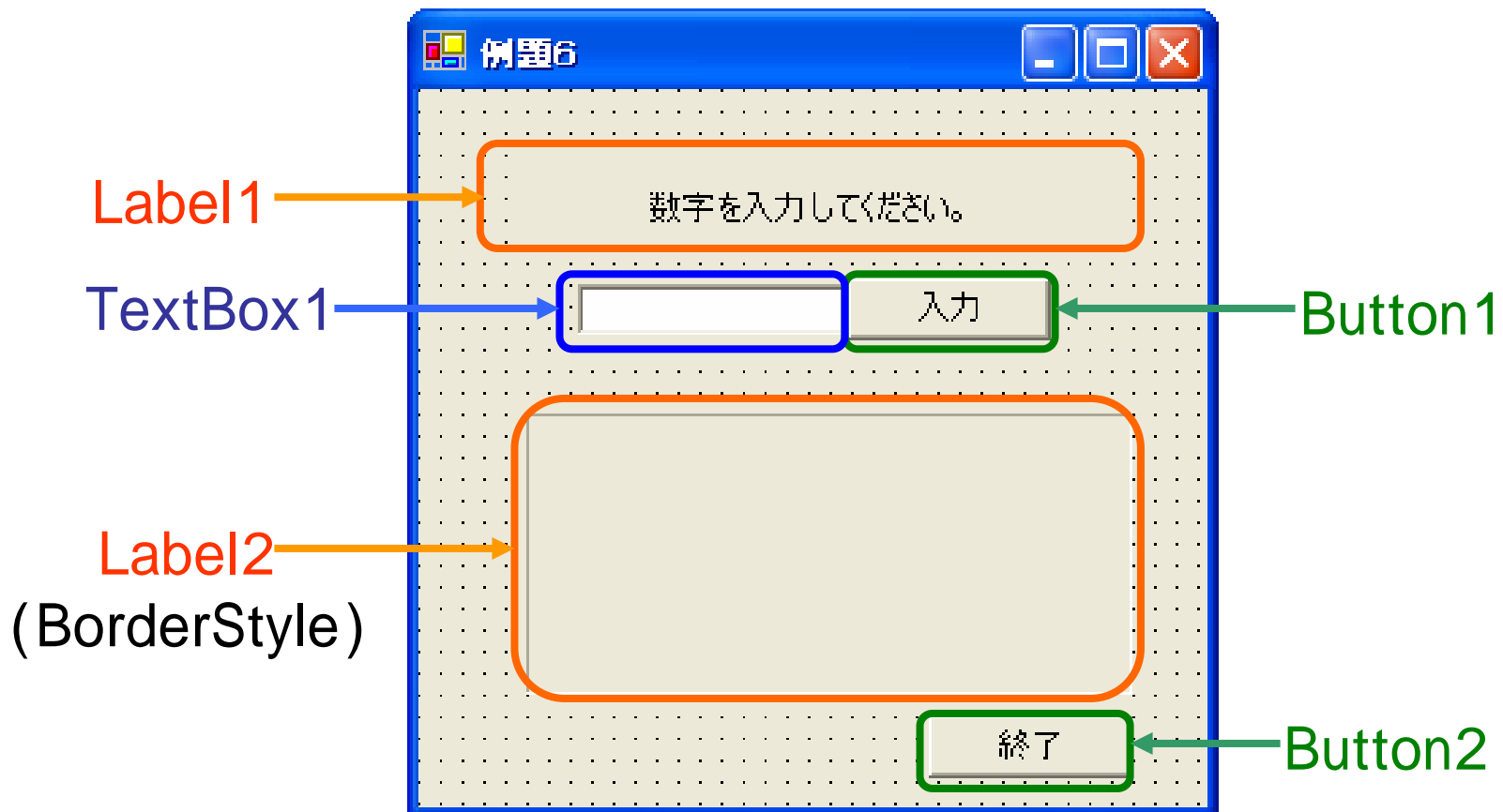
画面の構成要素



- ツールボックスの中から必要なコントロールを選択し、フォームに貼り付ける。
 - コントロールをダブルクリック
 - コントロールをフォーム上にドラッグ & ドロップ

