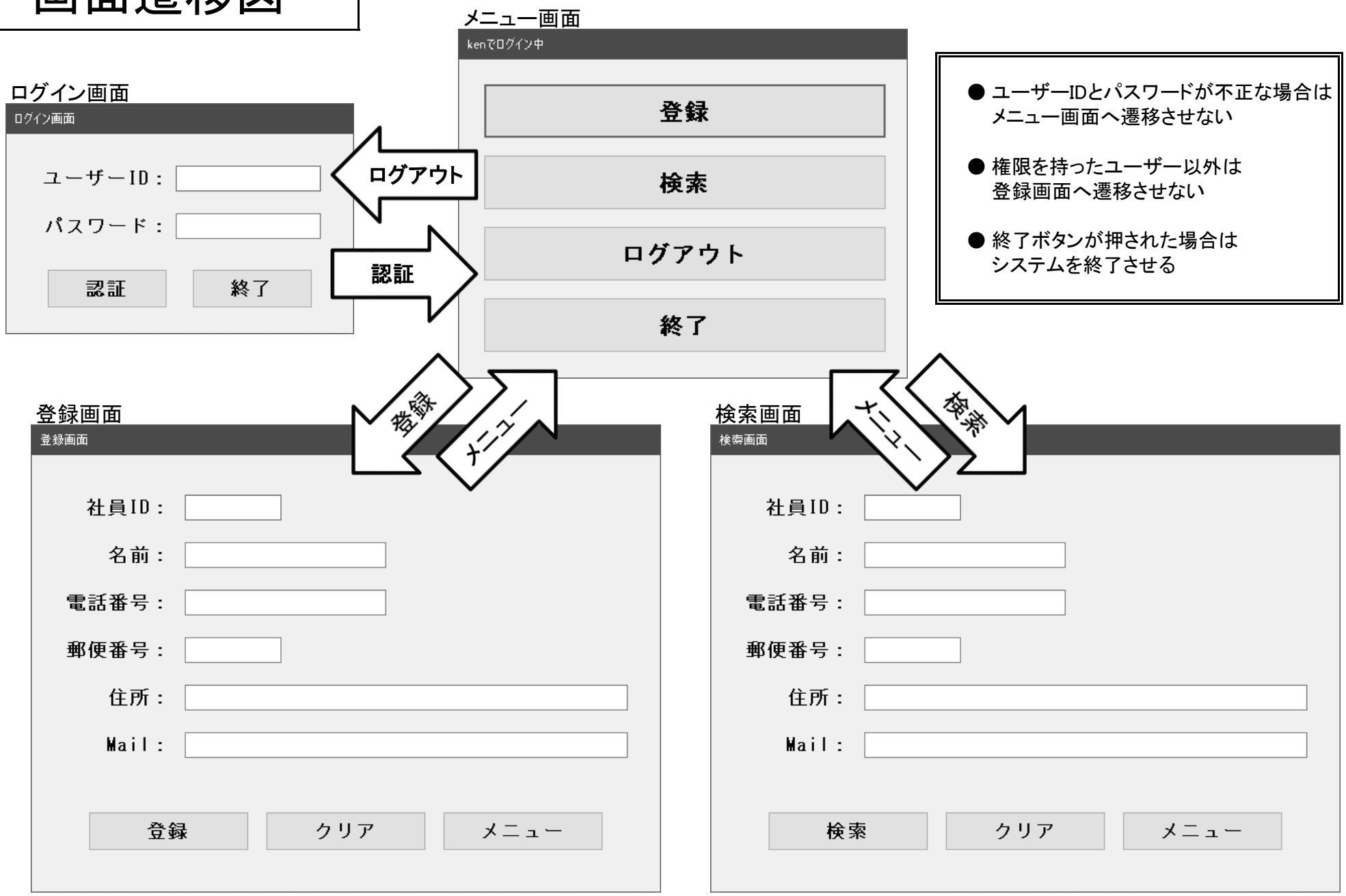
社員管理システム

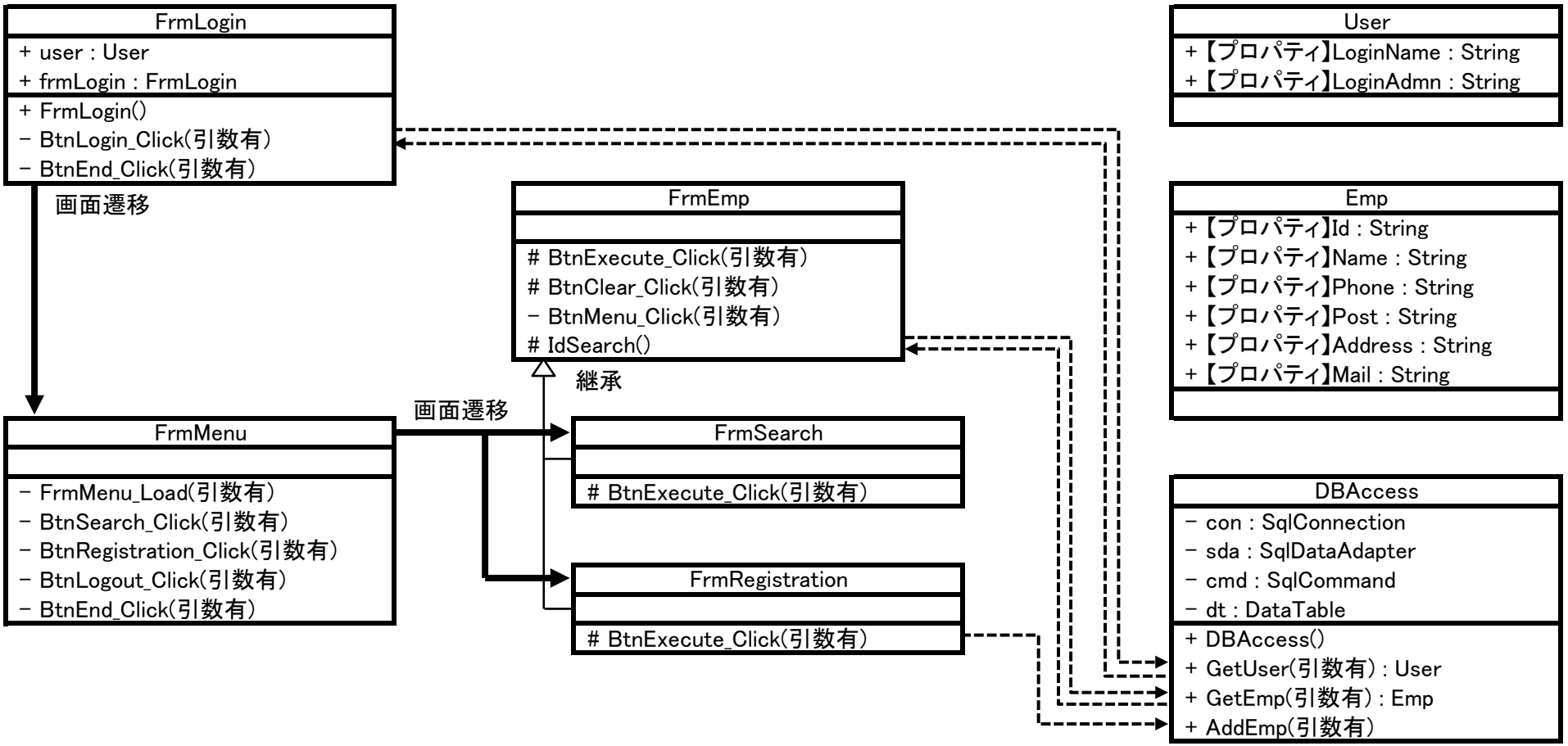
最終更新日	2022/03/04

画面遷移図



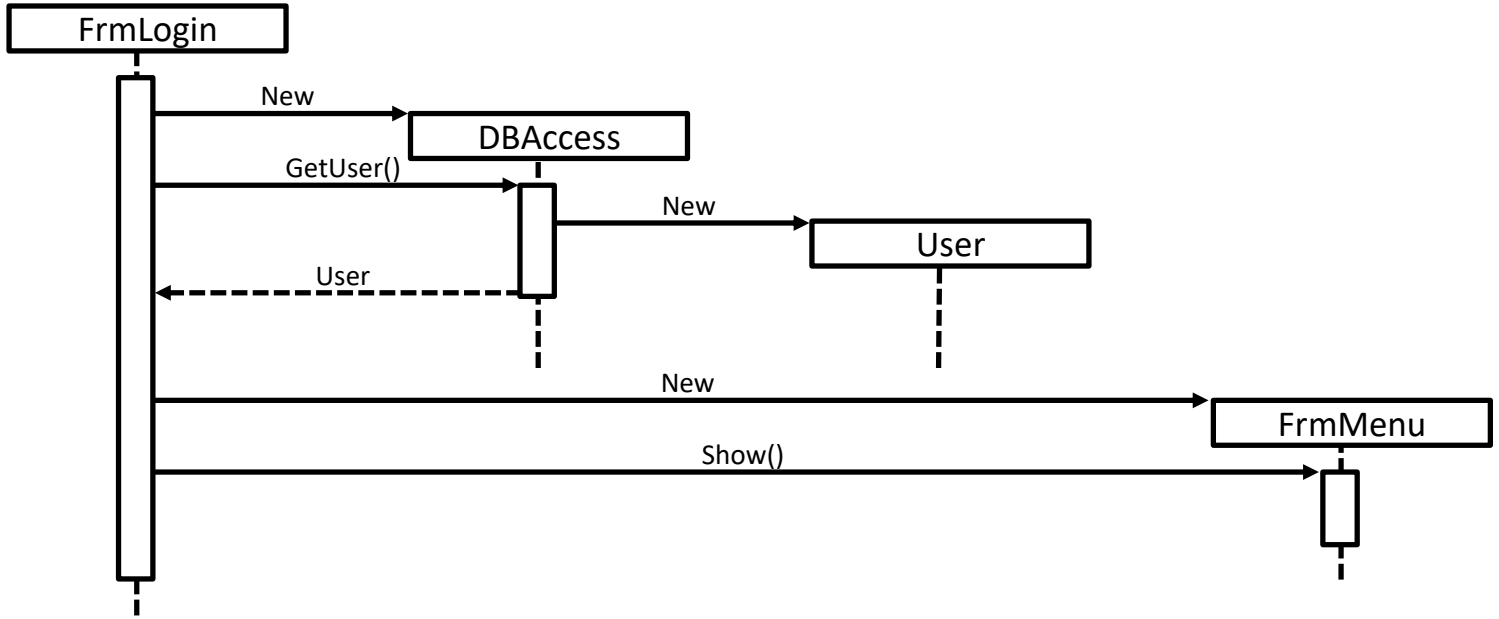
- ユーザーIDとパスワードが不正な場合はメニュー画面へ遷移させない
- 権限を持ったユーザー以外は登録画面へ遷移させない
- 終了ボタンが押された場合はシステムを終了させる

