高知大学教育学部の情報数学のテキスト 文責::高知大学名誉教授 中村 治

Windows フォーム・アプリケーション

通常の C++ の解説と異なり、結果が正しいかどうか見れば分かる、関数のグラフを描くこと から始めます。これには Microsoft Visual Studio 2010 固有の機能を使い、一般的ではありませ ん。C++ の構文からの逸脱した、Microsoft による長い長い改変の歴史があり、それを解説して いる本を少なくとも一冊は読まないと C++ の文法書を読むだけでは完全には理解できないので、 Microsoft Visual Studio 2010 固有の機能は魔法を使うための呪文だと思ってそのまま使って下さ い。これらのプログラムを雛形に、他の関数のグラフを描きたいときは、Microsoft Visual Studio 2010 固有の機能でない部分の必要な修正をしてプログラミングすればいいです。以下のプログラ ミングでは、C++ の機能を全面的に使っているわけではなく、「より良い C としての C++」 し か使っていません。中学校・高等学校の先生になるため知っておくべき、常識(かっての高等学校 の教科書「数学A・数学B・数学C」のプログラミングの内容+ α)だけを述べています。

Microsoft visual studio 2019 community でも、以下のプログ ラムを作成できる方法を vc++2010、 vc++2013の使い方の後方 に追加しました。

まず、Microsoft Visual Studio 2010 を立ち上げる。

スタートページ - Microsoft Visual Studio Aca	demic	
ファイル(E) 編集(E) 表示(⊻) デバッグ(D) :	チーム(M) データ(A) ツール(I) テスト(<u>S</u>) ウィンドウ(<u>W</u>) ヘルプ(<u>H</u>)	
	- 💭 - 📖 🕨 - 🛛 - 💋 loseing	- 🔩 🕾 📷 😒 📯 🙄
スタートページ × ソリューション エクスプロ	-5-	★ 検索と置換 × マ
		フォルダーを指定して検索・
Microsoft" Visual Studio: 2010 P	refersional	検索する文字列(N):
	rolessional	-
	作業の関始 ガイダンスとリソース 昇新ニュース	検索対象(L):
1 Team Foundation Server への接続		·
前しいプロジェクト	ようこそ Windows Web クラウド Office SharePoint	
「「」プロジェクトを開く		+ 検索オプション(0)
し こ こ こ こ こ こ こ こ こ こ こ こ こ こ こ こ こ こ	Visual Studio 2010 の新機能 のリリースの新機能をご覧ください。 Visual Studio 2010 の既要 NET Framework 4 の新機能 Visual Studio スタート ページのカスタマ れば Visual Studio スタート ページのカスタマ れば	 ● 結果オプション(5) ● すへて終
覧 ■ コマンド ウィンドウ ■ 出力 推供言了		
準備元」		

スタートページがなければ、「表示」のメニューでスタートページをクリックすればよい。

上図は Professional 版であることを示していますが、それは私が幾何学の研究の為のプログ ラムや数独など各種パズルを解いたり問題を作ったりするプログラムやオセロや囲碁などの各種 ゲームのプログラムを作るためにメモリーを沢山使う64ビットのプログラミングがしたいから、 Professional 版を使っているからです。学生さんはアカデミック版を購入できますから、1万円位で 生協で Professional 版を購入出来ますが、以下のプログラムは Professional 版固有の機能は使って いないので、学生さんが勉強のために、以下のプログラムを実行するだけなら、無料の Microsoft Visual Studio 2010 Express 版で十分です。 新しいプロジェクト … をクリックする。

新しいプロジェクト 最近使用したテンプレー	- 1				
RDI(使用) UCF シフレー インストールされたテン インストールされたテン インスト CLR 全般 MFC テスト Win32 ▷ 他の言語 ▷ その他のプロジェク ▷ データベース ▷ データベース ▷ テスト プロジェクト オンライン テンプレー	ト の種類 、 ト		mework 4 ・ 並べ替え基準: 既定 Win32 コンソール アプリケーション MFC アプリケーション Win32 プロジェクト をのプロジェクト ATL プロジェクト MFC DLL Windows フォーム アプリケーション CLR コンソール アプリケーション	Visual C++ Visual C++ Visual C++ Visual C++ Visual C++ Visual C++ Visual C++	 イシストールされたテンブレートの検索 ぞ初: Visual C++ Windows ユーザー インターフェイスを 含むアブリケーションを作成するための プロジェクトです。
名前(<u>N</u>):	<名前を入力し	本 てください	空の CLR プロジェクト \>	Visual C++	•
場所(<u>L</u>):	D:¥jyugyou¥			•	参照(<u>B</u>)
ソリューション名(<u>M</u>):	<名前を入力し	てください	/>		 ✓ ソリューションのディレクトリを作成(<u>D</u>) ○ ソース管理に追加(<u>U</u>)
					OK OK キャンセル

Windows フォームアプリケーションをクリックし、図では場所 (L):が D:\jyugyou\となっていま す(自分で適当なディレクトリ、例えば D:\jyugyou\を作り、ここを指定すれば、プログラムが D:\jyugyou\に作られます)が、何処にプログラムが保存されるか気にしなければ、そのままで良 いですので、名前 (N): のエディットボックスに例えば en2 と入力し、

新しいプロジェクト				? <mark></mark>
最近使用したテンプレ-	-ト 🚺	NET Framework 4 ▼ 並べ替え基準: 既定	•	インストールされたテンプレートの検索 👂
インストールされたテン	ップレート	(************************************	Visual C++	種類: Visual C++
▲ Visual C++ ATL CLR 全般 MFC テスト Win32		 MFC アプリケーション Win32 プロジェクト 空のプロジェクト 	Visual C++ Visual C++ ⋿ Visual C++	Windows ユーザーインターフェイスを 含むアプリケーションを作成するための プロジェクトです。
 ▶ 他の言語 ▶ その他のプロジェク ▶ データペース ▶ テスト プロジェクト オンライン テンプレー 	トの種類 ト	▲ ATL プロジェクト ● MFC DLL ● Windows フォーム アプリケーション ● CLR コンソール アプリケーション	Visual C++ Visual C++ Visual C++ Visual C++	
		🍫 空の CLR プロジェクト	Visual C++ *	
名前(<u>N</u>):	en2			
場所(<u>L</u>):	D:¥jyugyou¥			参照(<u>B</u>)
ソリューション名(<u>M</u>):	en2] ソリューションのディレクトリを作成(<u>D)</u>] ソース管理に追加(<u>U)</u>
				OK キャンセル

OK ボタンをクリックする。

👓 en2 - Microsoft Visual Studio Academic	
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) データ(A) 書式(O) ツール(T) テスト(S) パ 日・四・22 日 (2) (2) (2) (2) (2) (2) (2) (2) (2) (2)	ウィンドウ(W) ヘルプ(H) - 😡 😤 🍶 🖄 🎇 🍹
Form1.h (デザイン) × スタート ページ ソリューション エクスプローラー ・	プロパティ 検索と温強 × マ (2) フォルダーを指定して検索 • * * * * * * * * * * * * * * * * * *
民 エラー一覧 ■ コマンド ウィンドウ 目 出力 プロジェクト 'en2' の作成に成功しました。	.1

Microsoft Visual Studio 2010 では、Window フォーム・アプリケーションを作るのはこのよう に簡単でしたが、Microsoft Visual Studio 2013 では、Window フォーム・アプリケーションの選 択肢がなくなり、「猫でもわかる Windows プログラミング」にあるような昔ながらのコードによる Window フォーム・アプリケーションは作れますが、Microsoft Visual Studio 2010 で普通にやっ ていた手軽さに慣れてしまうと Microsoft Visual Studio 2013の選択肢にあるような複雑なソフト ではなく、単にこのような単純なソフトを作りたい素人は途方にくれます。幸い、VC++ 2013 で Windows フォームアプリケーションのプログラムを作る方法をインターネットで見つけたので、 かなり複雑ですが、やってみます。

「新しいプロジェクト」で、「Visual C++」の「CTR」を選択し、「空の CLR プロジェクト」を 選択し、「名前」を指定し、

		新しいプロジェクト		? ×
▶ 最近使用したファイル		.NET Framework 4.5 → 並べ替え基準: 既定	• # 🗉	インストール済み テンプレート の検索 (Ctrl: 🔎 -
▲ インストール済み		りつう ライブラリ	Visual C++	種類: Visual C++
 ▲ テンプレート ▷ Visual Basic ▷ Visual C# 		で、 CLR コンソール アプリケーション	Visual C++	ローカル アプリケーションを作成するための空のプ ロジェクトです。
✓ Visual C++ Windows	217	空の CLR プロジェクト	Visual C++	
ATL CLR 全般 MFC デスト Win32 Visual F# SQL Server TypeScript JavaScript P JavaScript	h⊾ns⊞#ã ▼			
▶ オンライン		オンラインでテンプレートを検索するには、ここをクリックします		
名前(<u>N</u>):	WinFormApp			
場所(<u>L</u>):	C:¥vcwinprog	Dialog¥	•	参照(<u>B</u>)
ソリューション名(<u>M</u>):	WinFormApp			✓ ソリューションのディレクトリを作成(<u>D</u>) ○ ソース管理に追加(<u>U</u>)
				ОК + +>>セル

「OK」ボタンをクリックする。

WinFormApp - Microsoft Visual Studio アイル(E) 編集(E) 表示(Y) プロジェジト(E) ジェバッグ(D) チーム(M) アール(I) テスト(S) 分析(M) タインドグ(M) ヘルプ(H) ○・○ ○ 〒・⑤・■ 単 ツ ・ ○・・ ▶ ローカル Windows デバッパー・ ○・ ○ Debug ・ ● #・ □	▼5 クイック起動 (Ctrl+0	2) / × 中村治 <mark>-</mark>
9-A 10770- 3-	yuz->>> エカスカーラー ● ● ▲ ● ● = ● ● yuz->>>> コカスカーラー ■ yuz->>> マショー ■ yuz->>>> WinForm ■ y-スファイル ■ yuz-スファイル ■ yuz-スファイル ● ● 参照在存留系 > Winto-Say, 702 m-5-	● マンゴン モー クランドー
	プロパティ	- ↓ ×
	WinFormApp プロジェクトの	プロパティ・
	11 g. P	
	日 その他	
	(名明) ブロジェクト ファイル	WinFormApp C:XvcwinprogXDialogXWinForr
	プロジェクトの依存関係	C.+vcwinprog+biolog+winPon
	ルート名前空間	WinFormApp
	(名前) プロジェクト名を指定します。	
1万一一覧 出力	ツールボックス プロパティ	
·		

右側の「ソリューションエクスプローラー」で、WinFormApp(プロジェクト名)を右クリック し、「追加」->「新しい項目」をクリックする。左欄で「Visual C++」->「UI」を選択し、右 欄で、「Windows フォーム」を選択して、

		新しい項目の追加	- WinFormApp	? ×
▲ インストール済み		並べ替え基準: 既定 ・	# E	インストール済み テンプレート の検索 (Ctrl・ 🔎 🗸
▲ Visual C++ Windows ストン	v	E目 Windows フォーム	Visual C++	種類: Visual C++ 他の Windows コントロールを合わ CL P フォー
		CLR ב-ザー コントロール	Visual C++	ムを作成します。
コード データ リソース Web ユーティリティ プロパティ シート Test		MFC リボン定義 XML ファイル	Visual C++	
▶ オンライン		オンラインでテンプルートを検索す	るには、ここをクリックします。	
名前(<u>N</u>):	MyForm1.h			
場所(<u>∟</u>):	C:¥vcwinprog¥l	Dialog¥WinFormApp¥WinFormApp	•	参照(B) 追加(A) キャンセル

「追加」をクリックする。名前は「MyForm.h」となっていて、従来の Form1.h に相当する。

♥ WinFormApp - Microsoft Visual Studio マイル(E) 編集(E) 表示(Y) プロジェウト(E) ビルド(E) デパッグ(D) チーム(M) アール(I) テスト(S) 分析(M) ウインドク(M) ヘルプ(H) ◎ • • ○ 図・○ □ 20 · ○ □ 10 · ○ □	】 て5 クイック起動 (Ctrl+Q	♪ ● × 中村治 - Ⅲ
Y MyForm ●	yJューション エクスプローラー () ((((((((((((((((● チ ー 第 (Ctrl+:) ゆ (1 力5:20F)
		-6 19270-5- 952 Ez-
出力 出力元の表示(<u>S</u>):】 全 当 当 短 №		WinFormApp C:¥vcwinprog¥Dialog¥WinForr
エラー「賢 出力 単概77	ルー・ロッジョン回 (名前) プロジェクト名を指定します。 ツールボックス プロパティ	тигонодр

「MyForm.h」ができ、フォームデザイナに、Windows フォームが表示される。

次に、エントリーポイントを作成します。エントリーポイントとは、プログラムを起動したとき、 最初に実行するプログラムであり、具体的には main() 関数のことである。右側の「ソリューション エクスプローラー」で、「ソースファイル」を展開し、「MyForm.cpp」を表示する。「MyForm.cpp」 を右クリックし、「名前の変更」を選択して、「WinFormApp.cpp」(WinFormApp.: プロジェク ト名)に書き換える。

 WinFormApp - Microsoft Visual Studio アイル(E) 編集(E) 表示(Y) プロジェクト(E) どルド(E) デパッグ(E) チーム(M) ツール(I) デスト(S) 分析(M) ウインドク(M) ヘルプ(E) ○・○ 図 ・ □ 目 ・ □ - カル Windows デパッガー・ ○・ □ Debug ・ 声 ・ □ + □ □ 中 山 国 工 器 座 中 + ゴ ゴ ・ 	↓ 「 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、 、	2) P - P : 中村治 - [×
└ MyForm.h [デザイン] + ×	▼ ソリューション エクスプローラー	▼ 早 :	×
	00 Å 0- # Ø	1 · · · -	
5 MyForm DD 🖾	ソリューション エクスプローラー の	検索 (Ctrl+:)	- C
	」 ソリューション 'WinForm	App' (1 プロジェクト)	-
ý l	🔺 🐚 WinFormApp		
	4 🐖 ソース ファイル		
		hrchb	
	MyForm.h		
	📁 リソース ファイル		
	▶ 🥫 外部依存関係		
	ソリューション エクスプローラー	チーム エクスプローラー クラス ビュー	
	プロパティ		
	WinFormApp.cpp ファイル	のプロパティ	
	2 9 P		
出力 - サン	< I その他		
出力元の表示(5): ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	(名前)	WinFormApp.cpp	11
	コンテンツ	False	
	ファイルの種類	C/C++ 3-F	
	ノロシェクトに言まれる	Irue	٣
	(名前) ファイル オブジェクトに名前を付け	けます。	
1万	ツールボックス プロパティ		
· · · · · · · · · · · · · · · · · · ·			

「ソリューションエクスプローラー」で WinFormApp (プロジェクト名)を選択した状態で、メ ニューから、「プロジェクト」ー>「プロパティ」をクリックする。

MyForm.h [デザイン] + ×					- جالا - 0 0 1	ン エクスプローラー ☆ 'o・ ≠	- 4 ×
MyForm		WinFormApp	プロパティページ	? 🔷		ン エクスプローラー の検索(ニーション 'WinFormApp'(WinFormApp	(Ctrl+:) ይ (1 ፓዐジェクト)
	→ 構成(C): N/A ▲ 共通プロパティ 参照	 ブラットフォーム(P): 対象のフレームワーク: .NETFran 参照(R): 	N/A nework,Version=v4.5	∨ 構成マネージャー(0)		 ↓ ソース ファイル ** WinFormApp.cpp ヘッダー ファイル ■ MyForm.h)
		2त्रम - अ System - अ System.Data - अ System.Vindows.F - अ System.Xml		¥.,¥.,¥Program Files (x8) False 2) True True C:¥Program Files (x86)¥Refe System, Version=4.0.0.0, Q.	УЛ1-Эз 7D/7-7	 リンース・プァイル 小部依存関係 ンエウスプローラー チーム: 	エクスプローラー クラス ピュー ~ ♀ >
出力 出力元の表示(S):		新しい参照の追加(N)	参照の削除(E)		WinForm 豊 型 日 その他 (名前)	mApp プロジェクトのプロパ: ダ	74
			ОК	キャンセル 適用(A)	(186) プロジェ プロジェ ルート名	クト ファイル C:: クトの依存関係 G前空間 Wi	¥vcwinprog¥Dialog¥WinFo
					(名前) プロジェクト	名を指定します。	
エラー一覧出力					ツールポック	フス プロパティ	

「WinFormApp(プロジェクト名)プロパティページ」が開くので、左の欄で、「構成プロパ ティ」->「リンカー」->「システム」と選択し、右の欄で「サブシステム」を選択し、右端に現 われる「V」をクリックし、現われたリストから「Windows(/SUBSYSTEM:WINDOW)」を選択する。