フォームの構成要素

ラベル・ボタン・テキストボックスを配置する



コードの記述

テキストボックスに入力された文字列を Integer 型の整数に変換するとき、例外を捕らえる

```
Public Class Form1
    Inherits System.Windows.Forms.Form

    Windows フォーム デザイナで生成されたコード

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim n As Integer
        Dim str As String

        Try
            n = CInt(TextBox1.Text)
            str = "受け付けた数字は" & n.ToString() & "です。"
        Catch ex As System.Exception
            str = "数字を入力してください。"
        End Try

        Label2.Text = str

    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        Me.Dispose()
    End Sub
End Class
```

変数の宣言

プログラムで使用する変数を宣言する

【構文】 Dim **変数名** As **データ型**

Dim **n** As **Integer** ...変数 **n** を **Integer** 型で宣言する

Dim **str** As **String** ...変数 **str** を **String** 型で宣言する

Integer は整数、**String** は文字列を示す

同じデータ型の複数の変数を同時に宣言できる

Dim **a, b, c** As **Integer**

データ型の変換

明示的（または暗黙的）にデータ型を変換する

- 文字列から整数にデータを変換する

```
n = CInt(TextBox1.Text)
```

TextBox1.Text = "12" のとき n = 12 となる

- 整数から文字列にデータを変換する

```
str = n.ToString()
```

n = 12 のとき str = "12" となる

Try ~ Catch ステートメント

Try

例外の発生が想定されるコード

Catch ex As System.Exception

例外発生時の処理

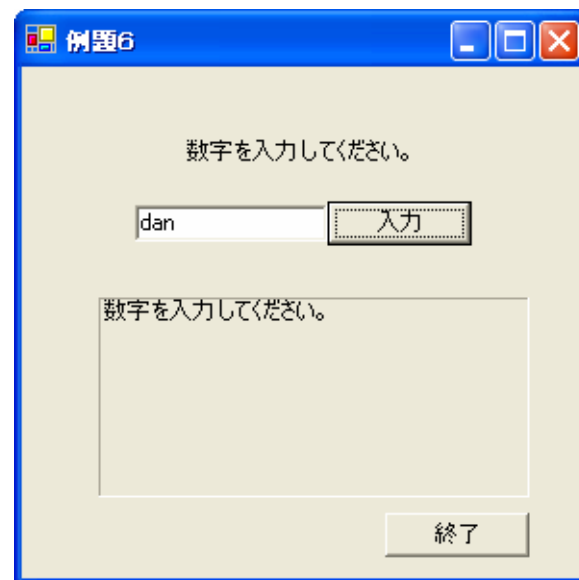
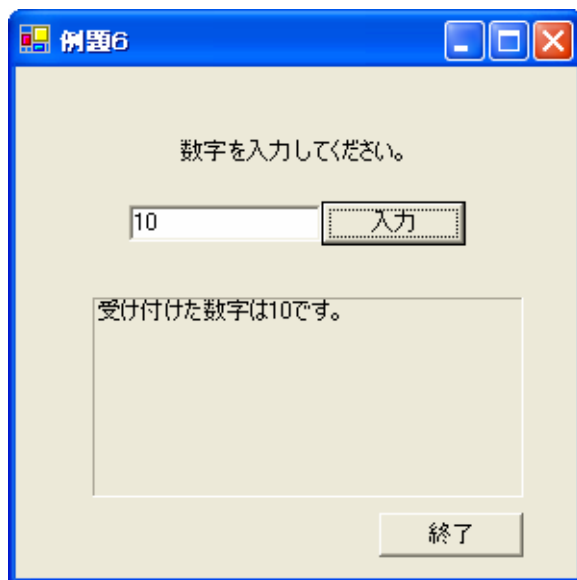
End Try

例えば `TextBox1.Text` に数字でない文字列が入っているとき `n = CInt(TextBox1.Text)` で例外が発生する。

実行 (デバッグ)



- [デバッグ] [開始]でプログラムを実行する
- ツールバーの**開始ボタン**をクリックしてもよい



課題

簡易電卓プログラム (ex_20061017)

- 入力された2つの数字を使って、和・差・積・商を計算する。ただし、例外処理を記述せよ。



まとめ

- Try ~ Catch ステートメントを使い、例外の発生時に対処する例外処理を学んだ。
 - 変数の宣言
 - データ型の変換
 - Try ~ Catch ステートメント

次回予定

- ファイル処理
 - ファイルを開く・保存する