クラス相関図



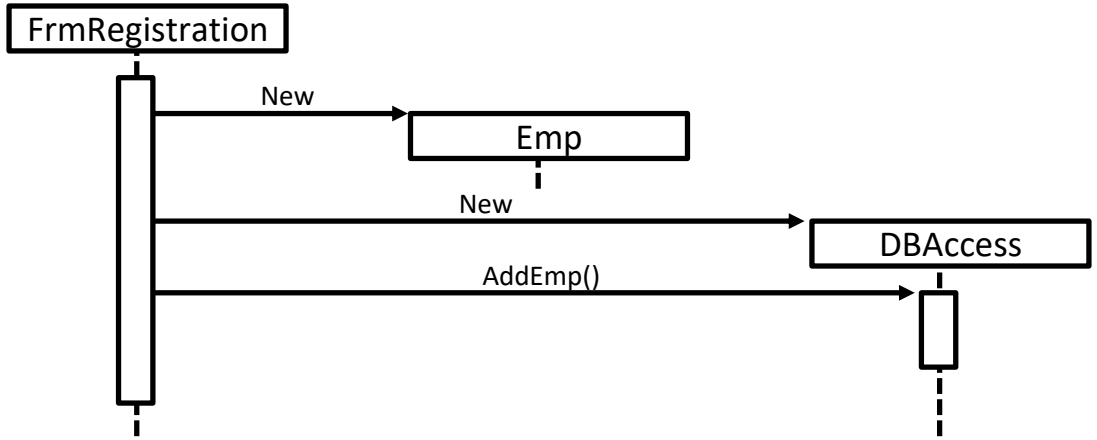
シーケンス図

FrmLogin 認証押下時処理

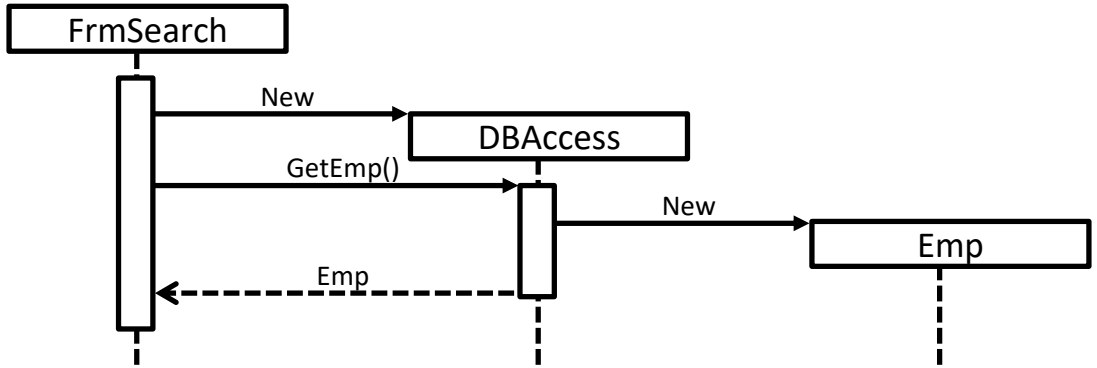


シーケンス図

FrmRegistration 登録押下時処理



FrmSearch 検索押下時処理



社員管理システム

システム名	社員管理システム
処理名	ログイン情報保持
プログラム名	User

処理名	システム共通情報保持	クラス名	User
業務処理概要			

【処理概要】
- 各クラスから呼び出される
- ログイン情報を扱うプロパティを持つ

クラス詳細(クラス設計／変数設計)

【クラス設計】

項番	項目	値
1	アクセス修飾子	public

【変数設計】

項番	可視性	型	変数名	初期値	備考
1	public	String	LoginName		プロパティ ログインユーザー名
2	public	String	LoginAdmn		プロパティ ログイン権限

社員管理システム

システム名	社員管理システム
処理名	社員情報保持
プログラム名	Emp

処理名	システム共通情報保持	クラス名	Emp
業務処理概要			

【処理概要】
- 各クラスから呼び出される
- 社員情報を扱うプロパティを持つ

クラス詳細(クラス設計／変数設計)

【クラス設計】

項番	項目	値
1	アクセス修飾子	public

【変数設計】

項番	可視性	型	変数名	初期値	備考
1	public	String	Id		社員情報プロパティ ID
2	public	String	Name		社員情報プロパティ ユーザー名
3	public	String	Phone		社員情報プロパティ 電話番号
4	public	String	Post		社員情報プロパティ 郵便番号
5	public	String	Address		社員情報プロパティ 住所
6	public	String	Mail		社員情報プロパティ メールアドレス

社員管理システム

システム名	社員管理システム
処理名	データベース処理
プログラム名	DBAccess

処理名	データベース処理	クラス名	DBAccess
業務処理概要			

【処理概要】
- データベースに接続する
- 引数で渡されたユーザーIDを元に、loginuserテーブルを参照し、ユーザー名と権限情報を抽出する
- 引数で渡された社員IDを元に、employeeテーブルから社員情報を抽出する
- 引数で渡された社員情報をemployeeテーブルに登録する
- 検索時、既に存在しない社員IDを指定した場合は、エラーメッセージを出力する
- 登録時、既に存在する社員IDを指定した場合は、エラーメッセージを出力する

クラス詳細(クラス設計／変数設計)

【クラス設計】

項番	項目	値
1	アクセス修飾子	public
2	使用名前空間	System.Configuration
3	使用名前空間	System.Data
4	使用名前空間	System.Data.SqlClient

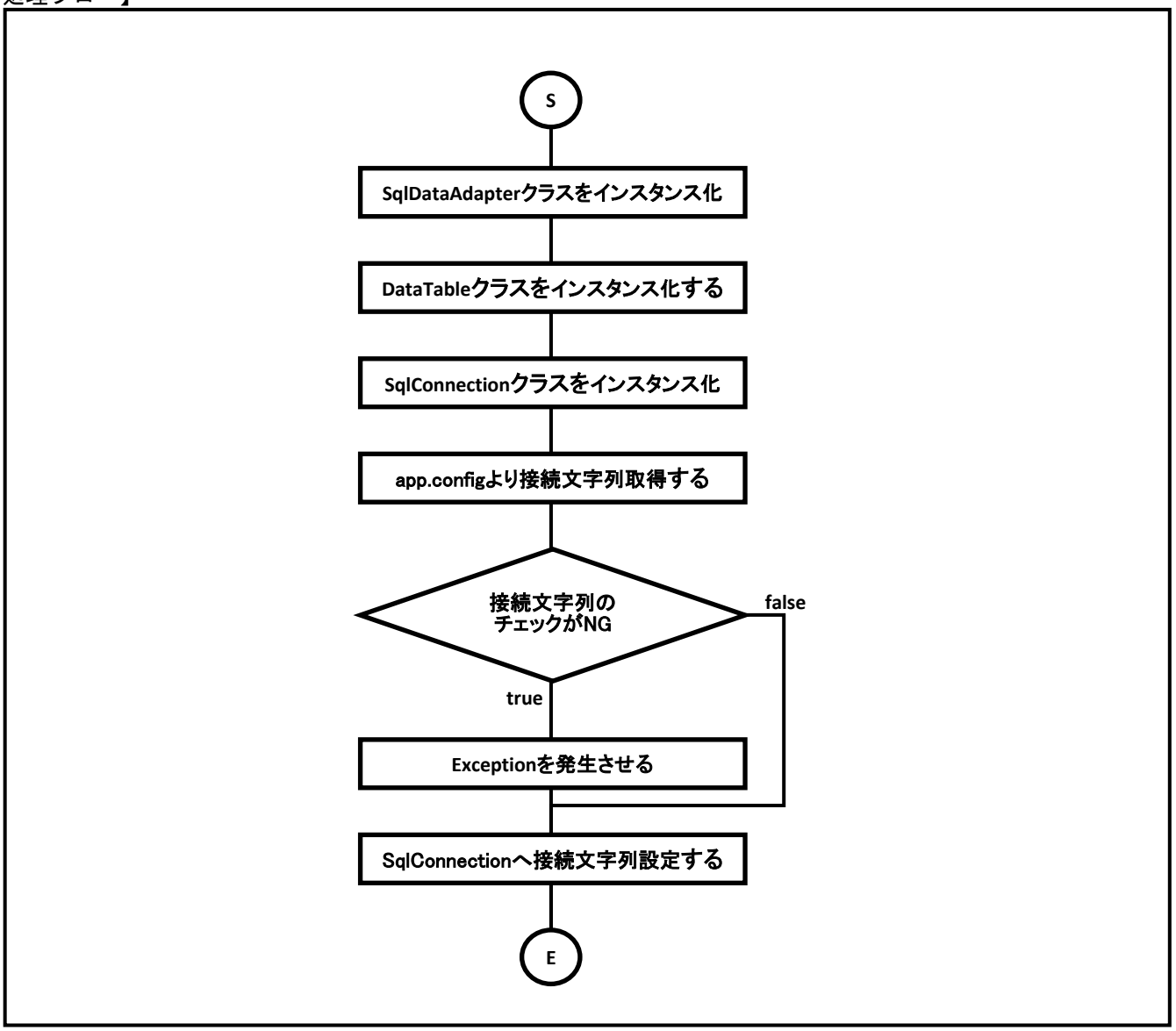
【変数設計】

項番	可視性	型	変数名	初期値	備考
1	private	SqlConnection	con		
2	private	SqlDataAdapter	sda		
3	private	SqlCommand	cmd		
4	private	DataTable	dt		

コンストラクタ詳細

コンストラクタ名	DBAccess

【処理フロー】



【処理内容】

1 SqlConnectionクラスをインスタンス化し、変数conへ格納する
2 SqlDataAdapterクラスをインスタンス化し、変数sdaへ格納する
3 SqlCommandクラスをインスタンス化し、変数cmdへ格納する
4 DataTableクラスをインスタンス化し、変数dtへ格納する
5 ConfigurationManagerクラスのConnectionStringsプロパティを利用して
App.Configより接続文字列を取得し、変数settingへ格納する
6 変数settingがnullであった場合
6.1 「データベースへの接続文字列が正しく設定されていません」というメッセージを持った
Exceptionオブジェクトを生成し、例外を発生させる
7 変数settingがnullでなかった場合
7.1 変数settingのConnectionStringプロパティを、変数conのConnectionStringプロパティへ代入する
7.2 変数conを変数cmdのConnectionプロパティへ代入する

メソッド詳細

メソッド名	GetUser

項番	項目	値	備考
1	アクセス修飾子	public	
2	引数	型：String　名：strId	ログインユーザーID
3	引数	型：String　名：strPw	ログインユーザーパスワード
4	戻り値	User	

【処理内容】

1 変数strIdが空文字　または　変数strPwが空文字　または　変数strIdが数値でなかった場合
　1.1 「IDおよび名前を正しく入力してください」というメッセージを持った
　　Exceptionオブジェクトを生成し、例外を発生させる
2 以下の処理はtry句内に記述する
　2.1 変数conのOpen()メソッドを呼び出す
　2.2 以下の文字列を変数cmdのCommandTextプロパティへ格納する
　　"SELECT * FROM loginuser WHERE id = @key"
　2.3 変数cmdのParametersプロパティのClear()メソッドを呼び出す
　2.4 パラメータクエリの"@key"に対して、変数strIdの値をセットする
　2.5 変数cmdを変数sdaのSelectCommandプロパティへ格納する
　2.6 変数sdaのFill()メソッドを呼び出す。その際の引数は次のとおり
　　第一引数：変数dt
　2.7 変数dtのRowsプロパティのCountプロパティが1の場合
　　2.7.1 Userクラスのインスタンスを生成し、変数userへ格納する
　　2.7.2 変数dtのRowsプロパティの0番目の"password"が変数strPwと一致している場合
　　　2.7.2.1 変数dtのRowsプロパティの0番目の"name"を変数userのLoginNameプロパティへ格納する
　　　2.7.2.2 変数dtのRowsプロパティの0番目の"admin"を変数userのLoginAdmnプロパティへ格納する
　　　2.7.2.3 変数userを戻り値として返す
3 以下の処理はfinally句内に記述する
　3.1 変数conのClose()メソッドを呼び出す
　3.2 変数conのDispose()メソッドを呼び出す
　3.3 変数sdaのDispose()メソッドを呼び出す
　3.4 変数cmdのDispose()メソッドを呼び出す
　3.5 変数dtのDispose()メソッドを呼び出す
4 「該当するユーザーが存在しません」というメッセージを持った
　Exceptionオブジェクトを生成し、例外を発生させる