	WinFormApp 2	プロパティ ページ	? ×
構成(C): アクティブ(Debug)	✓ プラットフォーム(P):	アクティブ(Win32) 、	ゲ 構成マネージャー(O)
全般 デバッグ VC++ディレクトリ ト C/C++ リンカー 全般 入力 マニフェストファイル デバッグ システム 最適化 埋め込み IDL Windows メタデータ	サブシステム 最低限必要なパーション ヒープ サイズの設定 ヒープ コミット サイズの設定 スタックのサイズの設定 スタックのコミット サイズ 大きいサイズのアドレス ターミナル サーバー CD からスフップ実行 ネットワークからスフップ実行 ドライバー	Windows (/SUBSYSTEM いいえ いいえ 設定なし	:WINDOWS)
詳細設定 すべてのオプション コマンドライン ト マーフェフト ツール	サプシステム /SUBSYSTEM オプションは、オペL の選択によって、リンカーが選択する。	ノーティング システムに .exe ファイルの実行プ エントリ ポイント シンボル (またはエントリ ポイ OK =	ち法を指定します。サブシステム ント関数) が影響を受けます。 キャンセル 適用(A)

右下の「適用」をクリックする。

			WinFormApp 7	プロパティ ページ	? 🗙
構成(C):	アクティブ(Debug)		✓ プラットフォーム(P):	アクティブ(Win32)	▼ 構成マネージャー(0)
4 7 ∨ 0 C	一般 バッグ C++ディレクトリ /C++ シカー 全般 入力 マニフェストファイル デバッグ システム 毎週化 埋め込み IDL	^	サブシステム 最低限必要なパージョン とーブ サイズの設定 とーブ コミット サイズの設定 スタックのコミット サイズ スシックのコミット サイズ 大きいサイズのアドレス ターミナル サーバー CD からスフップ実行 ネットワークからスフップ実行 ドライバー	Windows (/SUBS) いいえ いいえ 設定なし	YSTEM:WINDOWS)
< N 7	Windows メタデータ 詳細設定 すべてのオプション コマンド ライン -フェフト ツー11. >	~	サブシステム /SUBSYSTEM オブションは、オペレ の選択によって、リンカーが選択するユ	ーティング システムに .exe ファイル こントリ ポイント シンポル (またはエン OK	の実行方法を指定します。サプシステム トリポイント関数) が影響を受けます。 キャンセル 適用(A)

つぎに、「構成プロパティ」->「リンカー」->「詳細設定」を選択し、右欄の「エントリポイント」に、「main」と入力して、

			WinFormApp プロパティ	(ページ	?	×
構成(C): 	アクティブ(Debug) 一でッグ 「マクライブ(Debug) 一でッグ 「マーナーディレクトリ 「マーナ・ディレクトリ 「マーナ・ディレクトリ 「マーナ・ディレクトリ 「マーナ・ディレクトリ ティンカー 全般 入力 マニフェスト ファイル デバッグ システム 最適化		WinFormApp プロパテ- エントリポイントなし チェックサムの設定 ペース アドレス ランダム化されたペース アドレス 固定ペース アドレス データ実行防止 (DEP) アセンブリの生成をオフにする 遅延読み込みされた DLL のパインドなし インポート ライブラリ セクションの結合 対象コンピューター 	(ページ いいえ はい (/DYNAMICBASE) いいえ (/FIXED:NO) はい (/NXCOMPAT) いいえ MachineX86 (/MACHINE:X86)	7	
	埋め込み IDL Windows メタデータ 詳細設定 すべてのオプション コマンド ライン	7	エントリ ポイント /ENTRY オプションは、.exe ファイルまたは D	LL の開始アドレスとしてエントリ ポイント閲数を指定しま	đ .	
< ► ⊽	- 7τ Z h W-II,	. *		OK キャンセル 通	箇用(A	.)

「OK」をクリックする。

♥ WinFormApp - Microsoft Visual Studio ワイル(E) 編集(E) 表示(Y) プロジェガト(E) どルド(E) デパッグ(D) チーム(M) ツール(I) アスト(E) 分析(M) ウインドク(W) ヘルプ(H) ◎ • ○ 後・② 論 論 例 ワ・♡・ ▶ ローカル Windows デパップ・・○ ・ Debug ・ 声・□ 下 非 山 神 工 器 音 非 ≑ 点 点	】 ▼5 りイック起動 (Ctrl+Q) P - 日 × 中村治 - 🏢
	ソリューション エクスプローラー	- ₽ ×
	© © ∰ `o - ≠ @	<u>ه ۲ –</u>
MyForm D 🖾	ソリューション エクスプローラー のれ	除索 (Ctrl+:) ・
	JULーション 'WinFormA	pp' (1 プロジェクト)
	WinFormApp	
	++ WinFormApp	.cpp
	▲ 🚍 ヘッダー ファイル	
p	ジョー リソース ファイル	
	▶ 👼 外部依存関係	
lb	ソリューション エクスプローラー 🗧	ーム エクスプローラー クラス ビュー
	プロパティ	- ¢ ×
	WinFormApp プロジェクトの	プロパティ ・
	11 gu 🖌	
	日 その他	
	(名前) プロジェクト ファイル	WinFormApp C:XvcwinprogXDialogXWinForr
	プロジェクトの依存関係	C.+veminprog+blulog+vini on
	ルート名前空間	WinFormApp
	(名前)	
	プロジェクト名を指定します。	
1万一一覧 出力	ツールボックス プロパティ	

「ソリューションエクスプローラー」で、「WinFormApp.cpp」(WinFormApp.:プロジェクト名) をダブルクリックする。

WinFormApp - Microsoft Visual Studio アイル(E) 編集(E) 表示(y) プロジェジト(E) デバッグ(D) チーム(M) ツール(I) デスト(S) 分析(M) タインドジ(M) ヘルプ(H) ○・○ 後・⑤ 単 単 ツ ・ ⑦・ ○ ト □ーカル Windows デバッパー・ ○・ ○ Debog ・ 輝 ・ □ 指 軍 強 東 領 領 領。	」 ▼5 ケイック起動 (Ctrl+Q) P = 日 × 中村治 - 日
Y WinFormApp.cpp * WinFormApp.cpp * (7D-/CVL Z3-7) * * Hinclude "WyForm-h" *	ソリューション エクスプローテー ・ Ⅱ × ● ▲ ● ・ # ● ● ● ● ソリューション エクスプローテー の装着 (Cri+1) P ・ □ ソリューション WinFormApp (1 プロジェクト) ・ ■ ▲ ③ ソース ファイル ・ ● ● ● M oFormApp ▲ ③ ハッチー ファイル ・ ● ● ● ● ● ● ● ○ ● ● ○ ● ● ○ ● ● ○ ● ● ○ ● ● ○ ● ● ○ ● ● ○ ● ● ○ ● ● ● <t< td=""></t<>
100%。4 出力 出力元の表示(S):	レーション レンジョー テーション レーラー ウラス ビュー プロ(ティ ・ 非 ×
25	ッ1xinyX <mark>プロ(ティ</mark> 1行 1列 1文字 棒入

ここに、

#include "MyForm.h"

using namespace WinFormApp;

```
[STAThreadAttribute]
int main()
{
```

```
Application::Run(gcnew MyForm());
return 0;
```

}

となるようにプログラムを打ち込む。



実行します。



上手くいきました。

パソコンが急に壊れ、新しく購入したパソコンに Microsoft visual studio 2012 professional をインストールしてみましたが、うまく

いきませんでした。Microsoft visual studio 2019 community を デフォルトのままインストールすると「猫でもわかる Windows プ ログラミング」で解説されているような昔の「使って天国、作って 地獄」の時代のプログラミングしかできません。しかし、Microsoft visual studio 2019 community でも、以下のプログラムを作成で きる方法をインターネットで見つけました。こんな抜道を見つけ出 す人たちはどんな勉強をしてきたんでしょうか?教育に携わってき た人間として興味があります。

すべてを書き直すのは大変ですから、最低限の修正で済ませること にします。

まず Visual Studio の公式サイトから、Community エディションのインストーラーをダウンロードします。



実行をクリックして、インストールを始めます。途中の



の画面のページになったとき、デフォルトではなく、デスクトップとモバイル(5)の 「.NET デスクトップ開発」と「C++によるデスクトップ開発」と右側のオプショ ンの「v142 ビルドツール用 C++/CU サポート」にチェックを入れることを忘 れないでください。後はデフォルトでいいです。

まず、Microsoft Visual Studio 2019 を立ち上げる。

最近開い	いた項目(B)		開始す	3
	• م		*	リポジトリのクローン(C) GitHub や Azure DevOps などのオンライン リポジトリか らコードを取得します
.	winTest.sln	2020/07/11 0:25		
	C:¥C++src¥winTest		?)	プロジェクトやソリューションを開く(<u>P</u>)
1	winTest2.sln	2020/07/11 0:10		ローカルの Visual Studio プロジェクトまたは .sln ファイル を照きます
	C:¥C++src¥winTest2			21MC 873
昨日			2-	ローカル フォルダーを閉く(に)
N	whist.sln	2020/07/10 1:49		日 パリレンスリレン そいわく(上) 任意のフォルダー内のコードに移動して編集します
4	C:¥texsrc¥情報数学¥集中講義3¥cardGame¥c#¥whist			
今週				
	whist.sln	2020/07/09 10:53	t D	新しいプロジェクトの作成(<u>N</u>)
	C:#texsrc#情報数学#集中講義3#cardGame#c##whisttest		_	開始するには、コードスキャフォールディング付きのプロジェ クト テンプレートを選択します
N	whist.sln	2020/07/09 7:29		
(G:¥texsrc¥情報数学¥集中講義3¥cardGame¥c#¥whist			コードなしで続行(<u>W</u>) →
_	at he so t	2020/07/05 12:00		

私は既に使っていて、左側にその履歴が残っていますが、それは気にしなくていいです。右の一 番下の「新しいプロジェクトの作成(N)」をクリックします。



この画面で、右側の選択肢の下の方にある「CLR 空のプロジェクト(.NET Framework)」を選択し、

見)には田」もプロジェクレニンプ!		ナバアの受強/() - ナバアのゴニットフォー / /D) - ナバアのゴロミークトの研研
取近1年用したノロシェクト ノノノル	$ (\underline{\mathbf{K}}) $	9 Y C (の) 目 2 Y C (の) J J F (の) I J F (0) I
町 CLR 空のプロジェクト (.NET Framework)	C++	NUnit ラスト プロジェクト (.NET Core) Windows、Linux および MacOS 上の .NET Core で実行できる NUnit テストを含むプロジェク ト。
A1 その10517F	C++	デスクトップ F# Linux macOS デスト Web Windows 「た ダイナミックリンク ライブラリ (DLL) まり行中の複数の Windows アプリ間で共有できる .dll をビルドします。 C++ ライブラリ Windows
		CLR 空のプロジェクト (.NET Core) 開始コアイルを持たず、.NET Core をターゲットとする C++ プロジェクトです。.NET と C++ コード語の相互運用性を提供します。 C++ うイブラリ Windows
		 CLR 空のプロジェクト (.NET Framework)
		CLR コンソール アプリ (.NET Framework) .NET Framework をターゲットとして Windows ターミナルで C++ コードを実行します。.NET と C+ + コード間の相互運用性を提供します。

「次へ (N)」のボタンをクリックします。

	-	
新しいプロジェクトを構成します		
CLR 空のプロジェクト (.NET Framework) ロンソール C++ Windows		
プロジェクト名(N)		
Project1		
場所心		
C:¥Users¥sakaw¥source¥repos •	2	
ソリューション名(M) ()		
Project1		
ソリューションとプロジェクトを同じディレクトリに配置する(D)		
7レームワーク(E)		
.NET Framework 4.7.2		
.NET Framework 4.7.2 -		
	〒ろ(P) が	(の)

「場所(L)」の表示が異なっていると思いますが気にしなくてもいいです。どこにプロジェクト(作るプログラム)を保存するか気にしなければ、デフォルトでもいいですが、右側の四角をクリックすると

📢 プロジェクトの場所						×
\leftrightarrow \rightarrow \checkmark \bigstar sakaw \Rightarrow source \Rightarrow repos		~	Ō	reposの検索		P
整理 マ 新しいフォルダー						?
▲ クイック アクセス ■ デスクトップ ★ ■ ダクンロード ★ ■ ドキュメント ★ ■ ビクチャ ★ ■ jyohou ■ jyohou ■ 特徴 ■ 二歩突き切り戦法 Microsoft Visual S ■ repos ● OneDrive ■ PC	更新日時 検索条件に一致する項	種類	<u><u><u></u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u><u></u></u>			
フォルダー:						
			7:	ォルダーの選択	キャンセノ	ل ا

とプロジェクトを保存するフォルダを指定できます。私は「C++src」というフォルダを作って、 C++のプログラムはそこに保存するようにしています。

→ * ↑ ₩ > PC	> Windows (C:) >		`	0 N	Windows (C:)の検索	
里▼ 新しいフォルダー						EE • (
- 二歩突き切り戦法 ^	名前	更新日時	種類	サイズ	(
Microsoft Visual S	baduk	2020/07/02 11:30	ファイル フォルダー			
repos	C++src	2020/07/11 0:17	ファイル フォルダー			
	FreeSoft	2020/07/08 7:14	ファイル フォルダー			
OneDrive	GAMES	2020/07/09 8:12	ファイル フォルダー			
PC	go	2020/07/01 15:12	ファイル フォルダー			
■ 3D オブミナクト	goengine	2020/06/30 19:36	ファイル フォルダー			
	📊 inetpub	2020/05/14 10:25	ファイル フォルダー			
	Intel	2020/07/10 1:47	ファイル フォルダー			
テスクトップ	JUST	2020/05/12 15:26	ファイル フォルダー			
ドキュメント	PerfLogs	2020/05/15 0:53	ファイル フォルダー			
■ ピクチャ	Program Files	2020/07/11 0:07	ファイル フォルダー			
📕 ビデオ	Program Files (x86)	2020/07/11 0:07	ファイル フォルダー			
	🔄 shogi	2020/07/04 21:11	ファイル フォルダー			
Windows (C)	SWSETUP	2020/05/11 20:34	ファイル フォルダー			
December 1	texlive	2020/05/12 22:52	ファイル フォルダー			
Recovery Image V	texsrc	2020/06/09 13:43	ファイル、フォルダー			
フォルタ	/-: C++src					

「フォルダーの選択」のボタンをクリックします。

新しいプロジェクトを構成します CLR 空のプロジェクト(.NET Framework) コンソール C++ Windows プロジェクトを(N) Project1 場所(L) C¥C++src¥ ・ r* YJコージョンを(M) ● Project1 JUコージョンを力切ジェクトを同じディレクトリに配置する(D) フレームワーク(D) INET Framework 4.7.2 ・			-		\times
CLR 空のプロジェクト (.NET Framework) コンソール C++ Windows プロジェクト名(N) Project1 場所(L) C-¥C++src¥ ・ ・ ・・ ソリューション名(M) ① Project1 」 ソリューションとプロジェクトを同じディレクトリに配置する(D) フレームワーク(D) .NET Framework 4.7.2	新しいプロジェクトを構成します				
プロジェクト名(N) Project1 場所(L) C Y C + + src ¥ ・ ・ ・・ ソリューションを (M) ● Project1 □ ソリューションと プロジェクトを同じディレクトリに配置する(D) フレームワーク(D INET Framework 4.7.2 ・	CLR 空のプロジェクト (.NET Framework) マンソール C++ Windows				
Project1 場所(L) C+YC++srcギ ソリューションズ(M) ① Project1 ソリューションズプロジェクトを同じディレクトリに配置する(D) フレームワーク(D) INET Framework 4.7.2	プロジェクト名(N)				
場所(L) C.¥C++src¥ ・ *** ソリューション名(M) ① Project1 ソリューションとプロジェクトを同じディレクトリに配置する(D) フレームワーク(D .NET Framework 4.7.2 ・	Project1				
C+¥C++src¥ ・ ソリューションを(M) ① Project1 フリューションとプロジェクトを同じディレクトリに配置する(D) フレームワーク(E) .NET Framework 4.7.2	場所(L)				
ソリューション名(M) ① Project1 フリューションとプロジェクトを同じディレクトリに配置する(D) フレームワーク(E) .NET Framework 4.7.2	C:¥C++src¥ v				
Project1 ソリューションとプロジェクトを同じディレクトリに配置する(D) フレームワーク(E) .NET Framework 4.7.2	ソリューション名(M) 🚺				
□ ソリューションとプロジェクトを同じディレクトリに配置する(D) フレームワーク(E) .NET Framework 4.7.2 ~	Project1				
フレームワーク(E) .NET Framework 4.7.2 -					
.NET Framework 4.7.2 •	フレームワーク(E)				
	.NET Framework 4.7.2				
		戻る(<u>B</u>)	作成(Q	
戻3(<u>B</u>) 作成(<u>C</u>)					