メソッド詳細

メソッド名	GetEmp

項番	項目	値	備考
1	アクセス修飾子	public	
2	引数	型：String　名：strId	社員ID
3	戻り値	Emp	

【処理内容】

1 変数strIdが空文字　または　変数strIdが数値でなかった場合
　1.1 「IDを正しく入力してください」というメッセージを持った
　　Exceptionオブジェクトを生成し、例外を発生させる
2 以下の処理はtry句内に記述する
　2.1 変数conのOpen()メソッドを呼び出す
　2.2 以下の文字列を変数cmdのCommandTextプロパティへ格納する
　　"SELECT * FROM employee WHERE id = @key"
　2.3 変数cmdのParametersプロパティのClear()メソッドを呼び出す
　2.4 パラメータクエリの"@key"に対して、変数strIdの値をセットする
　2.5 変数cmdを変数sdaのSelectCommandプロパティへ格納する
　2.6 変数sdaのFill()メソッドを呼び出す。その際の引数は次のとおり
　　第一引数：変数dt
　2.7 変数dtのRowsプロパティのCountプロパティが1の場合
　2.7.1 Empクラスのインスタンスを生成し、変数empへ格納する
　2.7.2 変数dtのRowsプロパティの0番目の"id"を変数empのIdプロパティへ格納する
　2.7.3 変数dtのRowsプロパティの1番目の"name"を変数empのNameプロパティへ格納する
　2.7.4 変数dtのRowsプロパティの2番目の"phone"を変数empのPhoneプロパティへ格納する
　2.7.5 変数dtのRowsプロパティの3番目の"post"を変数empのPostプロパティへ格納する
　2.7.6 変数dtのRowsプロパティの4番目の"address"を変数empのAddressプロパティへ格納する
　2.7.7 変数dtのRowsプロパティの5番目の"mail"を変数empのMailプロパティへ格納する
　2.7.8 変数empを戻り値として返す
3 以下の処理はfinally句内に記述する
　3.1 変数conのClose()メソッドを呼び出す
　3.2 変数conのDispose()メソッドを呼び出す
　3.3 変数sdaのDispose()メソッドを呼び出す
　3.4 変数cmdのDispose()メソッドを呼び出す
　3.5 変数dtのDispose()メソッドを呼び出す
4 nullを戻り値として返す

メソッド詳細

メソッド名	AddEmp

項番	項目	値	備考
1	アクセス修飾子	public	
2	引数	型：Emp　名：emp	登録する社員情報
3	戻り値	なし	

【処理内容】

1 変数empのIdプロパティが空文字　または　変数empのNameプロパティが空文字　または
変数empのIdプロパティが数値でなかった場合
1.1 「IDおよび名前を正しく入力してください」というメッセージを持った
Exceptionオブジェクトを生成し、例外を発生させる
2 以下の処理はtry句内に記述する
2.1 変数conのOpen()メソッドを呼び出す
2.2 以下の文字列を変数cmdのCommandTextプロパティへ格納する
"SELECT count(*) FROM employee WHERE id = @key"
2.3 変数cmdのParametersプロパティのClear()メソッドを呼び出す
2.4 パラメータクエリの"@key"に対して、変数empのIdプロパティの値をセットする
2.5 変数cmdのExecuteScalar()メソッドを呼び出し、結果を変数resultScalarへ格納する
2.6 変数resultScalarが0ではない場合
2.6.1 「このIDはすでに登録済みです」というメッセージを持った
Exceptionオブジェクトを生成し、例外を発生させる
2.7 以下の文字列を変数cmdのCommandTextプロパティへ格納する
"INSERT INTO employee (id, name, phone, post, address, mail)
VALUES (@id, @name, @phone, @post, @address, @mail)"
2.8 変数cmdのParametersプロパティのClear()メソッドを呼び出す
2.9 パラメータクエリの"@id"に対して、変数empのIdプロパティの値をセットする
2.10 パラメータクエリの"@name"に対して、変数empのNameプロパティの値をセットする
2.11 パラメータクエリの"@phone"に対して、変数empのPhoneプロパティの値をセットする
2.12 パラメータクエリの"@post"に対して、変数empのPostプロパティの値をセットする
2.13 パラメータクエリの"@address"に対して、変数empのAddressプロパティの値をセットする
2.14 パラメータクエリの"@mail"に対して、変数empのMailプロパティの値をセットする
2.15 変数cmdのExecuteNonQuery()メソッドを呼び出し、その結果が1でない場合
2.15.1 「登録に失敗しました」というメッセージを持った
Exceptionオブジェクトを生成し、例外を発生させる
3 以下の処理はfinally句内に記述する
3.1 変数conのClose()メソッドを呼び出す
3.2 変数conのDispose()メソッドを呼び出す
3.3 変数sdaのDispose()メソッドを呼び出す
3.4 変数cmdのDispose()メソッドを呼び出す
3.5 変数dtのDispose()メソッドを呼び出す

社員管理システム

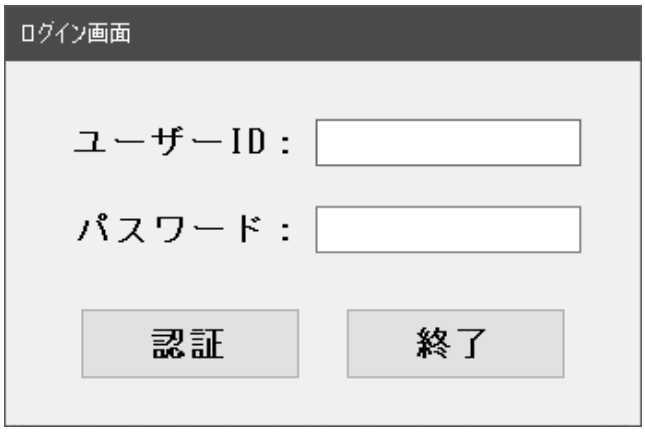
システム名	社員管理システム
処理名	認証画面
プログラム名	FrmLogin

処理名	認証画面	クラス名	FrmLogin
業務処理概要			

【処理概要】

- 社員管理システムにログインするための画面である
- ユーザーIDまたはパスワードが未入力の場合はエラーとする
- 入力されたユーザーIDを元にユーザー情報を取得し、入力されたパスワードを比較する
  一致していた場合はメニュー画面へ遷移する
  一致していなかった場合はエラーメッセージを表示し、再入力を促す
- 終了ボタンが押下された場合は、システムを終了させる

画面レイアウト



画面入力項目

項番	項目名	内容	特記事項
1	ユーザーID	ユーザーIDを入力する	
2	パスワード	パスワードを入力する	画面表示は伏字とする

画面操作項目

項番	項目名	内容	特記事項
1	認証	入力情報で認証を行い、メニュー画面へ遷移する	
2	終了	システムを終了する	

クラス詳細(フォーム設計(クラス設計)／変数設計)

【フォーム設計】

項番	設定項目	値
1	ファイル名	FrmLogin.cs
2	Name	FrmLogin
3	ControlBox	False
4	Size	360, 240
5	StartPostion	CenterScreen
6	Text	ログイン画面

【コントロール設計】

項番	コントロール名	設定項目	値
1	LblId	Font	MSゴシック,14.25pt,style=Bold
		Location	24, 24
		Size	144, 40
		Text	ユーザーID：
		TextArigin	MiddleRight
2	LblPass	Font	MSゴシック,14.25pt,style=Bold
		Location	24, 72
		Size	144, 40
		Text	パスワード：
		TextArigin	MiddleRight
3	TxtId	Font	MSゴシック,14.25pt
		Location	168, 32
		MaxLength	4
		Size	144, 26
		TabIndex	0
4	TxtPass	Font	MSゴシック,14.25pt
		Location	168, 80
		MaxLength	50
		Size	144, 26
		TabIndex	1
		UseSystemPasswordChar	True
5	BtnLogin	Font	MSゴシック,14.25pt,style=Bold
		Location	40, 136
		Size	120, 40
		TabIndex	2
		Text	認証
6	BtnEnd	Font	MSゴシック,14.25pt,style=Bold
		Location	184, 136
		Size	120, 40
		TabIndex	3
		Text	終了

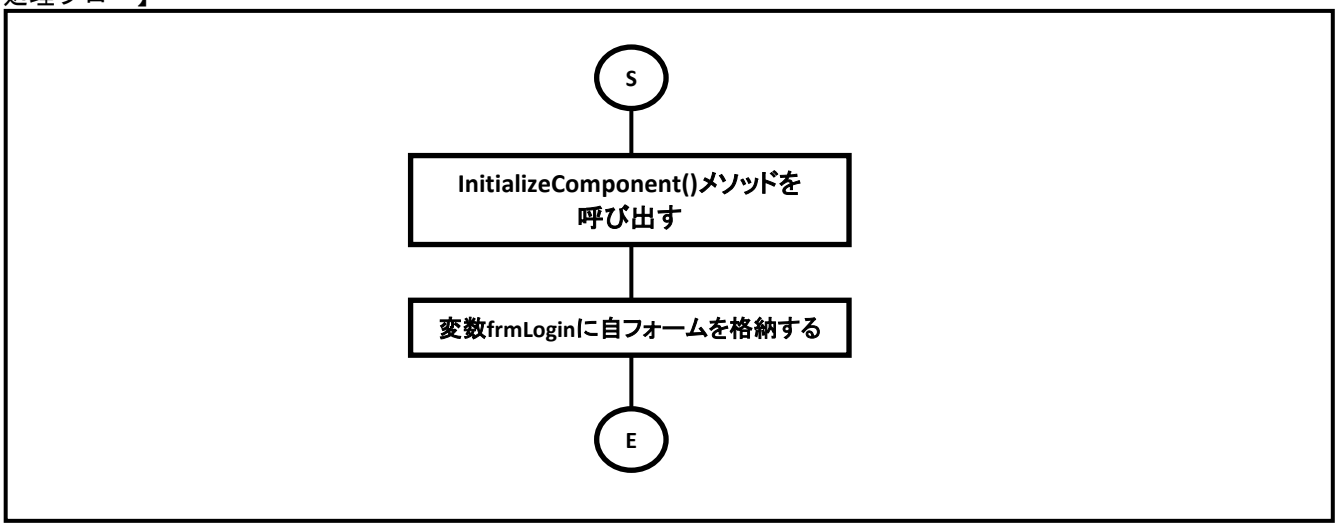
【変数設計】

項番	可視性	型	変数名	初期値	備考
1	public	User	user		静的変数とする
2	public	FrmLogin	frmLogin		静的変数とする

コンストラクタ詳細

コンストラクタ名	FrmLogin

【処理フロー】



【処理内容】

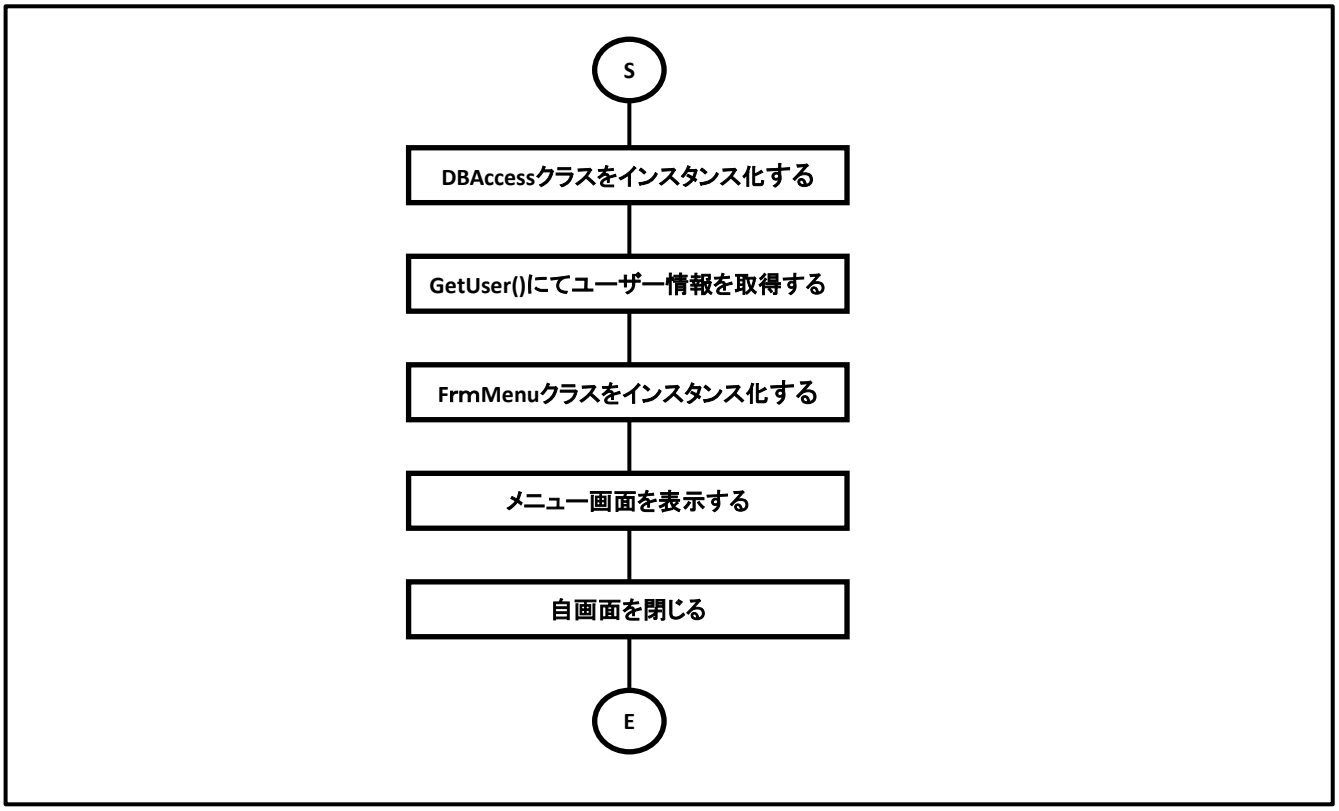
1 InitializeComponent()メソッドを呼び出す（デフォルトで記述済み）
2 変数frmLoginに自フォームを格納する

メソッド詳細

メソッド名	BtnLogin_Click

● 認証ボタン押下

【処理フロー】



【処理内容】

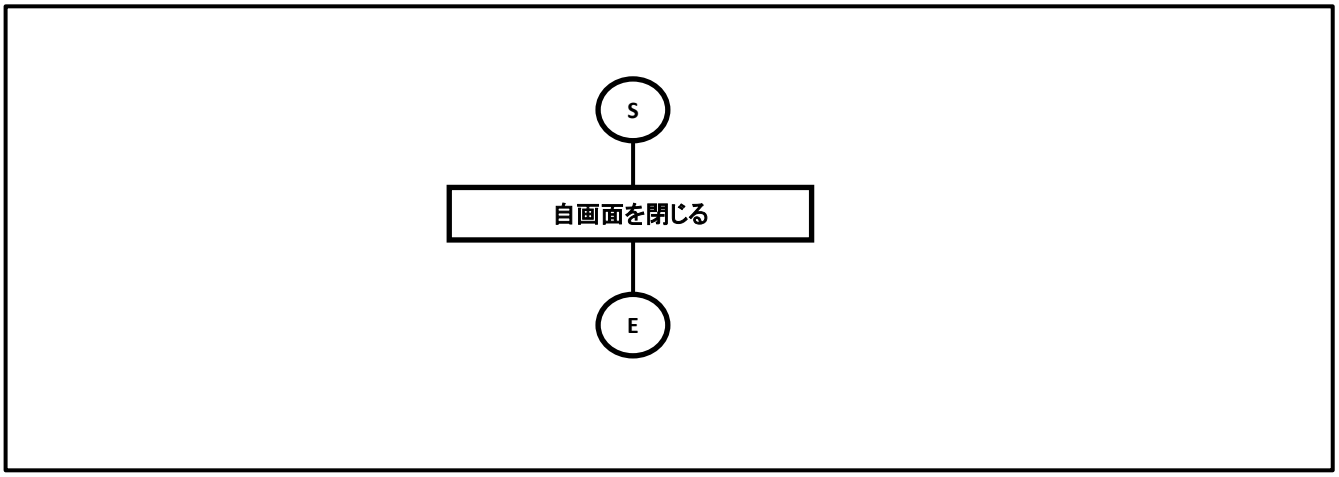
1 以下の処理はtry句内に記述する
 1.1 DBAccessクラスのインスタンスを生成し、変数dbへ格納する
 1.2 変数dbのGetUser()メソッドを呼び出し、結果をメンバ変数userへ格納する。その際の引数は次のとおり
  第一引数：ユーザーID
  第二引数：パスワード
 1.3 FrmMenuクラスをインスタンス化する
 1.4 メニュー画面を表示する
 1.5 自画面を非表示にする
 1.6 ユーザーIDのテキストボックスをクリアする
 1.7 パスワードのテキストボックスをクリアする
2 以下の処理はExceptionクラスを対象としたcatch句内に記述する
 2.1 捕捉した例外オブジェクト内のメッセージをメッセージボックスで表示させる

メソッド詳細

メソッド名	BtnEnd_Click

● 終了ボタン押下

【処理フロー】



【処理内容】

1 自画面を閉じる

社員管理システム

システム名	社員管理システム
処理名	メニュー画面
プログラム名	FrmMenu

処理名	メニュー画面	クラス名	FrmMenu
業務処理概要			

【処理概要】

- 社員管理システムのメニュー画面である
- 登録ボタンが押下された場合は、登録画面へ遷移する
  ただし、権限を持たないユーザーはエラーメッセージを表示させ、画面の遷移は行わない
- 検索ボタンが押下された場合は、検索画面へ遷移する
- ログアウトボタンが押下された場合は、ログイン画面へ遷移する
- 終了ボタンが押下された場合は、社員管理システムを終了させる

画面レイアウト

画面入力項目

項番	項目名	内容	特記事項

※なし

画面操作項目

項番	項目名	内容	特記事項
1	登録	登録画面へ遷移する	
2	検索	検索画面へ遷移する	
3	ログアウト	ログイン画面へ遷移する	
4	終了	システムを終了する	

クラス詳細(フォーム設計(クラス設計)／変数設計)

【フォーム設計】

項番	設定項目	値
1	ファイル名	FrmMenu.cs
2	Name	FrmMenu
3	ControlBox	False
4	Size	460, 360
5	StartPostion	CenterScreen
6	Text	メニュー画面

【コントロール設計】

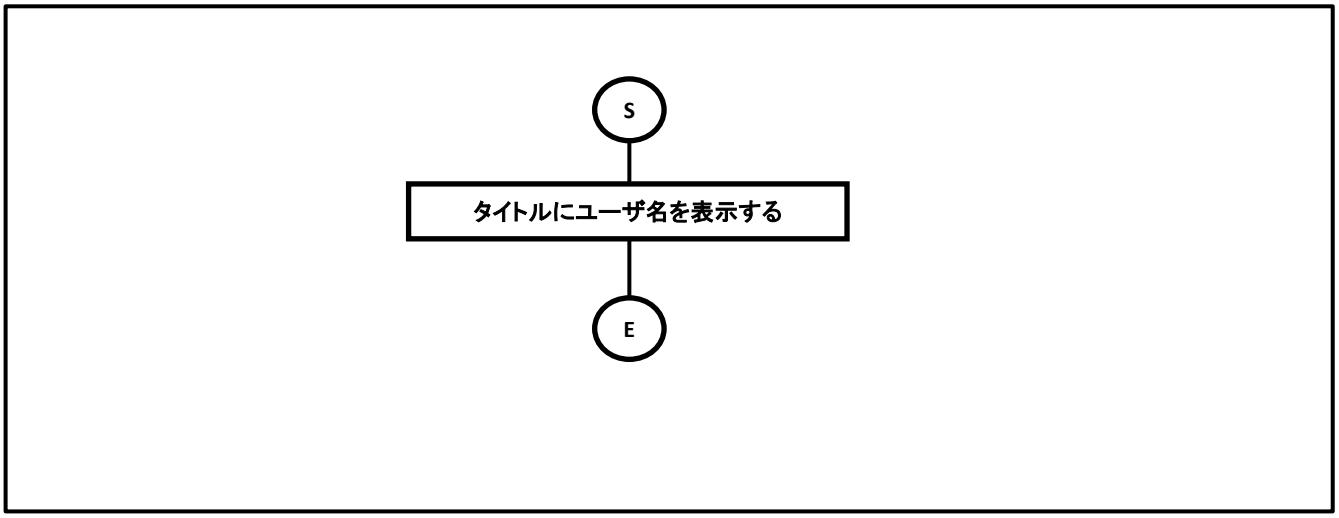
項番	コントロール名	設定項目	値
1	BtnRegistration	Font	MSゴシック, 18.0pt, style=Bold
		Location	24, 24
		Size	400, 56
		TabIndex	0
		Text	登録
2	BtnSearch	Font	MSゴシック, 18.0pt, style=Bold
		Location	24, 96
		Size	400, 56
		TabIndex	1
		Text	検索
3	BtnLogout	Font	MSゴシック, 18.0pt, style=Bold
		Location	24, 168
		Size	400, 56
		TabIndex	2
		Text	ログアウト
4	BtnEnd	Font	MSゴシック, 18.0pt, style=Bold
		Location	24, 240
		Size	400, 56
		TabIndex	3
		Text	終了

メソッド詳細

メソッド名	FrmMenu_Load

● 自画面表示

【処理フロー】



【処理内容】

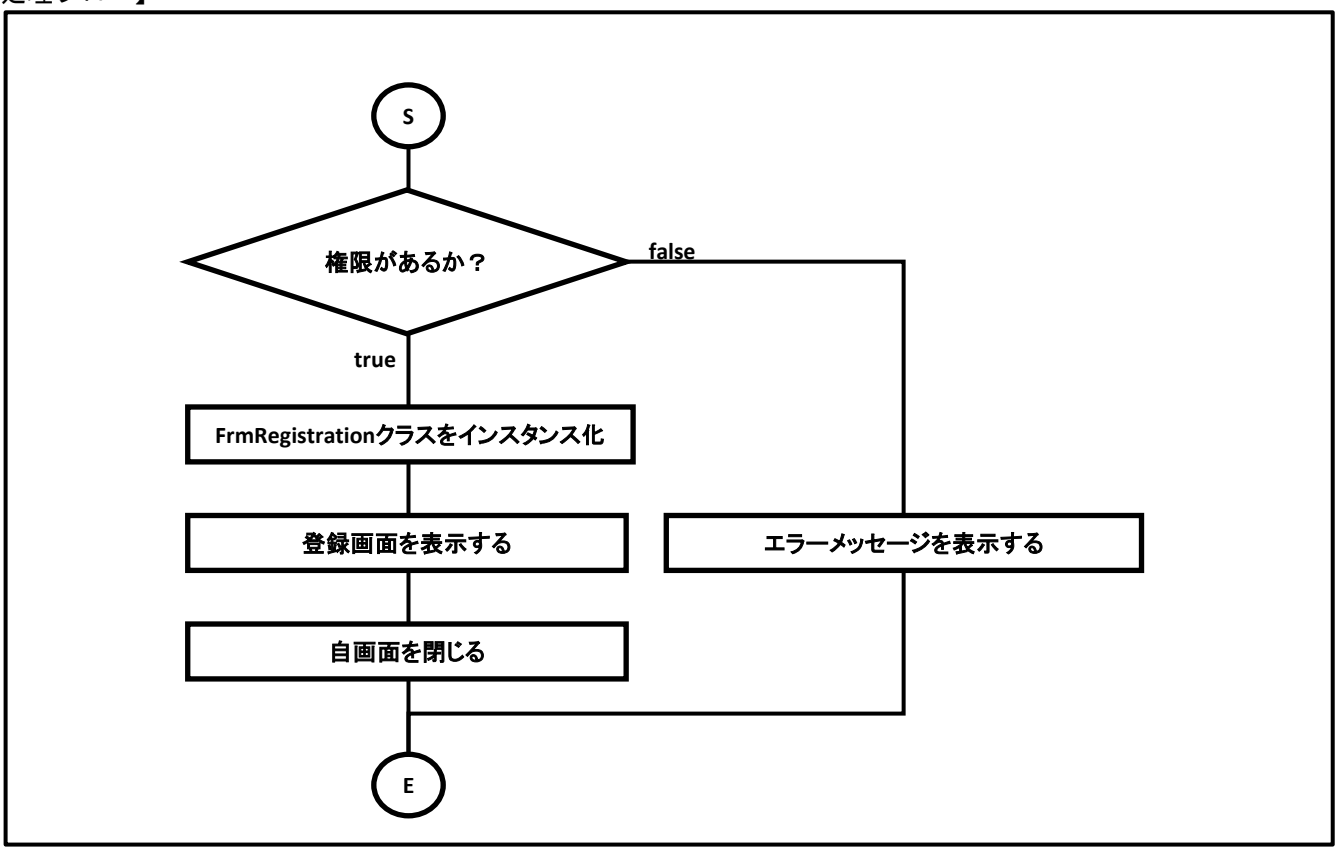
1 FrmLoginクラスの変数userのLoginNameプロパティを利用して
″【ユーザー名】でログイン中″とタイトルに表示する