「プロジェクト名(N)」をプログラムの内容を表す名前に変えることが出来ます。ここでは「windowsProject1」としてみます。プロジェクト名を設定すると自動的に同じ名前が「ソリューション名」に設定されます。「ソリューション名」を別の名前に設定できますが、あえて変える必要はありません。「ソリューション名」と同じ名前のフォルダが「場所」に指定された所に作成され、そのフォルダ内に必要なソフトやフォルダが作られます。

「作成 (C)」をクリックします。

📢 77/1/10 雑葉D 高売扱 70/227HB どん行動 デバク加 デスト会 分析包 アントロ 記録電配位 9/2/F200 ヘルプ10 物面(Cal-Q) 🔑 WindowsProjecti 1 0 - 0 1 1 - 1 - 1 - 1 - 1 - 1 - 1 Mindows デバッティー 🎢 図 _		🔁 – 🗗	×
скарта - астат ула	リロン クロン 10 70 日 0 10 10 10 0 10 10 10 0 10 10 10 0 10 10 10 0 10 10 10 0 10 10 10 10 0 10 10 10 10 10 10 10 10 10 10 10 10 1	Ø ♥ ₽ = (004) (014)	- a x - a
	701-98919370-9- 7- 70/(74 Window Decisit 70/(20/0	10/1=/	- 9 ×
	2 24 P		
	 こ その他 (名前) ブロジェクトファイル ブロジェクトの依存開係 ルート名前空間 	WindowsProject1 C#C++srcWWindowsPro WindowsProject1	iject1¥Wii
±力			
27xxxx • k = k ≡ f ²	(名前) プロジェクト名を指定します。		
		↑ ソース管理に追加 -	- 10

となります。

上段のメニューの「プロジェクト」の「新しい項目の追加」を選択します。

▶ ファイル(F) 編集(E) 表示(V)	プロジェクト(P) ビルド(B) デバッグ(D)	テスト(S) 分析(N)	ツール(T) 拡張機能(X)	ウィンドウ(W) ヘルプ	(H) 検索 (Ctrl+Q)	Q	WindowsProject1			•	- ø	×
0-0 8-0 8-0	↑ ソリューションの再ターゲット	de	ws デバッガー - 🏿 🎜 🚽								A Live Share	8
0 - 0 2 2 4 3 4 7		c Crid-Shift-A Shift-Aht-A Shift-Aht-A	ex \$7(07) µ (3) €						2012-022-02700-2- ○ ○	i d# [9] ★ (Cri+:) sject1'(1/1)	会 Live Share	φ × ρ.
									ソリューション エクスプローラー プロバティ WindowsProject1 プロジェクト・ マーション その他 (名称) プロジェクトの休存間係 ルート名称記録	しまりスプロ のプロパティ Windows Windows Windows	Project1 WindowsProji Project1	etiWi
出力								* ā X				
出力元(S):	• 2											
									(名前) プロジェクト名を指定します。			
15-一覧 出力						_		 				
[7] 準備完了										↑ ¥-3	御神に追加 -	-

K マケイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) テスト S マ ● ■ ● マ マ ● Debug マ x86	(5) 分析(N) ソール(T) 拡張機能(X) ウ・ ローカル Windows デバッガー * 「「「」」	(ソF ウ(W) ヘルプ(H) 検索(Ctrl+Q)	D WindowsProject1		🖨 – a × 10 line Pare a
	- 20 minute 2003 20 20 -			_	C UNA DAY 1622/10-25- C UNA DAY 1
2-11a	新しい項目の追加 - WindowsProject1			? ×	▶ 11 外部依存關係
5073	▲ インストール済み	並べ替え: 既定 💌 👯 籠	検索 (Ctrl+E)	<i>-</i> م	== y−x JM ル == ヘッダーファイル
	▲ Vezual C++ □-ド 書式設定	C++ ファイル (.cpp)	Visual C++ 1203 Visual C++		## 99-2.374 <i>0</i>
	UI CLR データ リゾース Web ユーディリディ プロパティシート HLSL Nett グラフィックス	πττ C++ 23λ	Visual C++		
	¥7947¥				1/03-030 20270-5+ 5-0 20270-5+ 74042
					プロジェクト ファイル C-VC++srcWindowsProject1WVi
	名前(N): ソースcpp 場所(L): C+V-src#Window	sProject IWVindowsProject IV	- 参照(8)		フロジェクトの依存層係 ルート名前空間 WindowsProject1
	110100		(A)U(2)	* 4 > 4 > 4 >	<
	1 = 1 V				(成前) プロジェクト-4-8指定します。
/7 半備完了					↑ ソース管理に追加 ▲ 🛃
P ここに入力して検索 O III	😑 🚍 🖻 💼 📀	6 🖪 🗃 N			A ← /6 48 A □ 1217 S120/07/11 S2 S120/07/11 S S120/07/11 S

となります。

新しい項目の追加 - Win	ndowsProject1					?	\times
▲ インストール済み		並べ替え	既定 🔹 🚦			検索 (Ctrl+E)	ρ-
 ✓ Visual C++ コード 書式設定 ŪI CLR データ リソース Web ユーティリティ プロパティシート HLSL Test 			Windows フォーム CLR ユーザー コノトロール	Visual	IC++ IC++	種類: Visual C++ 他の Windows コントロールを含む CLR を作成します	77-6
グラフィックス							
▶ オンライン							
名前(<u>N</u>):	MyForm.h						
場所(<u>L</u>):	C:¥C++src¥Windows	Project1¥	WindowsProject1¥		•	参照(B) 追加(A) キャン	セル

上図のように、「Visual C++」の「UI」の「Windows フォーム」を選択し、「追加(A)」をクリッ クします。

D	3 774/61 編集版 表示的 プロ2257/69 20/510 75/5/10 73745 分析60 7-6/0 営務無数0 75/75/00 AU7/10 常産(04-0 - P WeedowsPreject) ○・○ 13・② 副 14 フ・♡・ Debog ・ x86 - ▶ D-15/Windows 75/75・ 月 図 15 十 二 中 止 回 工会 (日 中 二 回 ス つ) 中 二 が 、		Ive Share R ²
4	Myform為(デザイン) + ×	- 0	ソリューション エクスプローラー
~-バ~ エクスプローラー ツールボックス	Writemal (1967) * * • 7-99% All LSG With SG With SG V.S. 9977-000.30% ANE UK TO IJ- #NK#56.00 % AND 55.00 • 1000000000000000000000000000000000000		998-5724 2016-5- 998-5724 2016-5- 998-5724 2017-5-98年(104) 1993-572 2017-5-98年(104) 1993-572 10400-1996 - 1995-592 10400-1996 - 1995-592 - 1
			ソリューション エクスプローラー ゲーム エクスプローラー
			70/(71 ~ 9 ×
			* 101 01 12
	IJ	マ - * # × 15東日秋奈 - 戸・ 274後 侍	E , * <i>C</i>
	13		
	孝儀充了		↑ ソース管理に追加 🔺 🐴

のようなエラーメッセージが表示されます。上段のメニューの「ファイル」の「ソリューションを 閉じる(T)」を

N	771	L(F) 編集(E) 表示(V) プロジェクト	・(P) ビルド(B)	グ(ジ(D) テスト(5) 分析(M) ソール(T) 拡張機能(X) ウインドジ(M) ヘルプ(H) 検索(Ctrl+Q) ク WindowsProjectl		🖨 – ø ×
10		新規作成(N)	•			10 live Share
		糵<(0)	•	• semicondation (1. 1. • • • • • • • • • • • • • • • • •		E occase M
4-	<u>Å</u>	リポジトリのクローン(E)		•	* y	U1-23210270-3- ▼ 4 ×
÷.	æ	29-1-01210(W)			~ 4	○ 🎧 🗄 • 🐻 • ≒ 🖉 🔞 🗠 🏓 💻
127		7-AWARCIALU		込み前に以下のエラーを解決する必要があります。	2	/リューション エクスプローラー の検索 (Ctrl+:) ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
3		:最70(D)	,			VU1=932 WindowsProject1 (1/1 709175) MindowsProject1
	_	閉(j'ð(C)		© <u>エのページが表示されている理由</u>	11	▲ HE 参照
÷	a)	ソリューションを閉じる(T)		SULT #50934-sh800000A)		■ mscorlib
195		Live Share セッションを開始				 System System Data
2		Live Share セッションに参加				 System.Drawing
		MyForm.h の保存(S)	Ctrl+S			System.Windows.Forms
		名前を付けて MyForm.h を保存(A)				▶■ System.Xmi b ── 4LmX你在期低
	۳.	すべく保任(L)	Ctrl+Shift+S	<u> Aindows フォームのデザイン時エラー一般のチェック</u>		▲ 💭 ソースファイル
		ソース管理(R)	,			++ Myform.cpp
		ページ設定(U)				 Myform.h
	۳.	L[18](P)	Ctrl+P			🚚 IJIJ-Ż JPHU
		アカウントの設定(0				
		最近使ったファイル(F)	,			
		最近使ったプロジェクトとソリューション(J)	,			
		MyForm.h の移動先(I)	,			
		終了(X)	Alt+F4			
	_					
					۷	リューション エクスプローラー チーム エクスプローラー
					プ	
					- 12	
					8	2 P /
1	15	-N			×	
	עעע	-9a9全体 • 😧 0I5-	🔥 0 🕾 🖢 🚺 0	1メッセージ 🏹 ビルド・IntellSense - エデー-現在検索 🔎	р -	
		* コード 説明		プロジェクト ファイル 行		
					- 12	
	19	覚 出力			EL.	
	偏完	7				↑ ソース管理に追加 🔺 👫
	1	つ ここに入力して検索		ㅇ 벼 🔒 🖬 💼 💿 🙋 🐧 🚔 刘		On A ■ 12:34 A ■ 12:34 2020/07/11

クリックします。

D① ファイルKF) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デパッグ(D) ※ ○ + ○ 記 + 🎥 🔐 ♡ + ♡ + □ Debug + x85	テスト(5) 分析(N) ソール(T) 拡張機能(X) ウィンドウ(W) ヘルブ(H) ・ ▶ ローカル Windows デバッガー ・ ^{pl} 図 _マ	検索 (Ctrl+Q)	٩	🕚 – O × Lê Live Share 🖉
2 274/40	23.45 5 560 2-2-27 0 33年80 2-27200 3-3791 ▶ 0-25 Wordson 9757 」 ■ 回。 実行する操作を選択してくた 最近期いた項目(R) ■近位世界(A)	19日 IICM-CD ださい 2003/07/11 11:55 2003/07/11 0:25 2003/07/10 1:69 2003/07/10 1:69 2003/07/10 1:69	 	ि – ० × Rituesteer R ००० £ स + ७०६ व ७ । ० ⊁ –
19	Chearcr道道君子道中道品KardSandeKWahittet	2020/07/09 7.29	プロジェクト ファイル (7 原制状態	 • • х <i>P</i> - У№->02.127.00-3- 3-1.127.70-3-
 ロ 単 2ごに入力して検索 O 	H 😄 🚍 💼 💼 🧿 🖉 🖪 🖆			40 () ∧ ← ≪ 48 A [] 1242 2020/07/11 🕎

となります。左側の「最近開いた項目 (R)」の「今日」の一番上にプロジェクト名の「WindowsProject1.sln」がありますから、それをクリックします。

X] アイルの 範囲の 高売10、プロジェアトの 22人内の 81大0/10 81大0/10 81大0/ 72人名 分析00 ソールの 起意発展の クロゲン(M) ヘルプル 10 第三(ローロ)	6) – ⊡ Live Share	×
		The second s	-	-
7	Myformh[97/12] + X - ♥	ソリューション エクスプローラー		- ¶ ×
-パー エクスプローラー ツールボックス		2012-0000000000000000000000000000000		ρ.
		ソリューション エクスプローラー <u></u> チーム エク	770-5-	
		70/(71		≁ q ×
		MyForm System Windows Forms Fo	orm	
		웬 말 안 두 온		
		StartPosition Wine	dowsDefaultLocation	1 ^
		Windowstate Non	mai	-
		BackColor	Control	
		BackgroundImage	(なし)	
		BackgroundImageLayout Tile		
	15	Cursor Defa	ult	- 1
	V/1Va/2dz • 🖸 015- A 0356 🕕 0.5/21-5 🧏 PLE-IntelSence •	ForeColor	ControlText	
		FormBorderStyle Siza	ble	
	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	RightToLeft No		
		RightToLeftLayout False		- 1
		lext MyF	orm	*
		Text		
		コントロールに関連付けられたテキストです		
	神徳??		ソース管理に追加 🔺	

となります。昔の visual studio C++ はこの状態でビルドするとよかったのですが、「ビルド」の「ソリューションのビルド(B)」をクリックすると

C	ペ (0+h2) 素材 (H/L) (M/L) <	ndowsProject1			🖶 – 🗗	×
Morena (1970) = x				Y11-74/2 12/2 12/3 C ← M = 1 → 0 → 0 → 0 → 0 → 0 → 0 → 0 → 0 → 0 →	10 00 00 00 00 00 00 00 00 00 00 00 00 0	ب م ب
				ソリューション エクスプローラー デー プロパティ Myform System.Windows.Fo	-ムエクスプローラー ・ irms.Form	4 ×
				2 말 한 후 P StartPosition WindowState B 중재 BackColor BackgroundImage BackgroundImage	WindowsDefaultLocation Normal	*
13〒 ソリュージン25日 - ◎ 1125- ▲ 0100 ◎ 00/0世-ジ ▼ 25F - IntelSence - ▼ 3-F 12時 ● INXTS01 22/9- ポインを定意しなけれなりません。	לעובי Wioław	► ⊃т√J↓ Project1 LINK	- 中 エラー-知名独衆 行 抑制伏聴 1	sex.sprouhalimage.syrout Cursor D = ForeColor ForeColor ForeColor ForeColor ForeColor Text Zyhout Stream Zyho	Default MS UI Gothic, Spt Control Text Sizable No False MyForm 21/C#1.	ļ
17**** 第 277 日 华佳先?					↑ ソース管理に追加 🔺	-

のようにエラーメッセージが表示されます。このフォームを実行時に作成して表示するためのプロ グラムを用意する必要があります。

上段のメニューの「プロジェクト」の「新しい項目の追加」を選択します。

新しい項目の追加 - Wir	ndowsProject1					?	\times
▲ インストール済み		並べ替え	: 既定 🔹 👔 📰		検索 (Ctrl+E)		<i>.P</i> -
✓ Visual C+・ コード 書式設定 UI CLR データ リソース Web ユーティリティ プロパティシート HLSL Test グラフィックス			C++ ファイル (.cpp) ヘッダー ファイル (.h) C++ クラス	Visual C++ Visual C++ Visual C++	程類: Visual C++ 1203		
▶ オンライン							
名前(<u>N</u>):	ソース.cpp						
場所(<u>L</u>):	C:¥C++src¥Windows	Project14	WindowsProject1¥	•	参照(B)	+1111	7 11.
					3 <u>2</u> /JI(<u>A</u>)	4771	- 40

の画面で、今度は、「Visual C++」の「コード」の「C++ファイル (.cpp)」を選択し、

新しい項目の追加 - Windows	vsProject1	?	×
▲ インストール済み	並べ替え: 既定 ▼ 詳 ! ! ! 検索 (Ctrl+E)		P -
 ✓ Visual C++ □-ħ = ₫式設定 UI CLR デ-タ リソース Web ユーティリティ プロパティ シート HLSL Test グラフィックス ▶ オンライン 	 		
名前(N): ソーフ 場所(L): C:¥C	·ス.cpp IC++src¥WindowsProject1¥WindowsProject1¥ □ 参照(B) 追加	ם(<u>A)</u> キャン	セル

「追加(A)」をクリックします。

📢 ファイル(D) 編集(E) 表示(U) プロジェクト(D) ビルド(B) デバック(D) テス	NS 分析(N) ツール(D) 拡張機能(X) ウインドウ(M) ヘルプ(H) 検索(Ctrl+Q)	P WindowsProject1		🛑 – a ×
🕴 🖸 - 🔘 🛐 - 🏠 🔛 🔐 🔊 - 🤆 - 🛛 Debug - 🕺 x86 - 🛛	• ローカル Windows デバッガー • 🏓 📾 🚎 🖿 🌾 🗍 🕲 👘 🗍 🖉 🗍 📮			ഥ Live Share 🖉
学 <mark>ソースcpp + ×</mark> MyForm.h (デザイン)			- 0	ソリューション エクスプローラー
10% •	 (ワレーCD-23-7) ビンド・IntelSense ・ 	• 70/221- WindowPojet1 LINK	・ ① ・ ① ・ ① ・ ① ・ ① ・ ① ・ ① ・ ② ・ ② ・ ② ・ ② ・ ② ・ ③ ・ ③ ・ ③ ・ ③ ・ ③ ・ ③ ・ ③ ・ ③	y12-592 12270-5- ● ● 3 ● ● ● ● ● ● ● ● ● ● ● ● ● y13-592 12270-5-0000 (0+0) ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●
15				
□ 神儀充了				↑ ソース管理に追加 🔺 🐴