メソッド詳細

メソッド名	BtnRegistration_Click

● 登録ボタン押下

【処理フロー】



【処理内容】

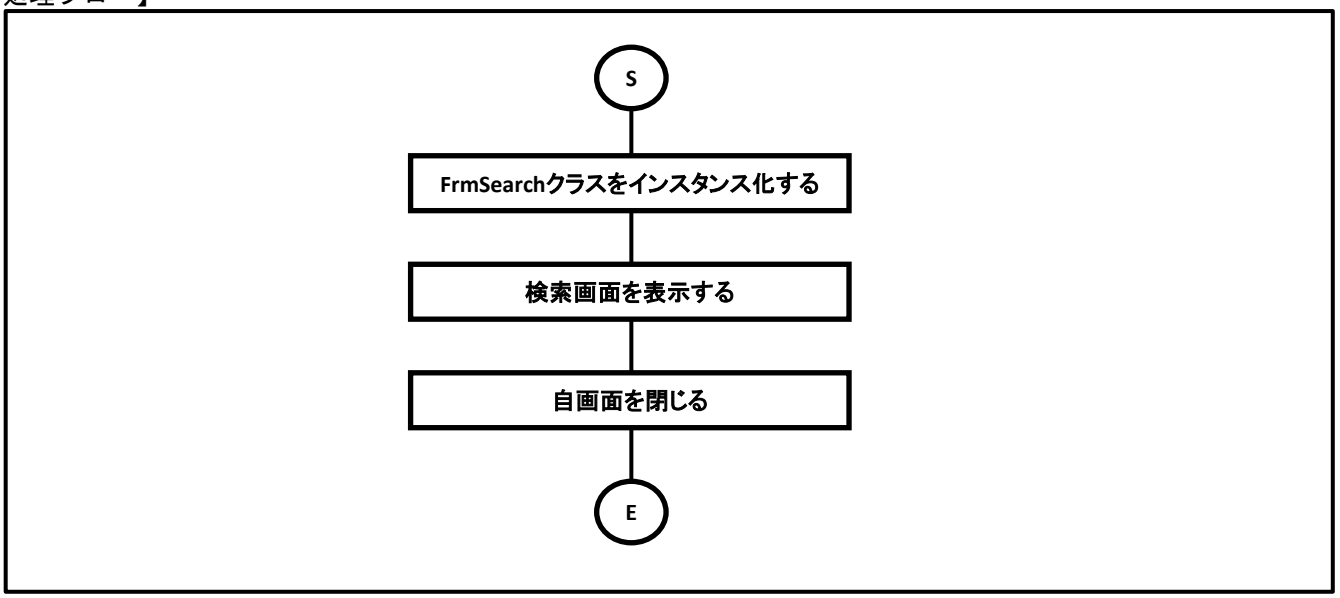
1 FrmLoginクラスの変数userのLoginAdmnプロパティが"1"の場合
 1.1 FrmRegistrationクラスをインスタンス化する
 1.2 登録画面を表示する
 1.3 自画面を閉じる
2 FrmLoginクラスの変数userのLoginAdmnプロパティが"1"ではない場合
 「権限がありません」とメッセージを表示する

メソッド詳細

メソッド名	BtnSearch_Click

● 検索ボタン押下

【処理フロー】



【処理内容】

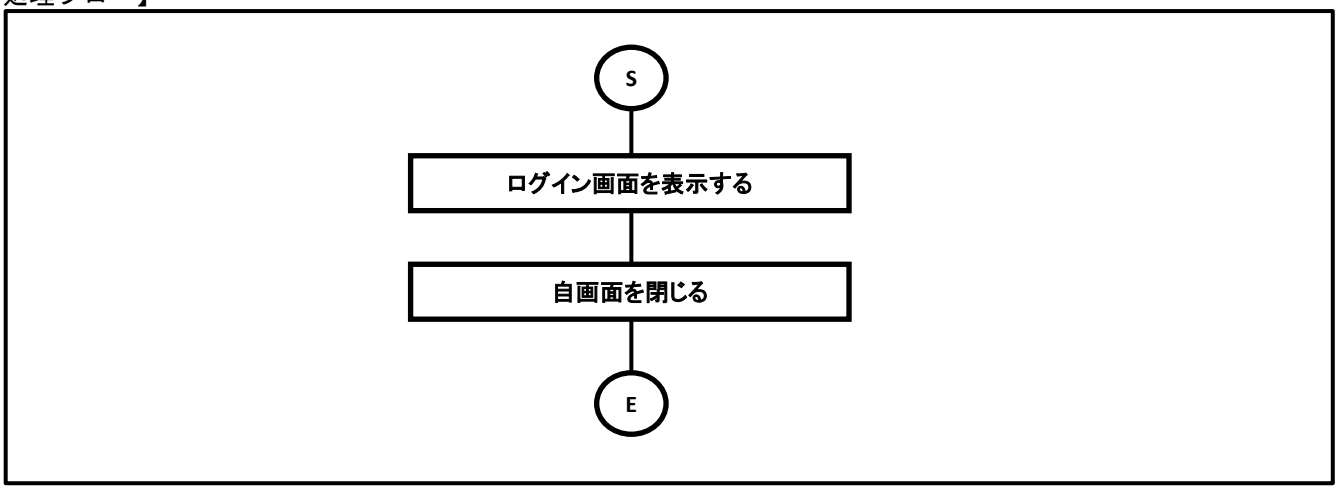
1 FrmSearchクラスをインスタンス化する
2 検索画面を表示する
3 自画面を閉じる

メソッド詳細

メソッド名	BtnLogout_Click

● ログアウトボタン押下

【処理フロー】



【処理内容】

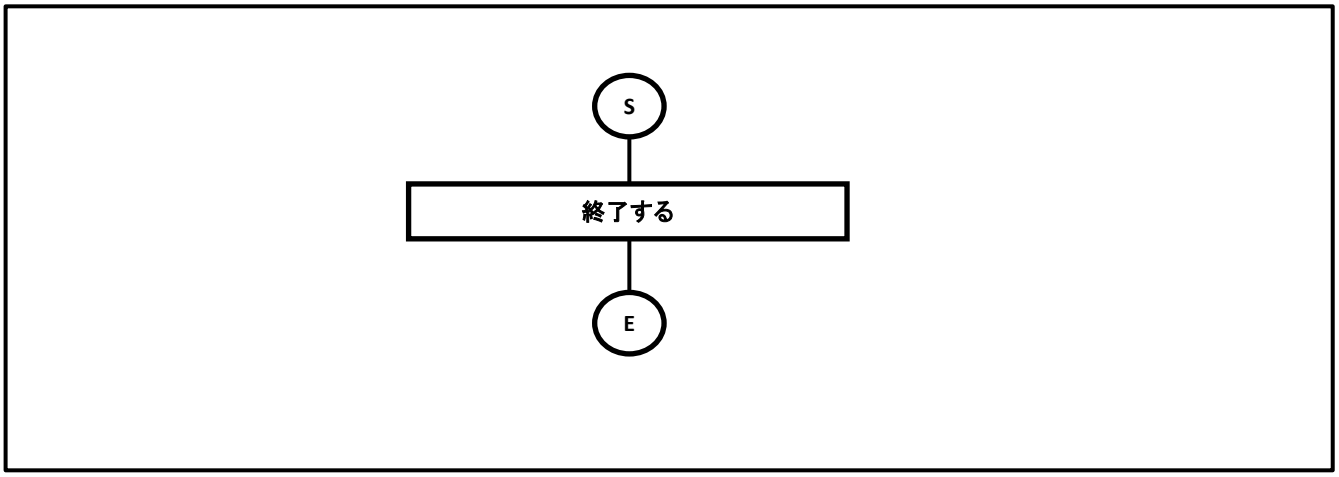
1 FrmLoginクラスの変数frmLoginのShow()メソッドを呼び出す
2 自画面を閉じる

メソッド詳細

メソッド名	BtnEnd_Click

● 終了ボタン押下

【処理フロー】



【処理内容】

1 FrmLoginクラスの変数frmLoginのClose()メソッドを呼び出す

社員管理システム

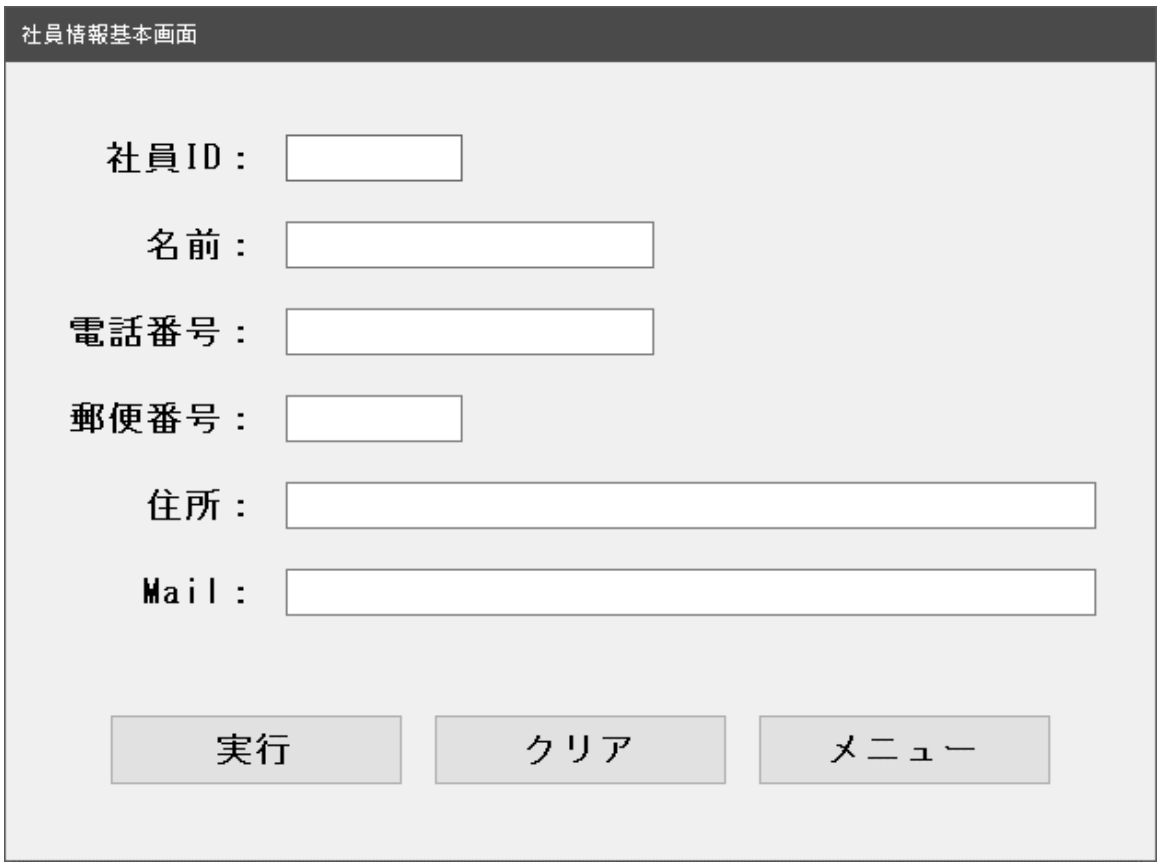
システム名	社員管理システム
処理名	社員情報基本画面
プログラム名	FrmEmp

処理名	社員情報基本画面	クラス名	FrmEmp
業務処理概要			

【処理概要】

- 登録画面のクラスと検索画面のクラスに対する基本クラスのため、このクラスの画面はシステム上表示されない
- 実行ボタンが押下された場合の処理は本クラスには記述せず、継承先のクラスで実装する
- クリアボタンが押下された場合は、すべての項目をクリアする
- メニューボタンが押下された場合は、メニュー画面へ遷移する

画面レイアウト



画面入力項目

項番	項目名	内容	特記事項

※ 登録画面のクラスと検索画面のクラスに対する基本クラスのため、このクラスの画面はシステム上表示されない

画面操作項目

項番	項目名	内容	特記事項
1	実行	（記述しない）	メソッドにvirtualを指定する
2	クリア	入力内容をクリアする	
3	メニュー	メニュー画面へ戻る	

クラス詳細(フォーム設計(クラス設計)／変数設計)

【フォーム設計】

項番	設定項目	値
1	ファイル名	FrmEmp.cs
2	Name	FrmEmp
3	ControlBox	False
4	Size	640, 480
5	StartPostion	CenterScreen
6	Text	社員情報基本画面

【コントロール設計】

項番	ロール名	設定項目	値
1	LblId	Font	MSゴシック,14.25pt,style=Bold
		Location	24, 32
		Size	120, 40
		Text	社員ID：
		TextArigin	MiddleRight
2	LblName	Font	MSゴシック,14.25pt,style=Bold
		Location	24, 80
		Size	120, 40
		Text	名前：
		TextArigin	MiddleRight
3	LblPhone	Font	MSゴシック,14.25pt,style=Bold
		Location	24, 128
		Size	120, 40
		Text	電話番号：
		TextArigin	MiddleRight
4	LblPost	Font	MSゴシック,14.25pt,style=Bold
		Location	24, 176
		Size	120, 40
		Text	郵便番号：
		TextArigin	MiddleRight
5	LblAddress	Font	MSゴシック,14.25pt,style=Bold
		Location	24, 224
		Size	120, 40
		Text	住所：
		TextArigin	MiddleRight
6	LblMail	Font	MSゴシック,14.25pt,style=Bold
		Location	24, 272
		Size	120, 40
		Text	Mail：
		TextArigin	MiddleRight

7	TxtId	Font	MSゴシック,14.25pt
		Location	152, 40
		MaxLength	4
		Size	96, 26
		TabIndex	0
8	TxtName	Font	MSゴシック,14.25pt
		Location	152, 88
		MaxLength	50
		Size	200, 26
		TabIndex	1
9	TxtPhone	Font	MSゴシック,14.25pt
		Location	152, 136
		MaxLength	50
		Size	200, 26
		TabIndex	2
10	TxtPost	Font	MSゴシック,14.25pt
		Location	152, 184
		MaxLength	8
		Size	96, 26
		TabIndex	3
11	TxtAddress	Font	MSゴシック,14.25pt
		Location	152, 232
		MaxLength	50
		Size	440,26
		TabIndex	4
12	TxtMail	Font	MSゴシック,14.25pt
		Location	152, 280
		MaxLength	50
		Size	440,26
		TabIndex	5
13	BtnExecute	Font	MSゴシック,14.25pt,style=Bold
		Location	56, 360
		Modifiers	Public
		Size	160, 40
		TabIndex	6
		Text	実行
14	BtnClear	Font	MSゴシック,14.25pt,style=Bold
		Location	232, 360
		Size	160, 40
		TabIndex	7
		Text	クリア
15	BtnMenu	Font	MSゴシック,14.25pt,style=Bold
		Location	408, 360
		Size	160, 40
		TabIndex	8
		Text	メニュー

メソッド詳細

メソッド名	BtnExecute_Click

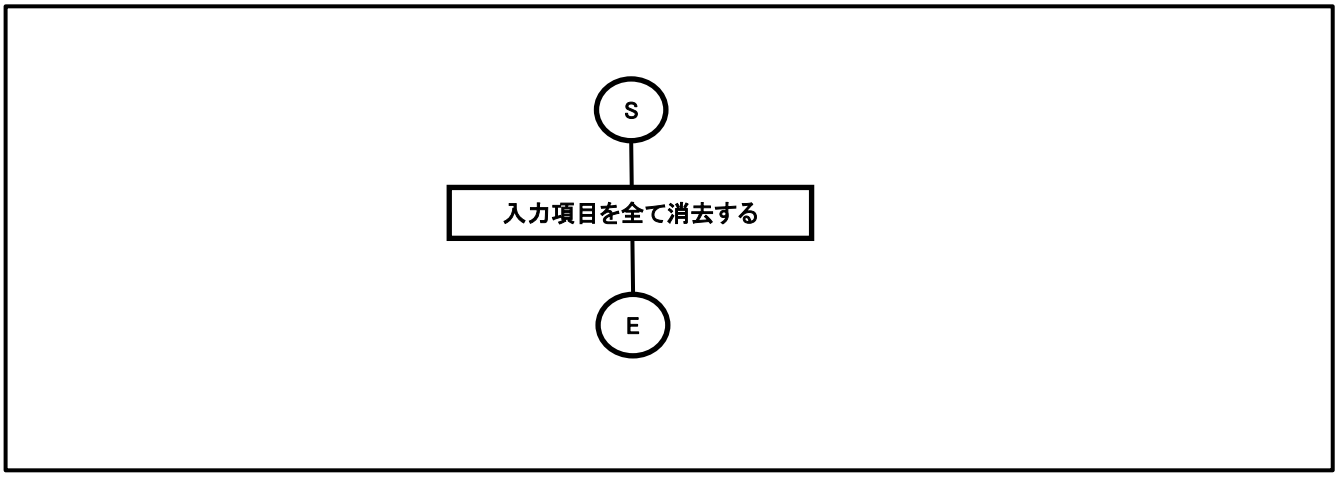
● 実行ボタン押下
修飾子にvirtualを指定し、処理は何も記述しない

メソッド詳細

メソッド名	BtnClear_Click

● クリアボタン押下

【処理フロー】



【処理内容】

1 各テキストボックスの文字列をクリアする
2 社員IDのテキストボックスへフォーカスを設定する

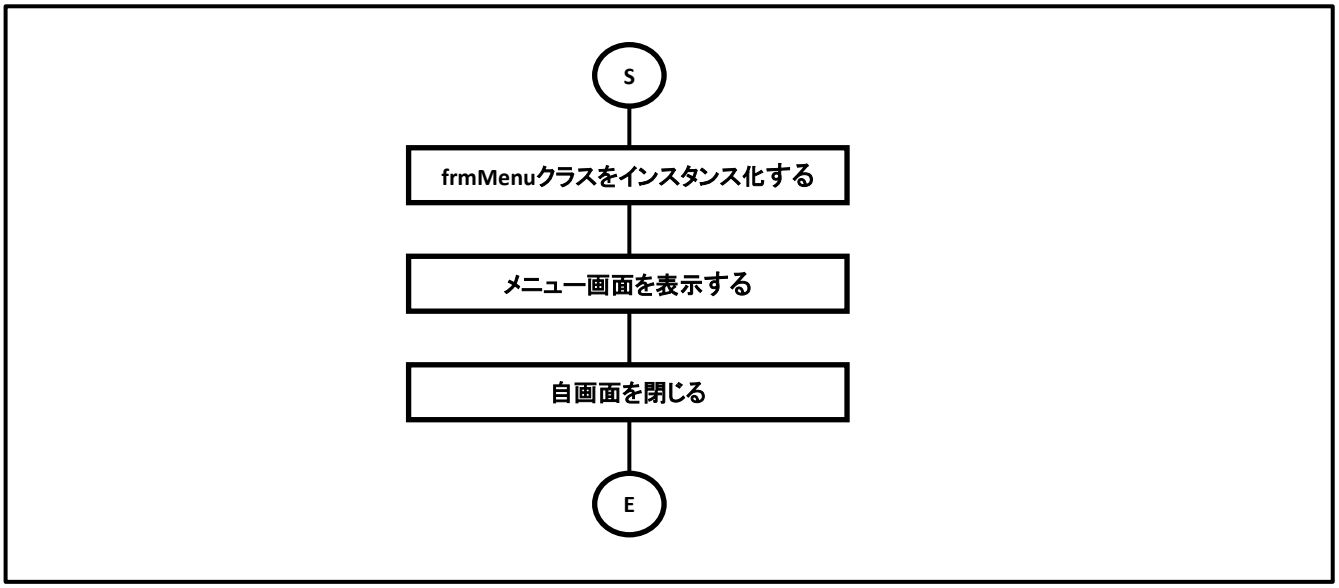
※ 継承先のクラスから呼び出せるように、メソッドのアクセス修飾子にprotectedを設定する

メソッド詳細

メソッド名	BtnMenu_Click

● メニューボタン押下

【処理フロー】



【処理内容】

1 FrmMenuクラスをインスタンス化する
2 メニュー画面を表示する
3 自画面を閉じる

社員管理システム

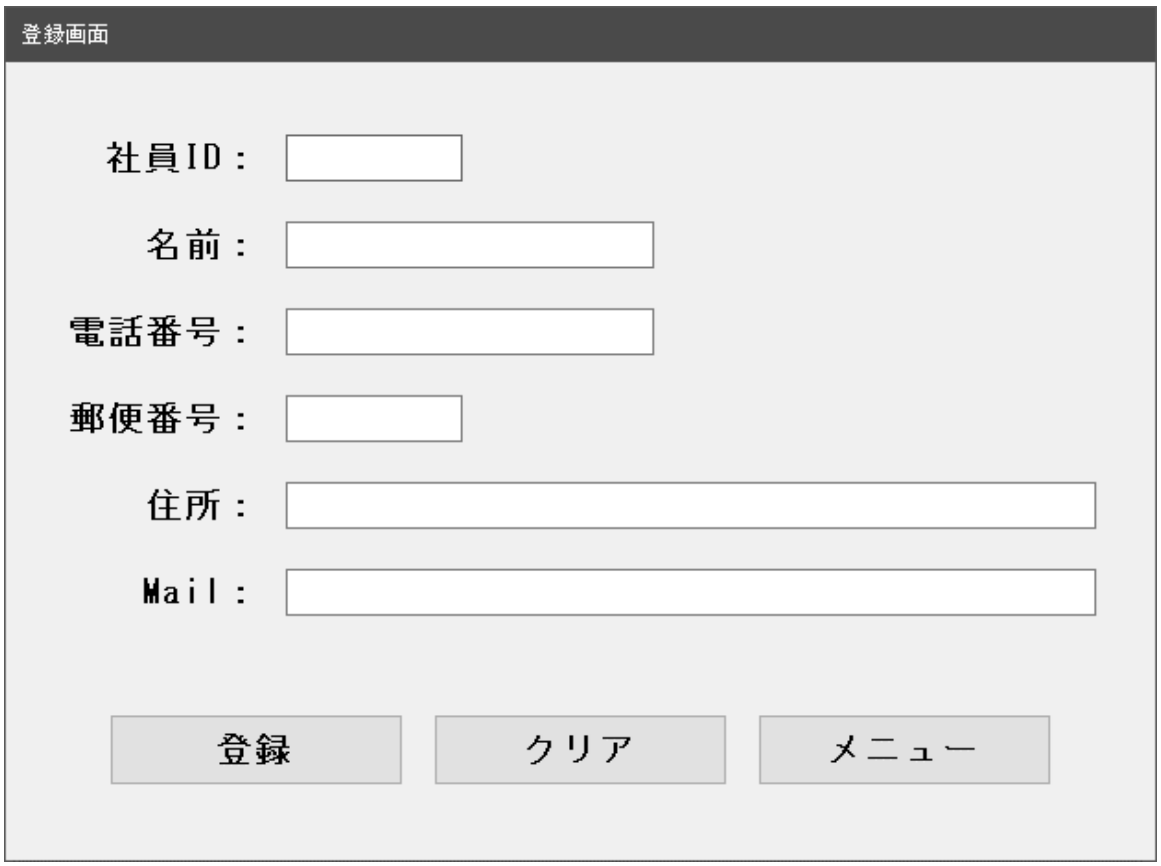
システム名	社員管理システム
処理名	登録画面
プログラム名	FrmRegistration