と「ソース.cpp」が作られ、何も書いてない空のファイルが表示されます。



のように

#include "MyForm.h"

```
using namespace WindowsProject1;
```

```
[STAThreadAttribute]
int main() {
    MyForm^ fm = gcnew MyForm();
    fm->ShowDialog();
    return 0;
}
```

と打ち込みます。ここで、

using namespace WindowsProject1;

の WindowsProject1 はプロジェクトの名前です。 「ビルド」の「ソリューションのビルド(B)」をクリックすると

×	】 ファイルビ 編集(E) 表示(M) プロジェクト(E) どルド(B) デパック(D) ◎ • ◎ 13 • 🖕 🔐 🄊 • 🤆 • Debug • x86	テストS 分析(M) ツール① 拡張機能(M) ウインドウ(M) ヘルプ(H) - ▶ ローカル Windows デバッカー - 声 回 - h (作 当 注 ■ *	技策(Ctri+Q) ア WindowsProject1	Ive State	×
サーバー エクスプローラー・ツールボックス	VPS200 + K (Micros)(STA) Vestorship() Vest	• (90-10,33-7)	• 0 main)	P 2/25-2022 25/25-2- P 2/25-2022 25/25-2- P 2/25-2023 25/25-2- P 2/25-2025 P 2/25-202 P	- α × ρ -
				ソリューション 20スプローチョー プロパティ 国際 日本 (多)	* 7 ×
	10% 0 数単な見つかりおんてした 立方 立方の回してなど した。一部 した。一部 に、した。可能は、プロジェングト1 NickowProject1, 単元で、 Balag した。一次の一次の一次の一次の一次の一次の一次の一次の一次の一次の一次の した。ここので、 した。ここので、 した。 とので、 した、 した。ここので、 した。 した。ここので、 した、 した。 した。ここので、 した、 した。 した。 した。 した。 した。 した。 した。 した。	Not		- 6:10 2;R2 p7 ctar - 0:2 X	
	17-一覧 <u>出力</u> ビルF正常終了				-{0

のように、正常にビルドできます。

「デバッグ」の「デバッグの開始(S)」をクリックすると

20】 アイ化的 製紙店 高売10 プロジスクト的 24A-10 デスト協 分析10 アスト協 分析10 0 アメ化内 製造機能の クレビカの A4プ(h) 地名Cal-Q タ Weedwark-spect ● ● ● □ 1 ● ● ● ● ア ● Enco = ● 総合・ ● 総合・ ● ● 総合・ ● ● 目 ● ● ● さ で 1 ● 夏 ● ● 「「」 ■ ● ● ● さ で 1 ● 夏 ● ● 「」 ■ ■ ● ● ● さ で 1 ● 夏 ● ● ○ □ ■ ● ● ● ○ □ ● ● ● ○ □ ● ● ● ● ● ● ● ● ● ●	🕒 – 🛛 × <u>B</u> Live Share 🕅
	Excylo ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
100 % • の 数数は見つかりませんでした 〈) 行いの 文象と タブ CRLF	
48	+ i ×
	· 言語 4
	A 11-70070-1044
	12.39

のように、フォームが表示されます。黒い画面も一緒に表示されますが気にしなくてもいいです。 随分、手間がかかりましたが、何もないフォームを表示することが出来ました。

これまでの手続きは魔法使いの儀式だと思って、以下には古い visual studio C++ の使い方を解説しているので、以下のように、フォームに「ツールボック ス」にあるツールを配置して、プログラミングをすればいいです。

後は以下のようにプログラミングすれば良いです。VC++ 2010 と VC++ 2013と VC++ 2019 に共通です。 マウスで Form1 を適当な大きさにする。



ツールボックスのメニューが無ければ、表示のメニューから

∞ en2 - Microsoft Visu	al Studio Academic		
ファイル(E) 編集(E)	表示(Y) プロジェクト(P) ビルド(P	A) デバッグ(D) チーム(M) データ(A)	書式(Q) ツール(I) テスト(S) ウィンドウ(W) ヘルプ(H)
🔂 • 🖽 • 😂 🖼 🤅	□-ド(<u>C</u>)	Ctrl+Alt+0	- 🛛 😼 loseing - 🚽 🖓 🕾 🞲 🖄
傳	□ デザイナー(□)	Shift+F7	
Form1.h [デザイン]*	💀 ソリューション エクスプローラ	-(P) Ctrl+Alt+L	マプロパティ 検索と置換 ×
	📑 チーム エクスプローラー(<u>M</u>)	Ctrl+¥, Ctrl+M	「換 フォルダーを指定して検索・
P Form1	ブックマークウィンドウ(B)	Ctrl+K, Ctrl+W	
	同: 呼び出し階層(日)	Ctrl+Alt+K	検索する文字列(N):
	🖳 クラス ピュー(<u>A</u>)	Ctrl+Shift+C	-
	□ード定義ウィンドウ(D)	Ctrl+Shift+V	検索対象(L):
	オブジェクト ブラウザー(1)	Ctrl+Alt+J	ソリューション全体 👻 🔐
	👶 エラー一覧(I)	Ctrl+¥, E	マリサブ フォルダーを含める(B)
	□ 出力(Ω)	Alt+2	□ 検索オブション(0)
	フソースビュー(B)	Ctrl+Shift+E	
	🛃 スタート ページ(<u>G</u>)		+ 結果オプション(S)
	ジールボックス(L)	Ctrl+Alt+X	
	検索結果(<u>N</u>)	•	ৰ না ব
	その他のウィンドウ(E)	•	
	ツールパー(工)	•	
	全画面表示(U)	Shift+Alt+Enter	
	≧! タブオーダー(良)		
	🗐 前に戻る(B)	Ctrl+-	
	回、次に進む(E)	Ctrl+Shift+-	
	次のタスク(区)		
🔓 エラー一覧 🗖 コマ	前のタスク(<u>R</u>)		
34+ 0++	プロパティマネージャー(P)		+15.15 ± ¹³ 420 × 412

ツールボックスを選択して、ツールボックスを表示する。

👓 en2 - Microsoft Visual Studio Academic	
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) データ(A) ツール(T) テスト(S) ウィンドウ	(W) ヘレプ(H)
💱 • 🛅 • 📷 🛃 😹 🐁 🕰 😤 🧉 • 🖓 • 🖓 • 🖏 🕨 Debug 🔹 Win32 🔹 💋 loseing	- 🖓 🕾 🖬 😒 🎌 📜
神 ゆる 豆の四 豆() 塩()	
Form1.h [デザイン]* × スタートページ ソリューション エクスプローラー ・	ツールボックス × =
	MonthCalendar ^
P Form1 C D X	
	NumericUpDown
	PictureBox
	🚥 ProgressBar
	 RadioButton
	RichTextBox
	abl TextBox
	ban ToolTip
	TreeView
	WebBrowser
	▲ コンテナー
	ポインター
	FlowLayoutPanel
	[^{xv}] GroupBox
	Panel
	SplitContainer
	TabControl
	TableLayoutPanel
	▲ メニューナッリール, パー ▼
🙀 エラー一覧 🖬 コマンド ウィンドウ 🗉 出力	
準備完了	

ツールボックスにある PictureBox をクリックし、マウスを Form1 に移動し、ドラッグして、Form1 の中央に Picture Box を配置する。

😎 en2 - Microsoft Visual Studio Academic						x
ファイル(F) 編集(E) 表示(V) プロジェクト	-(P) ビルド(B) デバッグ(D) ヲ	チーム(M) データ(A)	書式(O) ツール(T) テ	スト(S) ウィ	ィンドウ(W) ヘルプ(H)	
	(~ - 윤 - B) 🕨 Debug	- Win32	- 🙆 loseing		- 🔜 🕾 🖬	€ 🗒
	· · · · · · · · · · · · · · · · · · ·	*				
Form1.h [デザイン]* × スタートページ	ソリューション エクスプローラ	-		- 2	ソールボックス ×	₹
€ Form1					MonthCalendar MonthCalendar NotifyIcon NumericUpDown PictureBox PictureBox PictureBox RadbButton RadbButton RathTextBox ToolTip TreeVlew WebBrowser Jンテナー Knf1×/ター ForuLayoutPanel GroupBox Count	E
					SplitContainer TabControl	
					TableLayoutPanel	
					 メニューナツールパー 	-
□ ユラー一覧 ■ コマンド ウィンドウ ■ 出 準備完了	מ		125 , 34	1	<u>∓</u> वि334 × 300	.al

プロパティが表示されていなければ

👓 en2 - Microsoft Visu	ual S	udio Academic	General Street	With the second					
ファイル(F) 編集(E)	表示	(V) プロジェクト(P) ビルド(B) デ	バッグ(D) チーム(M)	データ	r(A) ツール(T) テスト(S) ウィ	ンドウ(V	V) ~レフ	'(H)	
🔂 • 🖼 • 😂 🛃 🖉	F	コード(<u>C</u>)	Ctrl+Alt+0		- 🖄 Solver		•		😤 🍱 🖄 🖓 🗒
🗊 🕸 🏊 🔺 👘		デザイナー(<u>D</u>)	Shift+F7						
ソリューション エクス	-2	ソリューション エクスプローラー(<u>P</u>)	Ctrl+Alt+L				プロパティ		- 4 ×
en2::Form1		チーム エクスプローラー(<u>M</u>)	Ctrl+¥, Ctrl+M	ht(v	pid)	-	Initializ	eComp	onent VCCodeF -
		ブックマーク ウィンドウ(旦)	Ctrl+K, Ctrl+W		コマンド ウィンドウ(C)	Ctr	+Alt+A		
ť	13	呼び出し階層(日)	Ctrl+Alt+K		Web ブラウザー(W)	Ctr	+Alt+R		
i i	23	クラス ビュー(A)	Ctrl+Shift+C	-	マクロ エクスプローラー(M))	InitializeCompo
1 2		コート定義ワイントワ(世)	Ctrl+Shitt+V		ソース管理エクスプローラー(S)				private
t t	100	FE(1)	CUTHAILTS		ドキュメント アウトライン(D)	Ctr	+Alt+D	erride	d-¥iyugyou¥en2
t t	-0	エラー 夏(王)		1	履歴(I)			ne	en2::Form1::Ini
t t		LIV-7 ピュー(R)	Ctrl+Shift+F	53	保留中の変更(旦)			tant	False
t		スタートページ(G)	Carromere	1	プロバティ ウィンドウ(<u>W</u>)	Alt	+Enter	ted	False
ť	SP.	ツールボックス(L)	Ctrl+Alt+X	-	サーバー エクスプローラー(⊻)	Ctr	+Alt+S	loade	True
100 % - 1		検索結果(<u>N</u>)		2	タスク一覧(<u>K</u>)	Ctr	l+¥, ⊤	ed	False
出力		その他のウィンドウ(E)	•	F	F# Interactive	Ctr	+Alt+F	ed	False
出力元の表示(S):		ツール バー(I)	•				IsTer	nplate	False
		全画面表示(U)	Shift+Alt+Enter				IsVin	tual	False
	P	前に戻る(旦)	Ctrl+-				Type	Impiem String	raise
		次に進む(E)	Ctrl+Shift+-					oung	Tota
		次のタスク(区)					L++		
		前のタスク(<u>R</u>)							
		プロパティ マネージャー(E)					רם ל 🚰	(ティ	や ツールボックス
進備完了		プロパティ ページ(Y)			71 行 9 列		3 文字		挿入。言

または、ツールボックスの欄の右端のアイコンを押して

	• Win32	• 🙆 loseing		· · · · · · · · · · · · · · · · · · ·	¢ -	delle .	
	,			MonthCalendar	*	ツールボックス プロパティ	-
				NumericUnDown	13	検索と置換	- Cale
			12	PictureBox		F	
				ProgressBar		-	100 m
			۲	RadioButton			1 1 A 10
			- A	RichTextBox	-		E Colorest
			obl	TextBox			100 年4月1日日
			S	ToolTip			- Section and
6			10	TreeView			and the second
				WebBrowser	- 1		No.
				ガインター	- 1		- 3949 4
			T	Flowd avoutPanel			States - and
			(XY)	GroupBox			
				Panel			
				SplitContainer			and the second
			-	TabControl			
				TableLayoutPanel			
			- ×=	ニューとツール バー			
			•	ポインター	*		
一覧 📁 コマンド ウィンドウ 💼 出力							-

で、プロパティを探し、プロパティを表示する。

∞ en2 - Microsoft Visual Studio Academic		No. of Concession, Name	
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D)	チーム(M) データ(A) ツール(T) テ	Fスト(S) ウィンドウ(W)	ヘルプ(H)
	* Win32	loseina	- I I
Form1.h [デザイン]* x スタートページ ソリューション エクスプロー	-5-		
	n in the second s	pict	tureBox1 System.Windows.F -
Form1			2↓ 🔲 🖋 🖾
			Size 334, 300 ^
		4	非同期
		⊳	ErrorImage 🔀 System.Dra
		D	InitialImage System.Dr.
			WaitOnLoad False
			表示
			BackColor Control
			BackgroundIr (なし)
			BackgroundIr Tile
			Cursor Default
			Image (なし) …
			UseWaltCursc False
		12	(一ジの選択
0		Terry	
		Pict	ureBox に表示されるイメージ
	•		
🎏 エラー一覧 📭 コマンド ウィンドウ 🛢 出力			
准備完了			

プロパティで、雷のアイコンをクリックして、イベントを表示する。



一番下にある paint をダブルクリックする。

👓 en2 - Microsoft Visual Studio Academic				
ファイル(E) 編集(E) 表示(V) プロジェクト	<u>P)</u> ビルド(<u>B</u>) デバッグ(<u>D</u>) チーム	(M) データ(A) ツール(I)	テスト(5) ウィンドウ	9(<u>W</u>) ヘルプ(<u>H</u>)
🛛 🖥 • 🗁 • 🚰 😹 🖉 🖌 🖦 🏙 🤊 • 0	- 📮 - 📑 🕨 Debug -	Win32 •	loseing	- 🖓 🕾 🚮 🖄 🎌 🗒
■ № № № 個 谭 谭 ■ 월 ■ ₽	G 🖗 🖨 🎝 🖓 =			
Form1.h [デザイン]* x スタート ページ	ソリューション エクスプローラー			Form1.h* × プロパティ =
Porm1				% en2::Form1 • • s, aronm1.:pictureBox1.•
				1))->EndInit();
	0			E∷PaintEventArgs^ e) {
0	6			
				100 % • •
🧞 エラー一覧 🖬 コマンド ウィンドウ 目 出け 準備完了		7	9行 115列	112文字 挿入

必要なら Form1.h をドラッグして、左側に移す。

🛥 en2 - Microsoft Visual Studio Academic		198-1				and the second second second	
ファイル(E) 編集(E) 表示(V) プロジェクト(P) ビ	ビルド(<u>B</u>) デバッグ(<u>D</u>)	チーム(M) データ(A)) ツール(I)	テスト(<u>5</u>)	ウィンドウ((W) ヘレプ(H)	
in	Debug	- Win32	-	10seing		- 2	🗢 🖬 🔿 📯 🗌
						11 -*	
						151-25-512-940(1552)1050252200	
Form1.h [デザイン]* スタート ページ ソリ	ューション エクスプロー	ラー Form1.h*	×		<u> </u>	プロパティーメ	–
🎕 en2::Form1	- 😽 pictu	reBox1_Paint(System	::Object ^ se	nder, System	::Wine +	pictureBox1_Pa	aint VCCodeFunc -
s, aroninpretureboxi_Paint),					÷	8= 41 (63	
					<u>^</u>	▲ C++	
						(Name)	pictureBox1 F
						Access	private
						CanOverride	False
Security and a state						File	d:¥jyugyou¥en
1))->EndInit();						FullName	en2::Form1::p
						IsConstant	False
						IsInjected	False
⊨::PaintEventArgs^ e) {						IsInline	True
						IsOverloaded	False
t						IsSealed	False
					E	IsShared	False
						IsTemplate	False
						IsVirtual	False
						Mustimpieme	False
						C++	System::void
					-	cri	
100 % - 4		"	и		•		
🙀 エラー一覧 🖿 コマンド ウィンドウ 📄 出力							
準備完了			79	9 177	115 列	112 文字	挿入

Form1.h の一番下にある



}

の { と } の間に、次のようになるようキーボードから打ち込む。

🐱 en2 - Microsoft Visual Studio Academic	San Andrew States	
ファイル(E) 編集(E) 表示(Y) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) データ(A) ツール(I) テスト(S) ウィンドウ	r(W) ヘルプ(H)	
🛐 • 🛅 • 🖉 🚽 🖓 🙏 🐁 🏨 🖉 • 🔍 • 🖓 • 🔍 • 💭 • 🖄 🕨 Debug 🔹 Win32 🔹 💋 loseing	- 🗠	l 🕾 🖬 🗠 🗶 🍹
[昆池温水临] 建建 [Ξ일] □ 🖓 Ϥ 🏟 및 👷		
Form1.h [デザイン]* スタートページ ソリューション エクスプローラー Form1.h* x	プロパティ ×	=
	pictureBox1_Pa	aint VCCodeFunc -
// Form1	81 21 🖾	
	▲ C++	<u>^</u>
this->AutoScaleDimensions - system:.Unidows::Forms::AutoScaleMode::Font;	(Name)	pictureBox1_F
this->ClientSize = System::Drawing::Size(404, 374);	Access	private
this-Name = L'form ';	CanOverride	False
this->Text = L"Form";	File	d:¥jyugyou¥en
this->ResumeLayout (false):	FullName	en2::Form1::p
	IsConstant	False
#pragma endregion	IsInjected	False =
private: System::Void_pictureBox1_Paint(System::Object^sender, System::Windows::Forms::PaintEvent	IsInline	True
$araphics g = e^{-\lambda araphics};$ Pen^ pen = gcnew Pen(Chicr::Blue);	IsOverloaded	False
double pi = Math::PI;	IsSealed	False
$g \rightarrow Vraw E I i p se (pen, 50, 50, 260);$ for (double t=0: t <= pi, 260)	IsShared	False
z-DrawLine(pen, (int)(180+130≭cos(t)), (int)(180-130≭sin(t)),	IsTemplate	False
(int)(180+130*cos(2*t)), (int)(180-130*sin(2*t)));	IsVirtual	False
	MustImplem	False
· ↓ };	TypoString	Suctom: Woid *
	C++	
	1.	
100 % - <		
😤 エラー一覧 🖿 コマンド ウィンドウ 🗉 出力		
IntelliSense: 'C++/CLI で使用できません' 87 行 19 列	7 文字	挿入!