処理名	登録画面	クラス名	FrmRegistration
業務処理概要			

【処理概要】

- 社員管理システムにデータを登録するための画面である
- 社員ID、名前とも必須入力で未入力の場合はエラーとする
- 既に登録されているIDが指定されている場合はエラーとする
- 社員IDに数値以外が入力された場合はエラーとする
- 各入力項目の内容のチェックは行わない
- 登録ボタンが押下された場合は、入力されたデータが登録できるデータか否かをチェックする
  登録できる場合は、入力されたデータをemployeeテーブルに登録する
  登録できない場合は、エラーメッセージを表示する
- クリアボタンが押下された場合は、すべての項目をクリアする
- メニューボタンが押下された場合は、登録画面を終了し、メニュー画面へ遷移する

画面レイアウト



画面入力項目

項番	項目名	内容	特記事項
1	社員ID	社員IDを入力する	
2	名前	名前を入力する	
3	電話番号	電話番号を入力する	
4	郵便番号	郵便番号を入力する	
5	住所	住所を入力する	
6	Mail	メールアドレスを入力する	

画面操作項目

項番	項目名	内容	特記事項
1	登録	登録処理を実行する	メソッドにoverrideを指定する
2	クリア	入力内容をクリアする	基本クラスから変更しない
3	メニュー	メニュー画面へ戻る	基本クラスから変更しない

クラス詳細(フォーム設計(クラス設計)／変数設計)

【フォーム設計】

項番	設定項目	値
1	ファイル名	FrmRegistration.cs
2	Name	FrmRegistration
3	Text	登録画面

【コントロール設計】

項番	コントロール名	設定項目	値
1	BtnExecute	Text	登録

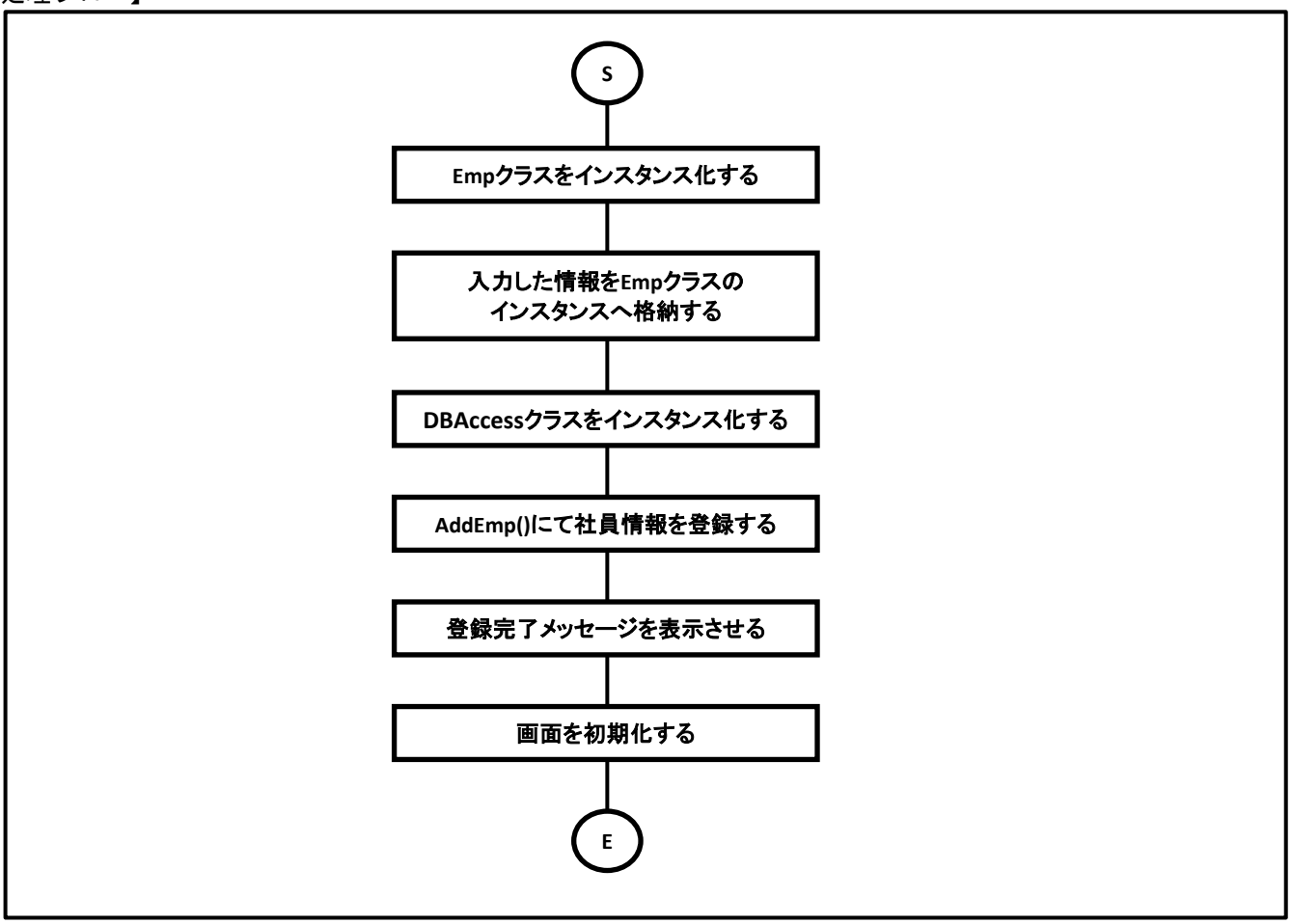
※ 上記以外は継承元であるFrmEmpクラスからの変更点はなし

メソッド詳細

メソッド名	BtnExecute_Click

● 登録ボタン押下

【処理フロー】



【処理内容】

1 以下の処理はtry句内に記述する
 1.1 Empクラスのインスタンスを生成し、変数empへ格納する
 1.2 各テキストボックスの内容をそれぞれ該当する変数emp内のプロパティへ格納する
 1.3 DBAccessクラスのインスタンスを生成し、変数dbへ格納する
 1.4 変数dbのAddEmp()メソッドを呼び出す。その際に、変数empを引数に指定する
 1.5 「登録が完了しました」とメッセージを表示する
 1.6 クリアボタンのイベント呼び出すことで入力内容をクリアする
  BtnClear_Click(this, System.EventArgs.Empty);
2 以下の処理はExceptionクラスを対象としたcatch句内に記述する
 2.1 捕捉した例外オブジェクト内のメッセージをメッセージボックスで表示させる
 2.2 社員IDテキストボックスへフォーカスを移動させる

※ 継承元のオーバーライドとするために、メソッドの修飾子にoverrideを設定する

社員管理システム

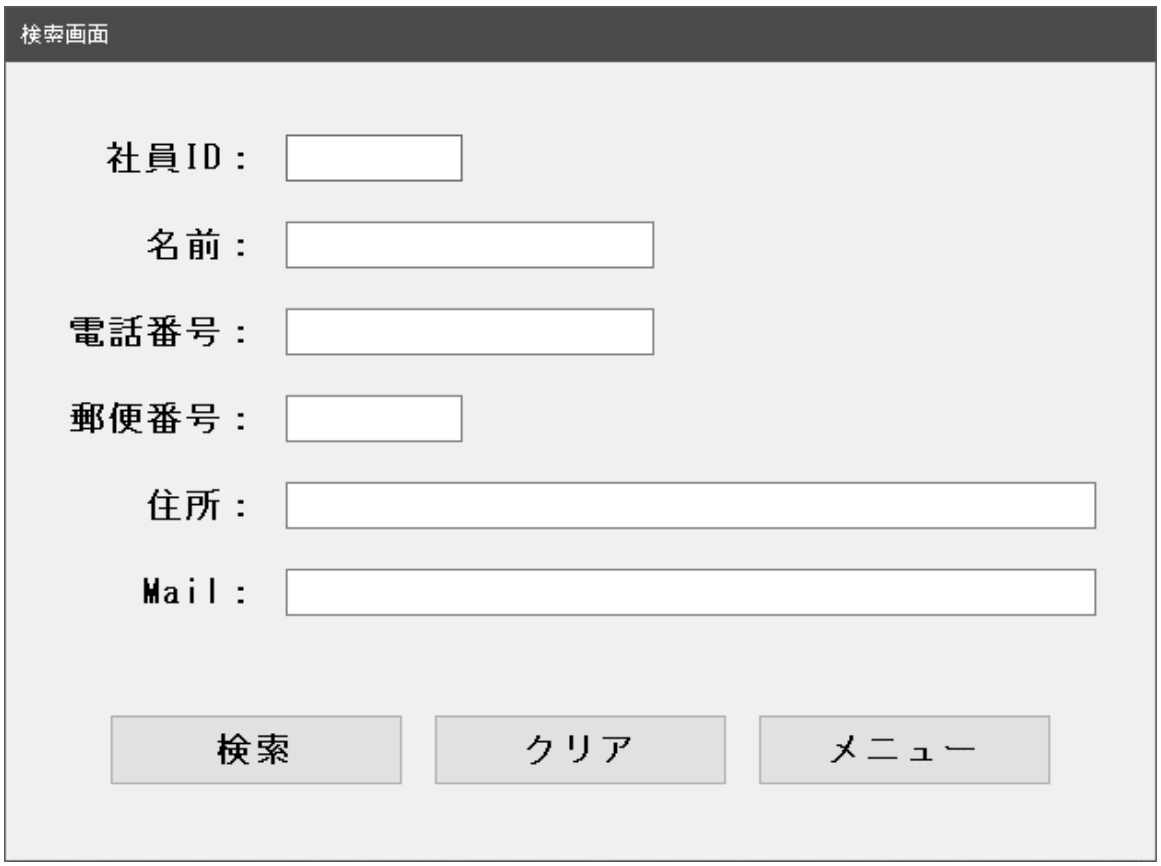
システム名	社員管理システム
処理名	検索画面
プログラム名	FrmSearch

処理名	検索画面	クラス名	FrmSearch
業務処理概要			

【処理概要】

- 社員情報を検索し、結果を出力するための画面である
- FrmEmpクラスを基本クラスとして画面を生成する
- 社員IDは必須入力で未入力の場合はエラーとする
- 社員IDに数値以外が入力された場合はエラーとする
- 検索ボタンが押下された場合は、社員IDをもとにデータベースから社員情報を取得する
  情報が取得できた場合は、取得したデータの内容を画面へ表示する
  情報が取得できなかった場合は、エラーメッセージを表示する
- クリアボタンが押下された場合は、すべての項目をクリアする
- メニューボタンが押下された場合は、登録画面を終了し、メニュー画面へ遷移する