これは picturebox1 のイベント paint が発生したとき、どのように反応すべきかを指示するプロ グラムを作ることになる。

```
private: System::Void pictureBox1_Paint(System::Object^ sender,
        System::Windows::Forms::PaintEventArgs^ e) {
    Graphics^ g = e->Graphics;
    Pen^ pen = gcnew Pen(Color::Blue);
    double pi = Math::PI;
    g->DrawEllipse(pen, 50, 50, 260, 260);
    for (double t=0; t<2*pi; t += pi/80) {</pre>
```





#include <math.h>

を追加する。

👓 en2 - Microsoft Visual Studio Academic	and the second se			
ファイル(E) 編集(E) 表示(⊻) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) データ(A) ツーク	レ(I) テスト(S)	ウィンドウ(<u>W</u>)	ヘルプ(日)	The second s
🛅 • 🛅 • 😂 🛃 🕔 🕹 🤹 🎕 🔄 • • • - 💭 • 🖏 🕨 Debug 🔹 Win32	- 🖄 loseing	1	- -7 1	2 🖬 🖄 🛠 🗒
[□] \$2 \$2 * 10 課 課 □ 2 □ 2 3 4 2 3 2 3 3 9 .				
Form1.h [デザイン]* スタート ページ ソリューション エクスプローラー Form1.h* ×		- JC	パティ x	
(グローバレスコープ)・		-		-
#include <math.h></math.h>		+	21 100	
□#pragma once				
Enamespace en2 {		-		
using namespace System: using namespace System::ComponentModel: using namespace System::Dollections; using namespace System::Didows::Forms; using namespace System::Data; using namespace System::Drawing;				
日 /// <summary> /// Form O 祝要 /// /summary> 日 public ref class Form1 : public System::Windows::Forms::Form</summary>				
public: Forml(void)				
InitializeComponent(); //T000: ここにコンストラクター コードを追加します //				
protected: /// <summary></summary>		-		
100 % - 4		•		
🙀 エラー一覧 📼 コマンド ウィンドウ 🗉 出力				
IntelliSense: 'C++/CLI で使用できません'	1行	18 列	18 文字	挿入

コンパイルし、実行する(Debug の左の赤い三角形のアイコンをクリックする)と



となる。図が欠けていれば、Form1.h[デザイン] のタブをクリックし、Form や PictureBox を 適当な大きさに修正し、再度 Debug の左の赤い三角形のアイコンをクリックする.。

```
private: System::Void pictureBox1_Paint(System::Object^ sender,
    System::Windows::Forms::PaintEventArgs^ e) {
    Graphics^ g = e->Graphics;
    Pen^ pen = gcnew Pen(Color::Blue);
    double pi = Math::PI;
    g->DrawEllipse(pen, 50, 50, 260, 260);
    for (double t=0; t<2*pi; t += pi/80) {
       g->DrawLine(pen, (int)(180+130*cos(t)),
            (int)(180-130*sin(t)),
                    (int)(180+130*cos(2*t)), (int)(180-130*sin(2*t)));
    }
}
```

は、pictureBox1_Paint という名前から分かるように、PictureBox1 を描くときにこの関数が実 行されます。この関数の

Graphics[^] g = e->Graphics;

は、グラフィックスを描くためにデバイスgを設定しています。PictureBoxを使って、自動的に グラフを描きたいときはいつでもこうします。魔法を使うための呪文です。ボタンを押せばグラフ を描くようにしたいときは、次の例を見て下さい。

Pen^ pen = gcnew Pen(Color::Blue);

は、線を描くときに使うペンを作成しています。色は Blue、線の太さは1ピクセルです。

Pen^ pen = gcnew Pen(Color::Red, 2);

とすれば、色は Red、線の太さは2ピクセルです。これも魔法を使うための呪文です。色(Black, Yellow, Green, White, …)と線の太さだけ変えて使って下さい。

double pi = Math::PI;

は、実数の変数 pi を円周率 Math::PI と定義しています。円周率は Math::PI とすれば使えるように VC++ で準備してくれています。 Math::PI といつも書くのは面倒なので pi という実数の 変数を準備しました。

g->DrawEllipse(pen, 50, 50, 260, 260);

は、g->DrawEllipse() が楕円を描けという命令です。g は上で定義したグラフィックスを描く ためにデバイスで、線分や矩形や楕円を描きたいときに g->kannsuu() という形で使います。 g->DrawEllipse() の引数の最初の pen は上で生成したペンです。今の場合、色は Blue、線の太 さは1ピクセルで線を描きます。次の 50, 50 は矩形の左上隅の座標を指示しています。次の 260 は矩形の幅を指示しています。次の 260 は矩形の高さを指示しています。つまり、50, 50, 260, 260 で、左上隅の座標が (50, 50) で、幅が 260、高さが 260 の矩形を表しています。

g->DrawEllipse(pen, 50, 50, 260, 260);

は、この矩形に内接する楕円を描けという命令です。今の場合、矩形が正方形なので、円を描きま す。言い換えると、中心が (180, 180)、半径が 130 の円です。

```
for (double t=0; t<2*pi; t += pi/80) {
    g->DrawLine(pen, (int)(180+130*cos(t)),
        (int)(180-130*sin(t)),
        (int)(180+130*cos(2*t)), (int)(180-130*sin(2*t)));
}
```

で、上の円周(中心が (180, 180)、半径が 130)上の角度が t と 2t の 2 点を結ぶ線分を t が 0 か ら 2*pi まで pi/80 だけ増やしながら描いています。

```
for (double t=0; t<2*pi; t += pi/80) {
}</pre>
```

は、 for 文で、4つの部分に分かれていて、まず (double t=0; で、初期設定: 実数変数 t を 0 と セットしています。この部分は最初に一度だけ実行されます。次に t<2*pi; の部分で条件式を書 き、 t<2*pi が真 (true) であれば、 ・・・ の部分を実行します。次に t += pi/80) の部分を実行 します。普通はこの部分に変数の再セットをする命令を書きます。今の場合 t += pi/80 は t に pi/80 を加えるよう指示しています。t += pi/80 は t = t + pi/80 と書いても良いです。そして、 2番目の t<2*pi の判定を再びします。これが真 (true) であれば、 ・・・ の部分を実行します。次 に t += pi/80) の部分を実行します。これが真 (true) であれば、 ・・・ の部分を実行します。と繰 り返します。2番目の t<2*pi の判定が偽 (false) であれば for 文の実行は終了です。

即ち、for 文は

```
for (初期設定; 条件式; 後処理) {
命令の並び
```

}

の形式で、まず、初期設定を実行し、条件式が真であれば、命令の並びを実行し、後処理をし、条 件式が真であれば、命令の並びを実行し、後処理をし、を条件式が偽になるまで繰り返します。条 件式の一回目の判定が偽であれば、命令の並びは一回も実行されません。

は、g->DrawLine()が線分を描くための命令です。引数の pen は線を描くためのペンを指定していま す。次の (int)(180+130*cos(t)), (int)(180-130*sin(t)), で始点の座標、次の (int)(180+130*cos(2*t)), (int)(180-130*sin(2*t)) で終点の座標を指示しています。(int)(...) は (...) を整数値に丸めて欲し いという指示です。180+130*cos(t) は double の値を取りますが、g->DrawLine() には、2番目 以降の引数に int を取るものと float を取るものがあり、どっちを使いたいのか分からないと Error が出るので、180+130*cos(t) を int に変換して欲しいと (int)(180+130*cos(t)) の形式で指示して います。面倒ですがこれが仕様です。こうするものだと思って下さい。

(int)(180+130*cos(t)), (int)(180-130*sin(t))の180は円の中心が(180,180)だからで、130は円の半径が130だからです。(int)(180-130*sin(t))の-はPictureBox1の座標系と数学で我々が使っている座標系が上下逆だからその修正をするためです。ここで、cos()とsin()という関数を使っているので、cos()とsin()という関数を使っていることをVC++に教えるために、Form1.hの最初に、

#include <math.h>

と打ち込む必要があります。関数を使うためには、前もってその関数を宣言または定義しておく必要がありますが、cos()と sin()といった数学関数は math.h というファイルで宣言されているので、include 文で math.h を呼び込んでいます。この部分を

```
g->DrawLine(pen, (int)(180+130*Math::Cos(t)),
  (int)(180-130*Math::Sin(t)),
  (int)(180+130*Math::Cos(2*t)),
  (int)(180-130*Math::Sin(2*t)));
```

としておけば、多分、namespace の Math で定義しているはずで、VC++ は Math::PI や Math::Sin() や Math::Cos() がどのようなものか知っていて、特別な宣言が必要なくて、

#include <math.h>

のような include 文は必要ありません。





更に、

```
g->DrawLine(pen, (int)(180+130*Math::Cos(t)),
  (int)(180-130*Math::Sin(t)),
  (int)(180+130*Math::Cos(3*t)),
  (int)(180-130*Math::Sin(3*t)));
```

と変更すれば、





```
となります。
```

更に、

```
g->DrawLine(pen, (int)(180+130*Math::Cos(2*t)),
    (int)(180-130*Math::Sin(2*t)),
    (int)(180+130*Math::Cos(5*t)),
    (int)(180-130*Math::Sin(5*t)));
```

と変更すれば、





となります。色々思いついたことをやってみるのが勉強になります。自分で見付けたことは宝物 になります。

Python というプログラミング言語と Pygame というパッケジを使うと簡単にアニメーションが 出来ます。

```
import pygame
from math import *
pygame.init()
black = (0,0,0)
white = (255, 255, 255)
green = (0, 255, 0)
red= (255, 0, 0)
size = (700, 500)
screen = pygame.display.set_mode(size)
pi = atan(1.0)*4
done = False
clock = pygame.time.Clock()
t = pi
while done == False:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done = True
    screen.fill(white)
    x = 0
```

```
while x <= 2*pi-2*t:
    pygame.draw.line(screen, green, \
        [200+100*cos(x), 200-100*sin(x)],\
        [200+100*cos(2*x), 200-100*sin(2*x)], 1)
        x += pi/40
    t -= pi/40
    pygame.display.flip()
    if t < 0:
        t = pi
    clock.tick(1)
```

```
pygame.quit()
```



この講義では時間がありませんから、Python の入門講座はしませんが、興味があれば、 http://www.youtube.com/playlistlist=PL1D91F4E6E79E73E1?

を見て下さい。 Python の基本の基本と簡単なアーケードゲームの作り方が youtube で 15 時間ぐ らいで説明されています。英語ですが、見ていれば何をしているか分かるはずです、また、

http://ProgramArcadeGame.com

には on-line のテキストがあります。アメリカの Simpson College の Dr. Paul Vincent Craven 氏 の講義録です。更に、Linux でのものですが、Pygame や Python の説明が youtube に沢山あり ます。

勿論、このようなプログラムを Timer を使って Visual C++ や C++ Builder で作れますが、こ の講義のレベルを超えています。複雑なアニメーションやゲームを作るには Visual C++ や C++ Builder で作りますが、このような単純なものは、イギリスで開発された子供の教育用の手のひら に乗るコンピュータ Raspberry Pi の言語としても採用されている Python で作る方が簡単です。 Python はオブジェクト指向言語です。

別のプログラム (関数 y=exp(x) を表示する) を作ってみます。

各プログラムごとに1つのプロジェクトを準備します。1つのプロジェクトに複数のプログラム を追加してはいけません。従って、上のプロジェクトに新しいプログラムを追加してはいけませ ん。最初から、上と同じ手順を踏みます。
「スタートページ」をクリックします。「スタートページ」のタグが見あたらなければ、「表示」 メニューの「スタートページ」をクリックします。



「新しいプロジェクト … 」をクリックします。

新しいプロジェクト				S ×
最近使用したテンプレー	-ト	.NET Framework 4 ▼ 並べ替え基準: 既定	• 111 111	インストールされたテンプレートの検索 🔎
インストールされたテン ✓ Visual C++ ATL CLR 企般 MFC テスト Win32 P 他の言語 その他のプロジェクト テータペース テータペース テスト プロジェクト オンライン テンプレー	ンプレート トの種類 、 ト	Win32 コンソール アプリケーション MFC アプリケーション Win32 プロジェクト 空のプロジェクト ATL プロジェクト MFC DLL Windows フォーム アプリケーション CLR コンソール アプリケーション CLR コンソール アプリケーション マの CLR プロジェクト	Visual C++ × Visual C++ ×	種類: Visual C++ Windows ユーザー インターフェイスを 含むアプリケーションを作成するための プロジェクトです。
名前(<u>N</u>):	<名前を入力し	てください>		
場所(<u>L</u>):	D:¥jyugyou¥		•	参照(<u>B</u>)
ソリューション名(<u>M</u>):	<名前を入力し	てください>		┃ ソリューションのディレクトリを作成(₽)
				ソース管理に追加(U)
				OK キャンセル

Windows フォームアプリケーションを選択し、名前 (N): のエディットボックスに例えば exp と 入力し、

Fしいプロジェクト			an andreas a cariste	Manherin III	S × ×
最近使用したテンプレー	-ト	.NET Framework 4 🔹 並べ替	記基準: 既定	• III III	インストールされたテンプレートの検索 🔎
インストールされたテン ✓ Visual C++ ATL CLR 全般 MFC テスト Win32 ▷ 他の言語 ▷ その他のプロジェク ▷ テスト プロジェクト オンラインテンプレー	>プレート トの種類 ト	Win32 コンソール アプリ ●	リケーション パリケーション ケーション	Visual C++ Visual C++ Visual C++ Visual C++ Visual C++ Visual C++ Visual C++ Visual C++	種類: Visual C++ Windows ユーザー インターフェイスを 含むアプリケーションを作成するための プロジェクトです。
名前(<u>N</u>):	exp				
場所(<u>L</u>):	D:¥jyugyou¥		-	参照(<u>B</u>)	
ソリューション(<u>5</u>):	新しいソリュー	ションを作成する	•		
ソリューション名(<u>M</u>):	exp		☑ ソリューションのディレクトリを作成(□) ☑ ソース管理に追加(□)		
					OK キャンセル

OK ボタンをクリックする。



プロパティで Size を 700,450 に変更し、Enter キーを押す。

Form1.h[デザイン]*x スタートページ ソリューションエクスプローラー プロパティ × マ Form1 Form1 System.Windows.Forms.F We form1 Form2 System.Windows.Forms.F Background IT Form2 StartPosition Windows.Formal Background IT Form2 Control Background IT Form2 Control Background IT Form3 UI Gothic, 9 Form2 Control Form2 Control Background IT Form3 UI Gothic, 9 Form2 ControlText Form3 ControlText Form3 Stable RightToLeftLi False Text Form1 UseWaitCorex False Size D> PD -Ju2047X (ピクセリレ単	● exp - Microsoft Visual Studio Academic ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) データ(A) ③・回・図・20 過 あ ね き つ・ペー・回・回 ・ Debug ・ Win32 第1 は 今 ヨ 可 ゆ 山 云 辺 路 奈 100 深 弊 弊 含 賞 許 許 年 [日 日 日 日	ツール(T) テスト(S) ・ 2 loseing	ローロー (H)てイントウ (W) に (W) C イントウ (H) こ に に (W) C イントウ (H) こ に に (W) C イントウ (H) こ (H) C イントウ (H) C イ) C イ (H) C イ) C (H) C イ) C (H)
	Form1.h (デザイン)* x スタート ページ ソリューション エクスプローラー		

ツールボックスを表示し、PictureBox を貼り付ける。

exp - Microsoft Visual Studio Academic アイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デパッグ(D) チーム(M) デー コ・ヨ・ピ J J J J A J A J A C - ロ・ロ・レ P Debug • Win32 第一日 今 司 可 ゆ 止 口 知 路 前 0 ~ 25 弊 第一号 支 許 許 一日 日 1	タ(A) 書式(O) ツール(T) テスト(S ・ 29 loseing ・ 3 3 3 3 =	
yuı->=> totton Form1		
(エラー一覧) III コマンド ウィンドウ III 出力 増売了	- - 1238 , 37	_포 년284 x 218

プロパティを表示し、

exp - Microsoft Visual Studio Academic ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) データ(A) ツール(T) デ	Fスト(S) ウィンドウ(W) ヘルプ(H)
[]・囲・22 』 21 & 国 23 ッ・ペ・回・□ ▶ [Debug ・ Win32 ・ 28 [◇ = 示 ◇ = 弓 凸 路 寺 ∞ 空 空 空 ▶ () ★ () 田 日 () も 国 通 。	y loseing 🔹 🖓 🖆 🗔 🖄 🛠 🍹
•omlh[デザイン]* × スタート ページ ソリューション エクスプローラー	プロ/デイ × マ PictureBox1 System.Windows.F・ System.Windows.F・ System.Dri Firs/IIII チョー Firs/IIIII Tage System.Dri InitialImage System.Dri BackgroundIr Tile BockgroundIr Tile BockgronIr Tile BockgronIr Tile BockgronIr Tile

PictureBox の size を 500,400 とする。

1 • 🖮 • 📂 🖬 🥔 X 🤅		(B) 97(99(D) -	チーム(M) データ(A) ・ Win32	書式(O) ツール(T) ・ 図 loseir	テスト(S) Ig	ウィンドウ(W) ヘル ・ 🔍 🛃	J(H) Г 🚮 🖄 🛠 🖁
[回望留寺	말 말 을 찾 않	* 42, 42, 1				
タートページ ソリュー	ション エクスプローラー	Form1.h [デザイン	v] ×		-	プロパティ ×	
••• Form1	O					pictureBox1 Syste いました。 Size Frorimage Frorimage Frorimage Frorimage Frorimage Frorimage Frorimage WaitOnLoad RackgroundIr BackgroundIr BackgroundIr BackgroundIr BackgroundIr BackgroundIr Size コントロールのサイン	m. Windows. F. 000, 400 System. Dr. System. Dr. Sys

ツールボックスを表示し、Button を貼り付ける。

• 🔚 • 😂 🚽 🍠 🐇 🐁 🧐 • ભ - 💭 • 🖏 🕨 Debug 🔹 Win32	• 🖄 loseing	(0) 0	- 🔩 🕾 📷 🖄 🌾	. :
12 冬山 四 4 二 二 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	II 21 -			
ート ページ ソリューション エクスプローラー Form1.h [デザイン]* x		-	ソールボックス ×	=
P Form1		<u>^</u>	▷ すべての Windows フォーム	-
	button 1	E	A X4729- A X4729- A X4729- A X4729- A X4729- CheckBox CheckedListBox ComboBox DateTimePicker ListBox ListBox ListBox MonthCalendar MothCalendar NotifyIcon NotifyIcon	

プロパティを表示し、Text を Next に変える。

oo exp - Microsoft Visual Studio Academic ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デパッグ(D) チーム(M) データ(A) □ ・ □ ・ ご ・ ご よ ふ ふ ・ ヴ ・ ロ ・ □ ・ □ ・ ○ ・ Debug ・ Win32 ■ 第 正 キ ヨ 可 や 皿 云 別 招 亦 @ ◎ 啓 際 廃 き 賞 計 & □ 記 □ = □ も =	ツール(T) テスト(S) : ・ 20	ウィンドウ(W) へルプ(H) ・∏ඥា:	
スタート ページ ソリューション エクスプローラー Form1.h (デザイン)* x ● ● Form1 ● ● Form1 ● ● Form1 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●			Vindows.Forms • itandard * is UI Gothic, 9 ControlText (#2U) iddleCenter (#2U) iddleCenter verlay verlay rue rue alase *
 ・エラー一覧 国 コマンド ウィンドウ 目 出力		Text コントロールに関連	付けられたテキ

PictureBox1 が Form1 からはみ出しているので、Form1 をクリックし、プロパティを表示し、 Size を 700,500 に変える。

🐱 exp - Microsoft Visual Studio Academic	
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) データ(A) 書式(O) ツール(T) テスト(S)) ウィンドウ(W) ヘルプ(H)
[姓氏수희 프아파 역간 破壞 이야 않 않 형 밖 태 또 된 경 경 류 및*	
スタートページ ソリューション エクスプローラー Form1.h <mark>Form1.h [デザイン] x</mark>	- プロパティ x =
	Form1 System.Windows.Forms.F + Padding 0, 0, 0, 0, 0 Padding 0, 0, 0, 0 StartPosition WindowsDefault WindowState Normal BackColor Control BackgroundIr Tile Cursor Default Cursor Default Cursor Default Cursor Default FormBorderS Sizable RightToLeft No RightToLeft No RightToLeft No RightToLeft No Size D>トロールのサイズ (ビクセル単
♀ エラー一覧 ■ コマンド ウィンドウ ■ 出力 集毎示了	

i

ボタンをダブルクリックする。



```
private: System::Void button1_Click(System::Object^ sender,
    System::EventArgs^ e) {
}
```

を

```
private: System::Void button1_Click(System::Object^ sender,
    System::EventArgs^ e) {
    Graphics^ g = pictureBox1->CreateGraphics();
    Pen^ pen = gcnew Pen(Color::Blue, 2);
    g->DrawLine(pen, 0, 200, 500, 200);
    g->DrawLine(pen, 250,0, 250, 400);
    for (int x=0; x<=500; x += 50)
        g->DrawLine(pen, x, 185, x, 215);
```

```
for (int y=0; y<=400; y += 50)
    g->DrawLine(pen, 235, y, 265, y);
int ox = (int)(250 + 50 * -5);
int oy = (int)(200 - 50 * Math::Exp(-5));
for (double x=-5; x<=5; x += 0.1) {
    g->DrawLine(pen, ox, oy, 250+50*x,
        200-50*Math::Exp(x));
    ox = 250+50*x;
    oy = 200-50*Math::Exp(x);
}
```

```
と修正する。実行する。
```



Next ボタンを押す。



座標軸と y = exp(x) の関数が -5 <= x <= 5 の範囲で表示されている。

```
private: System::Void button1_Click(System::Object^ sender,
    System::EventArgs^ e) {
    Graphics^ g = pictureBox1->CreateGraphics();
    Pen^ pen = gcnew Pen(Color::Blue, 2);
    g->DrawLine(pen, 0, 200, 500, 200);
    g->DrawLine(pen, 250,0, 250, 400);
    for (int x=0; x<=500; x += 50)
        g->DrawLine(pen, x, 185, x, 215);
    for (int y=0; y<=400; y += 50)
        g->DrawLine(pen, 235, y, 265, y);
    int ox = (int)(250 + 50 * -5);
    int oy = (int)(200 - 50 * Math::Exp(-5));
    for (double x=-5; x<=5; x += 0.1) {</pre>
        g->DrawLine(pen, ox, oy, 250+50*x,
            200-50*Math::Exp(x));
        ox = 250+50*x;
        oy = 200-50*Math::Exp(x);
    }
}
```

は button1_Click の名前から分かるように、Button1 がクリックされたときに実行される関数 です。

```
Graphics^ g = pictureBox1->CreateGraphics();
```

は、PictureBox1 に絵が描けるように、PictureBox1 のグラフィックスデバイスをg にセットして います。いつでもこのようにします。魔法を使うための呪文です。

```
Pen^ pen = gcnew Pen(Color::Blue, 2);
```

は、ペンを定義しています。色は Blue、線の太さは2ピクセルです。

g->DrawLine(pen, 0, 200, 500, 200); g->DrawLine(pen, 250,0, 250, 400);

は、g->DrawLine()を使って、 x 軸と y 軸を描いています。

```
for (int x=0; x<=500; x += 50)
  g->DrawLine(pen, x, 185, x, 215);
for (int y=0; y<=400; y += 50)
  g->DrawLine(pen, 235, y, 265, y);
```

は、g->DrawLine()を使って、座標軸の目盛りを 50 ピクセルごとに引いています。

```
for (int x=0; x<=500; x += 50)
  g->DrawLine(pen, x, 185, x, 215);
for (int y=0; y<=400; y += 50)
  g->DrawLine(pen, 235, y, 265, y);
```

では、x は、x=0, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500 の値を取り、

```
g->DrawLine(pen, x, 185, x, 215);
```

を実行します。

int ox = (int)(250 + 50 * -5); int oy = (int)(200 - 50 * Math::Exp(-5));

では、関数の描きはじめの点を ox と oy にセットしています。表示の座標系の原点が (250, 200) で、座標軸の目盛りを 50 ピクセルごとにしたので、50 倍しています。更に、数学の普通の座標 系とあわせるために ox は 250 + 50 * -5 と + していますが、 oy は 200 - 50 * Math::Exp(-5) と - を使います。+ だと上下が逆になります。Math::Exp() は指数関数です。指数関数は double $\exp(\text{double x})$ を使っても良いですが、この形の指数関数を使うためには、上のプログラムでやっ たように Form1.h の先頭に

```
#include <math.h>
```

```
が必要です。
```

```
for (double x=-5; x<=5; x += 0.1) {</pre>
           g->DrawLine(pen, ox, oy, 250+50*x,
               200-50*Math::Exp(x));
           ox = 250+50*x;
           oy = 200-50*Math::Exp(x);
       }
で、y=exp(x)の関数を描いています。線分の始点を (ox, oy) としています。このため、
           g->DrawLine(pen, ox, oy, 250+50*x,
               200-50*Math::Exp(x));
で、線分を描いた後、
           ox = 250+50*x;
           oy = 200-50*Math::Exp(x);
で、次の線分の始点 (ox, oy) を更新しています。これが分かりぬくければ、
       int ox = (int)(250 + 50 * -5);
       int oy = (int)(200 - 50 * Math::Exp(-5));
       for (double x=-5; x<=5; x += 0.1) {</pre>
           g->DrawLine(pen, ox, oy, 250+50*x,
               200-50*Math::Exp(x));
           ox = 250+50*x;
           oy = 200-50*Math::Exp(x);
       }
```

を

```
for (double x=-5; x<=5; x += 0.1) {
    g->DrawLine(pen, (int)(250+50*x),
       (int)(200-50*Math::Exp(x)),
       (int)(250+50*(x+0.1)),
        (int)(200-50*Math::Exp(x+0.1)));
}
```

としても良いです。(int)(...) が必要になったのは、なければ引数が int か float か判定できなくなっ たからです。元のものは、 ox と oy が int と宣言されていたので、(int)(...) がなくても VC++ がどちらか判断できたのです。しかし、これはマジックナンバー 0.1 が三カ所に出てくるので、

```
double dx = 0.1;
for (double x=-5; x<=5; x += dx) {
    g->DrawLine(pen, (int)(250+50*x),
        (int)(200-50*Math::Exp(x)),
        (int)(250+50*(x+dx)),
        (int)(200-50*Math::Exp(x+dx)));
}
```

とするのが良いです。このようにするとグラフを描く間隔を変えたいときに

double dx = 0.1;

の 0.1 だけを変えればいいからです。より良いプログラムを作るのは結構気を遣います。 このプログラムを作り替えて、Next のボタンを押すたびに、exp(x)の級数展開の近似式

$$e^x \sim 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} = \sum_{i=0}^n \frac{x^i}{i!}$$

を順次表示するようにする。

まず、階乗を計算する関数 double fact(double n) と最初の n+1 項までの和を計算する関数 double expN(int n, double x) を作る。

```
double fact(double n) {
    double r = 1;
    for (int i=1; i<= n; i++)</pre>
        r *= i;
    return r;
}
double expN(int n, double x) {
    double s = 0;
    for (int i=0; i<=n; i++)</pre>
        s += Math::Pow(x, i)/fact(i);
    return s;
}
```

これを

に

private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {

の前に打ち込む。



Next のボタンを押すたびに n を 0 から順に 1, 2, 3, 4, と増やしながら double expN(int n, double x) のグラフを表示すれば良い。

```
private: System::Void button1_Click(System::Object^ sender,
    System::EventArgs e) {
    Graphics^ g = pictureBox1->CreateGraphics();
    Pen^ pen = gcnew Pen(Color::Blue, 2);
    g->DrawLine(pen, 0, 200, 500, 200);
    g->DrawLine(pen, 250,0, 250, 400);
    for (int x=0; x<=500; x += 50)
        g->DrawLine(pen, x, 185, x, 215);
    for (int y=0; y<=400; y += 50)
        g->DrawLine(pen, 235, y, 265, y);
    int ox = (int)(250 + 50 * -5);
    int oy = (int)(200 - 50 * Math::Exp(-5));
    for (double x=-5; x<=5; x += 0.1) {</pre>
        g->DrawLine(pen, ox, oy, 250+50*x,
            200-50*Math::Exp(x));
        ox = 250+50*x;
        oy = 200-50*Math::Exp(x);
    }
}
```

```
static int N = 0;
   Pen^ pen2 = gcnew Pen(Color::Red, 2);
   ox = 250 + 50 * -5;
   oy = 200 - 50 * expN(N, -5);
   for (double x=-5; x<=5; x += 0.1) {</pre>
        g->DrawLine(pen2, ox, oy, 250+50*x, 200-50*expN(N, x));
       ox = 250+50*x;
       oy = 200-50 * expN(N, x);
   }
   N++;
を追加する。
   private: System::Void button1_Click(System::Object^ sender,
        System::EventArgs e) {
       Graphics^ g = pictureBox1->CreateGraphics();
       Pen^ pen = gcnew Pen(Color::Blue, 2);
       g->DrawLine(pen, 0, 200, 500, 200);
       g->DrawLine(pen, 250,0, 250, 400);
       for (int x=0; x<=500; x += 50)
            g->DrawLine(pen, x, 185, x, 215);
       for (int y=0; y<=400; y += 50)
            g->DrawLine(pen, 235, y, 265, y);
        int ox = (int)(250 + 50 * -5);
        int oy = (int)(200 - 50 * Math::Exp(-5));
        for (double x=-5; x <=5; x += 0.1) {
            g->DrawLine(pen, ox, oy, 250+50*x,
                200-50*Math::Exp(x));
            ox = 250+50*x;
            oy = 200-50*Math::Exp(x);
       }
       static int N = 0;
       Pen^ pen2 = gcnew Pen(Color::Red, 2);
       ox = 250 + 50 * -5;
       oy = 200 - 50 * expN(N, -5);
       for (double x=-5; x<=5; x += 0.1) {</pre>
           g->DrawLine(pen2, ox, oy, 250+50*x,
               200-50*expN(N, x));
           ox = 250+50*x;
           oy = 200-50 * expN(N, x);
        }
       N++;
   }
となる。
```

exp - Microsoft Visual Studio Academic	
기카이ルE 端葉E 정치(2) 기타보기가 ED EURE) 카이에/(2) 카드시(2) 카드시(3) 카드시(3) 카드시(3) 카디시(3) 카디시(3) 가디시(3) 가다(3) (3) (3) (3) (3) (3) (3) (3) (3) (3)	
スタートページ ソリューション エクスプローラー Form1.h × Form1.h (デザイン) ・	プロパティ × 🗧
<pre>private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) { Graphics* g = pictureBox(->OreateGraphics(); Pen^pen = genew Pen(Color::Blue, 2); g->Drawline(pen, 0, 200, 500, 200); g->Drawline(pen, 2500, 250, 400); for (int x=0; x<=50) g->Drawline(pen, x, 155, x, 215); for (int y=0; y<=400; y, y=400); } </pre>	COME COME
<pre>a=>10rawLine(pen, 235, y, 265, y); int ox = (int)(250 + 50 + -5); int oy = (int)(200 - 50 * Math::Exp(-5)); for (double x-51; x<51; x+ 0.1) {</pre>	- Dinjeccu - File
<pre>Pen^pent2 = zcnew Pen(Color::Red, 2); ox = 250 + 50 x - 5; oy = 200 - 50 x expN(Nx-5); for (double x - 5; x<=5; x += 0.1) { g->DrawLine(pent2, ox, oy, 250+50*x, 200-50*expN(N, x)); ox = 250+50*x; oy = 200-50*expN(N, x); }</pre>	2.08
	その 他
· · · · · · · · · · · · · · · · · · ·	1文字 挿入!