画面レイアウト



画面入力項目

項番	項目名	内容	特記事項
1	社員ID	社員IDを入力する	

※ 他の項目は表示のみで入力は不可とする

画面操作項目

項番	項目名	内容	特記事項
1	検索	検索処理を実行する	メソッドにoverrideを指定する
2	クリア	入力内容をクリアする	基本クラスから変更しない
3	メニュー	メニュー画面へ戻る	基本クラスから変更しない

クラス詳細(フォーム設計(クラス設計)／変数設計)

【フォーム設計】

項番	設定項目	値
1	ファイル名	FrmSearch.cs
2	Name	FrmSearch
3	Text	検索画面

【コントロール設計】

項番	コントロール名	設定項目	値
1	BtnExecute	Text	検索

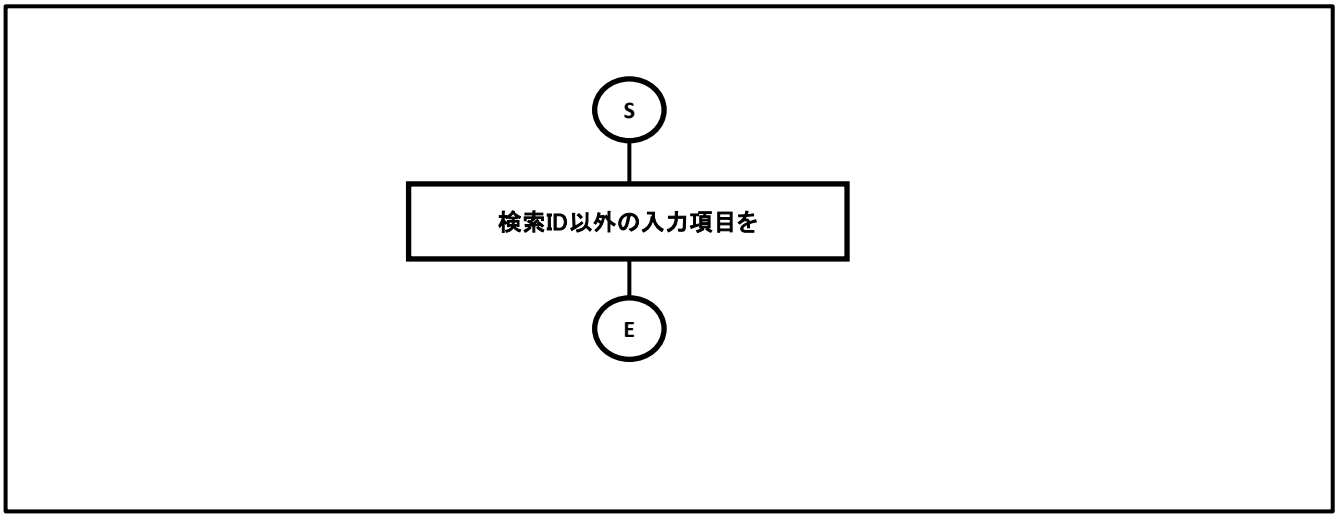
※ 上記以外は継承元であるFrmEmpクラスからの変更点はなし

メソッド詳細

メソッド名	FrmSearch_Load

● 自画面表示

【処理フロー】



【処理内容】

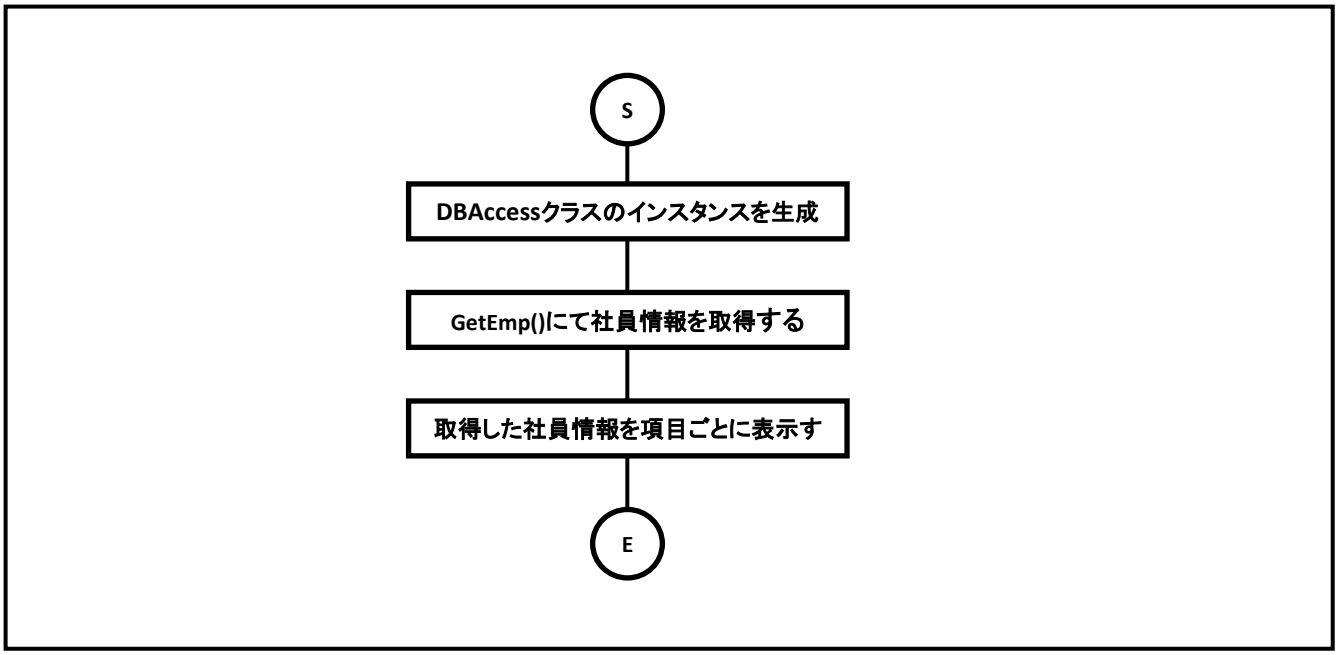
1 社員ID以外の各テキストボックスのReadOnryプロパティにtrueを代入する
2 社員ID以外の各テキストボックスのBackColorプロパティにSystemColors.Windowを代入する

メソッド詳細

メソッド名	BtnExecute_Click

● 検索ボタン押下

【処理フロー】



【処理内容】

1 以下の処理はtry句内に記述する

1.1 DBAccessクラスのインスタンスを生成し、変数dbへ格納する
変数dbのGetEmp()メソッドを呼び出し、結果をEmpクラス型の変数empへ格納する
変数emp内の各プロパティをそれぞれ該当するテキストボックスへ表示させる

2 以下の処理はExceptionクラスを対象としたcatch句内に記述する

2.1 捕捉した例外オブジェクト内のメッセージをメッセージボックスで表示させる

※ 継承元のオーバーライドとするために、メソッドの修飾子にoverrideを設定する

社員管理システム

システム名	社員管理システム
処理名	データベース設計

処理名	データベース設計	DB名	EmpSys_CSDB
データベース仕様概要説明			

【使用目的】
- 社員管理システムにおける必要データを保持しているデータベースである
- loginuserテーブルはパスワードおよび、処理権限の管理を行うテーブルである
- employeeテーブルは社員のデータの管理を行うテーブルである

使用テーブル一覧

項番	データファイル名	テーブル名	備考
1	EmpSys_DB	loginuser	ユーザーID、パスワード、権限の管理
		employee	社員データの管理

テーブル設計詳細

【loginuser】

項番	フィールド名	データ型	長さ	KEY	Null許容	備考
1	id	int	-	○		
2	name	nvarchar	50			
3	password	nvarchar	50			
4	admin	nchar	1			1:管理者　0:一般

【employee】

項番	フィールド名	データ型	長さ	KEY	Null許容	備考
1	id	int	-	○		
2	name	nvarchar	50			
3	phone	nvarchar	50		○	
4	post	nvarchar	50		○	
5	address	nvarchar	50		○	
6	mail	nvarchar	50		○	

データ例

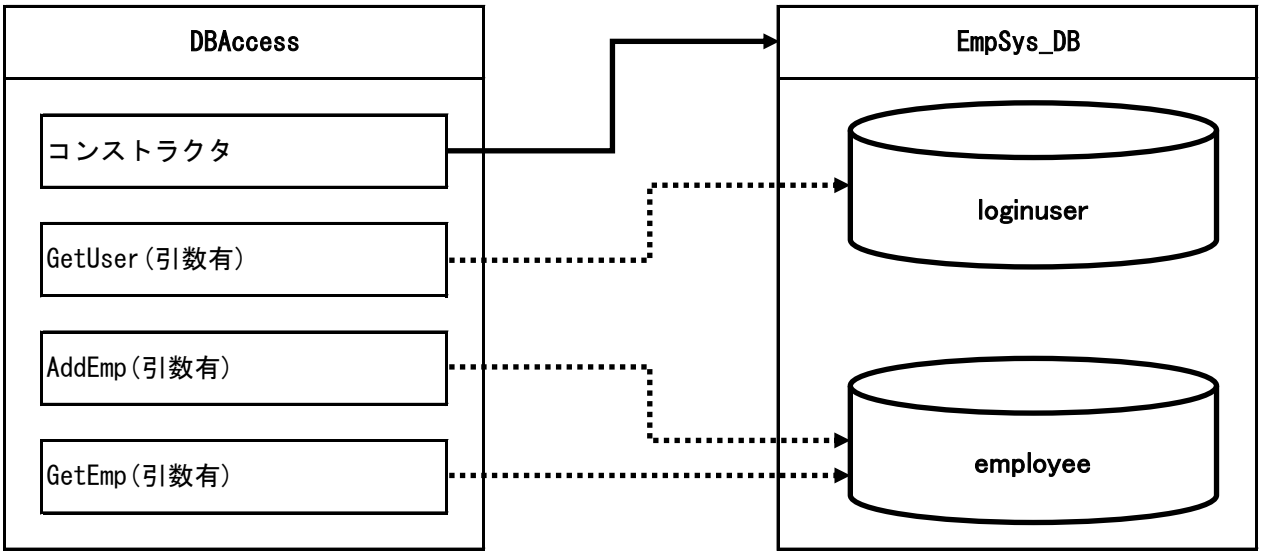
【loginuser】

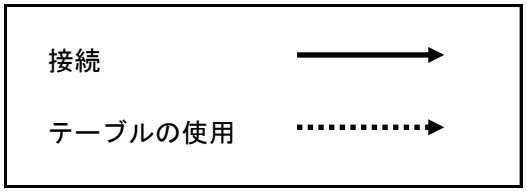
id	name	password	admin
1	ken	1111	1
2	test	2222	0

【employee】

id	name	phone	post	address	mail
1001	神田　拓海	000-1234-5678	123-4567	東京都千代田区	tk@kenschool.jp
1002	品川　葵	000-9876-5432	987-6543	東京都港区高輪	as@kenschool.jp

データベース相関図





テーブル仕様詳細

【loginuser】

項番	フィールド名	GetUser	AddEmp	GetEmp
1	id	検索条件	-	-
2	name	取得	-	-
3	password	取得	-	-
4	admin	取得	-	-

【employee】

項番	フィールド名	GetUser	AddEmp	GetEmp
1	id	-	検索条件/更新	検索条件
2	name	-	更新	取得
3	phone	-	更新	取得
4	post	-	更新	取得
5	address	-	更新	取得
6	mail	-	更新	取得