実行してみる。



Next ボタンを押す。







Next ボタンを押す。







次々新しいグラフが追加されている。

```
static int N = 0;
Pen^ pen2 = gcnew Pen(Color::Red, 2);
ox = 250 + 50 * -5;
oy = 200 - 50 * expN(N,-5);
for (double x=-5; x<=5; x += 0.1) {
  g->DrawLine(pen2, ox, oy, 250+50*x,
        200-50*expN(N, x));
  ox = 250+50*x;
  oy = 200-50*expN(N, x);
}
N++;
```

の

static int N = 0;

は、整数変数 N を宣言し、その値を 0 としなさいという定義ですが、 static という修飾語がつい ているのは、この関数が実行された後も N は記憶されていて、この関数の最後に

N++;

という、N を1つ増加しなさいという命令があるので、次にこの関数が実行されるときは、N = 0 と定義して、関数を実行するのではなく、前回の関数の呼び出しの時の N の値を使いますという 指示になります。従って、関数が呼ばれるごとに N の値は N = 0, 1, 2, 3, 4, 5, ... となります。

Pen^ pen2 = gcnew Pen(Color::Red, 2);

で、別のペン pen2 を定義しています。expN() は pen2 を使って描きます。

```
ox = 250 + 50 * -5;
       oy = 200 - 50 * expN(N, -5);
       for (double x=-5; x<=5; x += 0.1) {
           g->DrawLine(pen2, ox, oy, 250+50*x,
               200-50*expN(N, x));
           ox = 250+50*x;
           oy = 200-50 * expN(N, x);
       }
の部分で、関数 expN(N, x) を描いています。この部分も上で述べた修正
       double dx = 0.1;
       for (double x=-5; x<=5; x += dx) {
           g->DrawLine(pen, (int)(250+50*x),
               (int)(200-50*Math::Exp(x)),
               (int)(250+50*(x+dx)),
               (int)(200-50*Math::Exp(x+dx)));
       }
をしていれば
       for (double x=-5; x <=5; x += dx) {
           g->DrawLine(pen, (int)(250+50*x),
               (int)(200-50*Math::Exp(x)),
               (int)(250+50*(x+dx)),
               (int)(200-50*Math::Exp(x+dx)));
       }
とします。
       double dx = 0.1;
```

は、上

```
double dx = 0.1;
       for (double x=-5; x<=5; x += dx) {
           g->DrawLine(pen, (int)(250+50*x),
               (int)(200-50*Math::Exp(x)),
               (int)(250+50*(x+dx)),
               (int)(200-50*Math::Exp(x+dx)));
       }
で定義しているので、これを再び定義してはいけません。エラーになります。
       N++:
は、次回のこの関数の呼び出しのために、N を一つ増やしています。
 だから、g->DrawLine()の書き方が気に入らなければ
   private: System::Void button1_Click(System::Object^ sender,
       System::EventArgs^ e) {
       Graphics^ g = pictureBox1->CreateGraphics();
       Pen^ pen = gcnew Pen(Color::Blue, 2);
       g->DrawLine(pen, 0, 200, 500, 200);
       g->DrawLine(pen, 250,0, 250, 400);
       for (int x=0; x<=500; x += 50)
           g->DrawLine(pen, x, 185, x, 215);
       for (int y=0; y<=400; y += 50)
           g->DrawLine(pen, 235, y, 265, y);
       double dx = 0.1;
       for (double x=-5; x<=5; x += dx) {
           g->DrawLine(pen, (int)(250+50*x),
               (int)(200-50*Math::Exp(x)),
               (int)(250+50*(x+dx)),
               (int)(200-50*Math::Exp(x+dx)));
       }
       static int N = 0;
       Pen^ pen2 = gcnew Pen(Color::Red, 2);
       for (double x=-5; x<=5; x += dx) {
           g->DrawLine(pen2, (int)(250+50*x),
               (int)(200-50*expN(N, x)),
               (int)(250+50*(x+dx)),
               (int)(200-50*expN(N, x+dx)));
       }
       N++;
   }
とプログラミングすればいいです。
 補足: 階乗を計算する関数 double fact(double n)
```

```
53
```

```
double fact(double n) {
    double r = 1;
    for (int i=1; i<= n; i++)
        r *= i;
        return r;
}
は4つの部分から構成されています。関数は、
戻り値の型 関数名(引数の並び) {
        関数の本体
}</pre>
```

の構成になっています。今の場合は、戻り値の型は double、すなわち実数値を返す関数である と宣言しています。関数名は fact です。引数の並びは今の場合、引数は一個で、double n と実数 値を取る引数 n があることを示しています。関数の本体は と で囲まれた部分で、今の場合、

```
double r = 1;
for (int i=1; i<= n; i++)
    r *= i;
return r;
```

が関数の本体で、実数値を取る変数 r の定義 (r は実数値変数であるという宣言とその値を 1 とす るという初期設定が同時に行われています) があって、次の for 文で、整数値の変数 i を 0 から (int i=0 で初期設定) n まで (i<=n が成り立つ間(条件式))、1つづつ増やし(i++ は i を 1 つ増やせという命令で、i += 1 や i = i + 1 の省略形です) ながら変化させ、(簡潔に言うと、整 数値変数 i を 1 から n まで 1つづつ増やしながら、) r に順次掛けています。この結果、for 文が 修了したとき、r は n の階乗の値になっています。最後に、r の値を関数の値として return 文で 返しています。

これで、階乗の手続き的定義

```
n! = 1 * 2 * 3 * \dots * n
```

を C++ で表現したことになります。 これを Prolog や Scheme でやったように、再帰的に

```
double fact(double n) {
    if (n == 0)
        return 1;
    else
        return n * fact(n-1);
}
と定義することも出来ます。
    if 文は、Scheme の場合と異なり、2つの形式があります。
if (条件式) {
```

```
命令の並び}
```

```
の形で、条件式が真(true)の時、命令の並びを実行するか
if (条件式) {
   命令の並び
} else {
   命令の並び
}
の形で、条件式が真(true)の時、上の命令の並びを実行し、そうでないとき下の命令の並びを実
行します。
if (条件式) {
   命令の並び}
や
if (条件式) {
   命令の並び
} else {
   命令の並び
}
で、命令の並びが一個の命令の場合は、
   if (n == 0)
      return 1;
   else
      return n * fact(n-1);
のように、命令の並びを囲む {と}は省略できます。
   if (n == 0)
      return 1;
   else
      return n * fact(n-1);
の n == 0 は n が 0 の時は真 (true)、そうでなければ偽 (false) を与える条件式です。
 級数の最初の n+1 項までの和を計算する関数 double expN(int n, double x)
double expN(int n, double x) {
   double s = 0;
   for (int i=0; i<=n; i++)</pre>
      s += Math::Pow(x, i)/fact(i);
   return s;
}
も、戻り値の型が double で、関数名が expN で、引数の並びが int n, double x と2つの引数が
```

宣言されています。今の場合、整数値を取る引数 n と実数値をとる引数 x が,で区切られて並んでいます。関数の本体は

```
double s = 0;
for (int i=0; i<=n; i++)
    s += Math::Pow(x, i)/fact(i);
return s;
```

で、実数値変数 s の定義(s は実数値変数であるという宣言とその値を 0 とするという初期設定) がまずなされ、次に for 文で、整数値の変数 i を 0 から(int i=0 で初期設定) n まで(i<=n が 成り立つ間(条件式))、1つづつ増やし(i++ は i を 1つ増やせという命令で、i += 1 や i = i + 1 の省略形です)ながら変化させ、s に x の i 乗を i の階乗で割った値を加えています。最後 に、s の値を関数の値として return 文で返しています。

これも Prolog や Scheme でやったように、再帰的に

```
double expN(int n, double x) {
    if (n == 0)
        return 0;
    else
        return Math::Pow(x, n)/fact(n) + expN(n-1, x);
}
```

```
と定義することも出来ます。
```

これを新しいグラフだけが表示されるように修正するには、Next ボタンを押すたびにまず画面 を白く塗りつぶせばよい。そのためには2行目に

```
Brush^ brush = gcnew SolidBrush(Color::White);
g->FillRectangle(brush, 0, 0, 500, 400);
```

を追加する。







のようなグラフが得られるようになる。

Brush^ brush = gcnew SolidBrush(Color::White); g->FillRectangle(brush, 0, 0, 500, 400);

の

Brush^ brush = gcnew SolidBrush(Color::White);

で、矩形を塗りつぶすためのブラシを定義しています。White のブラシです。

g->FillRectangle(brush, 0, 0, 500, 400);

```
で、g->FillRectangle() で、矩形の領域を指示されたブラシで塗りつぶします。0,0,500,400
は左上隅の座標が (0,0) で、幅 500、高さ 400 の矩形を指示しています。
このプログラムが分かりぬくければ、上で述べたように、全体を
```

```
private: System::Void button1_Click(System::Object sender,
    System::EventArgs e) {
    Graphics g = pictureBox1->CreateGraphics();
    Brush brush = gcnew SolidBrush(Color::White);
    g->FillRectangle(brush, 0, 0, 500, 400);
    Pen pen = gcnew Pen(Color::Blue, 2);
    g->DrawLine(pen, 0, 200, 500, 200);
    g->DrawLine(pen, 0, 200, 500, 200);
    for (int x=0; x<=500; x += 50)
      g->DrawLine(pen, x, 185, x, 215);
    for (int y=0; y<=400; y += 50)
      g->DrawLine(pen, 235, y, 265, y);
    double dx = 0.1;
    for (double x=-5; x<=5; x += dx) {
      g->DrawLine(pen, (int)(250+50*x),
```

```
(int)(200-50*Math::Exp(x)),
    (int)(250+50*(x+dx)),
    (int)(200-50*Math::Exp(x+dx)));
}
static int N = 0;
Pen<sup>^</sup> pen<sup>2</sup> = gcnew Pen(Color::Red, 2);
for (double x=-5; x<=5; x += dx) {
    g->DrawLine(pen2, (int)(250+50*x),
        (int)(200-50*expN(N, x)),
        (int)(250+50*(x+dx)),
        (int)(200-50*expN(N, x+dx)));
}
N++;
}
```

```
とプログラミングすればいいです。
```

演習問題: sin(x) や cos(x) で同じ事をするプログラムを作成しなさい。 極方程式 r = $3sin(3 \theta)$ のグラフを描いてみよう。 Microsoft Visual Studio 2010 を起ち上げます。



「新しいプロジェクト … 」をクリックします。

最近使用したテンプレート	.NET Framework 4 ▼ 並べ替え基準: 既定	 ・ ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	の検索の
インストールされたテンプレート	Win32 コンソール アプリケーション	Visual C++ 種類: Visual C++	
Visual C++ ATL CLR	MFC アプリケーション	Windows ユーザー インターフェ Sむアプリケーションを作成する プロジェクトです。	・イスを いための
全般	「「」 Win32 プロジェクト	Visual C++ ≡	
FC テスト	*** 空のプロジェクト	Visual C++	
Win32 ▶ 他の言語	ATL プロジェクト	Visual C++	
▷ その他のプロジェクトの種類 ▷ データベース	M MFC DLL	Visual C++	
▶ テスト プロジェクト	Windows フォーム アプリケーション	Visual C++	
インライン ナンノレート	CLR コンソール アプリケーション	Visual C++	
	李の CLR プロジェクト	Visual C++ 🔻	
名前(№): <名前を	入力してください>		
易所(<u>L</u>): D:¥jyug	you¥	▼ 参照(B)	
ソリューション名(<u>M</u>): <名前を	入力してください>	▼ ソリューションのディレクトリを	作成(D)
			Aug The

Windows フォームアプリケーションを選択し、名前 (N): のエディットボックスに例えば sanyousen と入力し、

しいプロジェクト					3 ×
最近使用したテンプレ	/-ト	.NET Fra	amework 4 👻 並べ替え基準: 既定	• 111 [11	インストールされたテンプレートの検索 🔎
インストールされたテ	シプレート	÷.	Win32 コンソール アプリケーション	Visual C++	▲ 種類: Visual C++
▲ Visual C++ ATL CLR		FC	MFC アプリケーション	Visual C++	Windows ユーザー インターフェイスを 含むアプリケーションを作成するための プロジェクトです。
全般			Win32 プロジェクト	Visual C++	E
テスト		**	空のプロジェクト	Visual C++	
Win32 他の言語		ATL	ATL プロジェクト	Visual C++	
 その他のプロジェ・ データベース 	クトの種類	M C	MFC DLL	Visual C++	
テスト プロジェク	1	*	Windows フォーム アプリケーション	Visual C++	
125425222	- 1-	÷4	CLR コンソール アプリケーション	Visual C++	
		**	空の CLR プロジェクト	Visual C++	-
ろ前(<u>ℕ</u>):	sanyousen				
易所(<u>L</u>):	D:¥jyugyou¥				参照(B)
ノリューション名(Μ)	: sanyousen				✓ ソリューションのディレクトリを作成(D)
					ソース管理に追加(U)
					OK キャンセル

OK ボタンをクリックする。

-om1h[デザイン] × スタートページ ソリューションエクスプローラー ・	ツール(T) テス	=スト(S) 5	ウィンドウ(W) へ ・ 🛛 💦	レブ(H) 子 G 23 火	×
		•	JUJYFY X Form1 System.V Form1 System.V Fording Ø Padding Star StartPosition WindowState BackgroundIr Gursor BackgroundIr Cursor ForeColor ForeColor ForeColor ForeMoreColor ForeColor ForeMoreColor ForeMoreColor Text UseWaitCursc Text	Vindows.Forms. 0, 0, 0, 0 300, 300 WindowsDefau Normal Control (72 U) Tile Default MS UI Gothic, 1 ControlTex Sizable No Folse Form1 False	▼ .F

プロパティで Size を 600,450 に変更し、ツールボックスを表示し、PictureBox を貼り付け、プロパティを表示し、PictureBox の size を 350,350 とする。

ァイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) データ(A) 書式(O)		
) ツール(1) テスト(5)	ウィンドウ(W) ヘルプ(H)
회·희·철·철정정정정정정정정정정정정정정정정정정	Ioseing	- X 🖾 🗤 🗹 Y 🗧
<pre>rmih [テザイン]* × スタートページ ソリューション エクスプローラー</pre>	• •	プロパティ × 🗧
	=	pictureBoxt System.Windows.F - ・ ジョントロールのサイズ(ピクセンル単…・

プロパティのイベントで paint をダブルクリックする。

😎 sanyousen - Microsoft Visual Studio Academic			- • × •)
ファイル(E) 編集(E) 表示(⊻) プロジェクト(P) ビルド(B) デバッグ(D) チーム(M) データ(A) ツール(I) テスト	(<u>5</u>) ウィンドウ((₩) へレプ(H)	
🕴 🐨 🕶 😂 🛃 😹 🐇 🗠 🎕 🤲 🔹 🕬 🗸 🖓 🖬 🖓 🖬 🖓 🖬 🖓 🖬 🖓 👘 🖓	seing	- 🖓 🕾 🛛	🚮 🔁 🛠 🗒 👘
[□] 哈哈哈林市 建建 □ 2 □ 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2			
Form1.h [デザイン]* スタート ページ ソリューション エクスプローラー Form1.h* ×	-	プロパティ ×	₹
sanyousen::Form1 - 🖓 pictureBox1_Paint(System::Object ^ sender, S	ystem::Winc -		-
// E-mol	, arurmin.r	81 24 📼	
// FORMI //			
this->AutoScaleDimensions = System::Drawing::SizeF(6, 12); this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;			
this->ClientSize = System::Drawing::Size(584, 412); this->Controls->Add(this->pictureBox1);			
this->Name = L"Form1"; this->Text = L"Form1";			
<pre>(cli::safe_cast<system::componentmodel::isupportinitialize^>(this->pictureBox1 this->ResumeLayout(false);</system::componentmodel::isupportinitialize^></pre>))->EndInit		
}			
#pragma endregion	:PaintEvent		
}:			
	E		
100 % - <	Þ		
😪 エラー一覧 🖿 コマンド ウィンドウ 🗉 出力			
準備完了 79 行	115 列	112 文字	挿入!
		Santo - Contractor - Contractor	

極方程式 r = 3sin(3 θ) は θ を媒介変数として、x(θ)=3sin(3 θ)cos(θ), y(θ)=3sin(3 θ)sin(θ) と表されるから、

```
private: System::Void pictureBox1_Paint(System::Object^ sender,
   System::Windows::Forms::PaintEventArgs e) {
   }
の { と } の前に次のようにプログラムを打ち込む。
double X(double t) {
   return 3*Math::Sin(3*t)*Math::Cos(t);
}
double Y(double t) {
   return 3*Math::Sin(3*t)*Math::Sin(t);
}
更に、
private: System::Void pictureBox1_Paint(System::Object^ sender,
   System::Windows::Forms::PaintEventArgs^ e) {
   }
の{と}の間に次のようにプログラムを打ち込む。
       Graphics^ g = e->Graphics;
       Pen^ pen = gcnew Pen(Color::Black, 1);
       g->DrawLine(pen, 0, 175, 350, 175);
       g->DrawLine(pen, 175,0, 175, 350);
       for (int x=25; x<=350; x += 50)
           g->DrawLine(pen, x, 170, x, 180);
       for (int y=25; y<=350; y += 50)
           g->DrawLine(pen, 170, y, 180, y);
```

```
double dt = Math::PI/80;
Pen^ pen2 = gcnew Pen(Color::Red, 2);
for (double t=0; t<2*Math::PI; t += dt) {
  g->DrawLine(pen2, (int)(175+50*X(t)),
      (int)(175-50*Y(t)),
      (int)(175+50*X(t+dt)),
      (int)(175-50*Y(t+dt)));
```







```
double X(double t) {
    return 3*Math::Sin(3*t)*Math::Cos(t);
}
double Y(double t) {
    return 3*Math::Sin(3*t)*Math::Sin(t);
```

```
}
private: System::Void pictureBox1_Paint(System::Object^
                                                           sender,
    System::Windows::Forms::PaintEventArgs^ e) {
    Graphics^ g = e->Graphics;
    Pen^ pen = gcnew Pen(Color::Black, 1);
    g->DrawLine(pen, 0, 175, 350, 175);
    g->DrawLine(pen, 175,0, 175, 350);
    for (int x=25; x<=350; x += 50)
        g->DrawLine(pen, x, 170, x, 180);
    for (int y=25; y<=350; y += 50)
        g->DrawLine(pen, 170, y, 180, y);
    double dt = Math::PI/80;
    Pen^ pen2 = gcnew Pen(Color::Red, 2);
    for (double t=0; t<2*Math::PI; t += dt) {</pre>
        g->DrawLine(pen2, (int)(175+50*X(t)),
            (int)(175-50*Y(t)),
             (int)(175+50*X(t+dt)),
             (int)(175-50*Y(t+dt)));
    }
}
の
double X(double t) {
    return 3*Math::Sin(3*t)*Math::Cos(t);
}
double Y(double t) {
    return 3*Math::Sin(3*t)*Math::Sin(t);
}
は、x(\theta) = 3\sin(3 \theta)\cos(\theta), y(\theta) = 3\sin(3 \theta)\sin(\theta)を引数をtとして関数として定義している。
    Graphics^ g = e->Graphics;
    Pen^ pen = gcnew Pen(Color::Black, 1);
    g->DrawLine(pen, 0, 175, 350, 175);
    g->DrawLine(pen, 175,0, 175, 350);
    for (int x=25; x<=350; x += 50)
        g->DrawLine(pen, x, 170, x, 180);
    for (int y=25; y<=350; y += 50)
        g->DrawLine(pen, 170, y, 180, y);
で、座標軸を描き、
    double dt = Math::PI/80;
    Pen^ pen2 = gcnew Pen(Color::Red, 2);
    for (double t=0; t<2*Math::PI; t += dt) {</pre>
```

```
63
```

で、(175, 175) を原点とし、拡大係数を 50 として、x(t)= $3\sin(3t)\cos(t)$, y(t)= $3\sin(3t)\sin(t)$ を t=0 から 2 π まで $\pi/80$ づつ増やしながら、線分で結んで描いている。

別のプログラムを作ってみます。

Microsoft Visual Studio 2010 を立ち上げ、上と同じように適当な名前でプロジェクトを作り、 Form に PictureBox と Button 2個を配置する。



```
button1 をダブルクリックし、
```

```
private: System::Void button1_Click(System::Object^
                                                    sender,
System::EventArgs e) {
    Graphics^ g = pictureBox1->CreateGraphics();
    Brush^ brush = gcnew SolidBrush(Color::White);
    g->FillRectangle(brush, 0, 0, 360, 360);
    Pen^ pen = gcnew Pen(Color::Blue, 1);
    g->DrawEllipse(pen, 50, 50, 260, 260);
    double pi = Math::PI;
    for (double t=0; t<2*pi; t += pi/80) {</pre>
        g->DrawLine(pen, (int)(180+130*cos(t)),
            (int)(180-130*sin(t)),
            (int)(180+130*cos(2*t)),
            (int)(180-130*sin(2*t)));
    }
}
  と打ち込む。さらに、button2 をダブルクリックし、
private: System::Void button2_Click(System::Object^ sender,
```

```
System::EventArgs^ e) {
    Graphics^ g = pictureBox1->CreateGraphics();
    Brush^ brush = gcnew SolidBrush(Color::Yellow);
    g->FillRectangle(brush, 0, 0, 360, 360);
    Pen^ pen = gcnew Pen(Color::Red, 1);
    g->DrawEllipse(pen, 50, 50, 260, 260);
    double pi = Math::PI;
    for (double t=0; t<2*pi; t += pi/80) {</pre>
        g->DrawLine(pen, (int)(180+130*cos(t)),
            (int)(180-130*sin(t)),
            (int)(180+130*cos(3*t)),
            (int)(180-130*sin(3*t)));
   }
}
  と打ち込む。
 Form1.h ファイルの先頭に
#include <math.h>
  を追加する。
 実行すると
```

Hermi Formi	
	button1
	button2

となります。button1 をクリックすると



となり、button2 をクリックすると



となります。

例題: $y = x \sin x$ ($-5\pi \le x \le 5\pi$) のグラフを描け。 Form の中央に PictureBox を配置する。

🥵 G1 – Microsoft Visual Studio	🖮 🚭 A般 😂 🥔 😰 🎊 🗉	
ファイル(E) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) データ(A) 書式(D) ツール	ロ テストの ウベバウ畑 ヘルプロ	
📆 • 🕮 • 📸 🛃 🎒 🐰 🖙 🖭 🕫 - 🕫 - 🚑 • 🖏 🕨 Debus 🔹 Win32	🔹 🥶 pattern 🔹 🕫 🐨 🕸 🎾 🖬 🗔 🔹 🖕	
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○		
ソリューションエクスプローラー + + × sourceannotationsh G1.cpp Form1.h Form1.h [デザ・	12 25-1-X-9	ツールボックス 👻 🕂 🗙
June 2 J		I STC20 Windows 2+ A STC2 DVD-B R #C/29 D Duton C CheckBox C
any, any?; IT / any /		■ NotifySall サーバーエンステーシックールボックス フロパワイ ・3 × Form1 System Windows Forms For → また 日本の一日の一日の一日の Bockgroundinas Tit Dockgroundinas Tit Dockgroundina
A CONTRACTOR CONTRACTOR		ForeColor ControlTex1
87 #1=0#=0	* t ×	FormBorderStyl Sizable RightTol aft No
		RightToLeftLays False Text Form1
□□-ド宝蔵ウインドウ [201470日しブラウザ] □ 出力 □□検索結果 1 [2015ンボルの検索結果 [20155]		コントロールに関連付けられたテキストです。
77/8	-	

picturebox1 のイベント paint を次のように定義する。

```
private: System::Void pictureBox1_Paint(System::Object^ sender,
   System::Windows::Forms::PaintEventArgs^ e) {
    Graphics^ g = e->Graphics;
    Pen^ pen = gcnew Pen(Color::Blue);
    double pi = Math::PI;
    g->DrawLine(pen, 0, 180, 360, 180);
    g->DrawLine(pen, 180, 0, 180, 360);
    int ox = (int)(180-180.0*5*pi/15.7079);
    int oy = (int)(180-10*(-5*pi)*sin(-5*pi));
    for (double t=-5*pi; t<=5*pi; t += pi/80) {</pre>
        g->DrawLine(pen, ox, oy,
            (int)(180+180.0*t/15.7079),
            (int)(180-10*t*sin(t)));
        ox = (int)(180+180.0*t/15.7079);
        oy = (int)(180-10*t*sin(t));
    }
}
```

Form1.h ファイルの先頭に

```
#include <math.h>
```

を追加する。 コンパイルし、実行すると



となる。

例題: (1/2,0) を中心とする直径1の円上の点を動点として、直径 *OP* の円群を描け。輪郭線に カージオイドが浮かび上がる。

Form の中央に PictureBox を配置する。

🏟 G1 – Microsoft Visual Studio		🗄 🖶 🜒 A ARE 😂 🥔 😰 🎬 🗉	
ファイル(E) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) データ(P)	書式(Q) ツール(D) テスト(G) ウィンドウダ	∮ ヘルプ®	
🛐 • 🛅 • 📸 🛃 👹 🐰 🖏 🔃 🕫 - 🔍 - 🐉 • 🖏 🕨 Debue	- Win32 - 🍅 patt	xm 🔹 💀 🕾 🖬 🗉 🗸 📰	
·[集] [12 本 4] 同 4 四 1 同 到 图 集] ···· 25 25 25 3 5 3 5 3 5 5 5 5 5 5 5 5 5 5	상 아이 다 다 다 만 📑 📑 🔒 🚬		
ソリューションエクスプローラー + 年 × sourceannotationsh G1.cpp Form1	h Form1.h [7912] 28-1-8-9	-	× ツールボックス - + ×
			・ すべての Windows フォーム
- フリューション 'G1' (1 プロジェクト)			- 3t2 32F0-#
e- 🗿 G1			ab Button
- C Arrandelinia con			Check Box
- 🚰 G1.cpp			CheckedListBox
- 🗠 stdafx.cpp			ComboBox
			DateTimePicker
- 🖓 Form1.resX			A Label
nesource.h			A LinkLabel
- in stdetxh			E ListBox
- i app.ico	6		222 ListView
- Processor			 MaskedTextBox
HeadMe.txt			MonthGalendar
			Notifylcon
			3 -M- 19X9- 20 - 108995
			70//?r • # ×
			Form1 System.Windows.Forms.Forr +
			21 🔟 🖉 🖾
			日 表示
L	I		BackgroundIma (20)
			BackgroundImar Tile
an yy. an op. and a second sec			E Font MS UI Gothic 9c
W7		- 0	ForeColor ControlTex1
un-matrix		••	FormBorderStyl Sizable RightToLeft No
	2		RightToLeftLay False
			Text Form1
			Text
			コントロールに関連付けられたテキストで
□□ード定義ウィンドウ いっぽび出しブラウザ □□出力 🖳 検索結果 1 🚑 シンボルの検	緊結果 (為エラー→覧)		7.
אעקב			

picturebox1 のイベント paint を次のように定義する。

```
private: System::Void pictureBox1_Paint(System::Object^ sender,
   System::Windows::Forms::PaintEventArgs^ e) {
    Graphics^ g = e->Graphics;
    Pen^ pen1 = gcnew Pen(Color::Black);
    g->DrawLine(pen1, 0, 180, 360, 180);
    g->DrawLine(pen1, 180, 0, 180, 360);
    double pi = Math::PI;
    int K = 150;
    for (double t = 0; t<=2*pi; t += pi/20) {</pre>
        double x = 0.25 + \cos(t)/4.0;
        double y = sin(t)/4.0;
        int x1 = (int)(K*(x-sqrt(0.5+cos(t)/2.0)/2.0)+180);
        int y1 = (int)(K*(-y-sqrt(0.5+cos(t)/2.0)/2.0)+180);
        int x2 = (int)(K*sqrt(0.5+cos(t)/2.0));
        int y2 = (int)(K*sqrt(0.5+cos(t)/2.0));
        g->DrawEllipse(pen1, x1, y1, x2, y2);
    }
}
  Form1.h ファイルの先頭に
#include <math.h>
  を追加する。
  コンパイルし、実行すると
```



```
となる。
```

この図を描くアニメーションの Python and Pygame のプログラムは次のようになります。

```
import pygame
from math import *
```

```
pygame.init()
```

```
black = (0,0,0)
white = (255, 255, 255)
green = (0, 255, 0)
red= (255, 0, 0)
pi = 3.141592653
size = (700, 500)
screen = pygame.display.set_mode(size)
pi = atan(1.0)*4
done = False
clock = pygame.time.Clock()
theta = pi
K = 200
while done == False:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done = True
    screen.fill(white)
    t = 0
```

```
while t <= 2*pi - 2*theta:
   x = 0.25 + \cos(t)/4.0
   y = sin(t)/4.0
   x1 = (int)(K * (x - sqrt(0.5 + cos(t)/2.0)/2.0) + 350)
   y1 = (int)(K * (-y - sqrt(0.5 + cos(t)/2.0)/2.0) + 250)
   x2 = (int)(K * sqrt(0.5+cos(t)/2.0))
   y_2 = (int)(K * sqrt(0.5+cos(t)/2.0))
   if x2 == 0 or y2 == 0:
        t += pi/20
        continue
   pygame.draw.ellipse(screen, green, [x1, y1, x2, y2],2)
    t += pi/20
theta -= pi/20
pygame.display.flip()
if theta < 0:
    theta = pi
clock.tick(5)
```

```
pygame.quit()
```



例題: 原点を中心とし半径が a の円 O の外側を半径が b の円 C が円 O に外接しながら滑ることなく転がるとき、円 C 上の点 P の軌跡を考える。ただし、点 P のはじめの位置は、円 O と x 軸の正の部分との交点 A とする。

円 C が転がるとき、動径 OC が表す角を θ とし、そのときの外接する点を Q、点 P の座標を (x,y) とすると、P の軌跡は θ を媒介変数として

$$x = (a+b)\cos\theta - b\cos\frac{a+b}{b}\theta, \ \ y = (a+b)\sin\theta - b\sin\frac{a+b}{b}\theta$$

で表される。この曲線を エピサイクロイド という。エピサイクロイドを描くプログラムを作れ。 Form に PictureBox と Button と Label 2個と textBox 2個を配置する。

🏶 G5 – Microsoft Visual Studio	(📼)	📢 A 🕀 🥌 🥔 🛛 👬 🗉	
ファイル(E) 編集(E) 表示(W) ブロジェクト(E) ビルド(B) デバッグ(E)) データ(の) 書式(の) ツール(① テスト(の) ウィンドウ(の) ヘル	1719	
🛐 • 🖽 • 😂 🖬 🖓 💫 🖎 🖄 🗠 🕬 • 🚳 • 🖏	Debus • Win32 • 🌁 pattern	- 🔍 🕾 🖄 🏷 🖬 🖬 - 🖕	
(単) ほうき (中の) (日) (日) (日) (日) (日) (日) (日) (日) (日) (日	을 찾 않 야! 🕀 🐼 🦦 🖼 🖼 😹 🖕		
ソリューション エクスプローラー + 年 × /スタート ページ Form1h Fe	arm1.h [デザイン]	•	x ツールボックス - み >
			■ すべての Windows フォーム
□ 2/Jューション '35' (1 プロジェクト)			□ JCJ JJF0 ⁻ ル
B (1) Q5			ab Button
- C AssemblyInfo.cop			CheckBox
🚰 G6.cpp			CheckedListBox
- Stdefx.cpp			ComboBox
B- Corm1h	-		TateTimePicker
🔤 🚳 Form1.resX	0		A Label
h resource.h			A LinkLabel
□ □ リソース ファイル			ListBox
🔟 app.ico	b		22 ListView
BeadMetyt	2		Masked lextBox
			- Notifulaan
			TR Atomatical to Dama
	d button	B	A サーバー エクスブー ア・ツールボック
	3		プロパティ - キ >
			button1 System.Windows.Forms.B.
			80 21 III 🖉 📧
			FlatStyle Standard
			E Font MS UI Gothic, 9p ForeColor
			Image (20)
			Image Align MiddleCenter
出力		* #	× ImageKey 030
出力元の表示(5): デバッグ 🔹 🕥 🗐 😥			ImageList (121,) DidutTal att
'G5.exe' (マネージ型): 'C:WWINDOWSWassemblyWGAC_MSILWSystem.W	indows.FormsW2.0.0.0_b77a5c561934e089WSystem.#indows.Form	es.dll'が読み込まれました	Text button1
'G5.exe' (マネージ型): 'C:WWINDOWSMassemblyWGAC_MSILWSystem.D 'G5.exe' (マネージ型): 'C:WWINDOWSMassemblyWGAC_MSILWascorlib	resources¥2.0.0.0_ia_b77a5c561934e089¥ascor1ib.resources.	ウムまれました dll が読み込まれました	TextAlign MiddleCenter
プログラム [2012] G5.exe: マネージ はコード 0 (0x0) で終了し	、ました。		 Text コントロールに関連付付られたテキストで
* コード定義ウベンドウ1*= 呼び出しブラウザ1 == 出力 (== 検索結果 11.55.5)	○ボルの検索結果 気エラー一覧		3.
		111 APR	
3476		LJ 425.	264 ID 91 × 21

button1 をダブルクリックし、

```
private: System::Void button1_Click(System::Object^ sender,
   System::EventArgs^ e) {
    Graphics^ g = pictureBox1->CreateGraphics();
    double a = System::Convert::ToDouble(textBox1->Text);
    double b;
    Double::TryParse(textBox2->Text, b);
    double K = 170/(a+2*b);
    Brush^ brush = gcnew SolidBrush(Color::White);
    g->FillRectangle(brush, 0, 0, 360, 360);
    Pen^ pen2 = gcnew Pen(Color::Red, 2);
    g->DrawEllipse(pen2, 180-(int)(K*a), 180-(int)(K*a),
                         (int)(2*K*a), (int)(2*K*a));
    Pen^ pen = gcnew Pen(Color::Blue, 2);
    double pi = Math::PI;
    int ox = (int)(180+K*((a+b)*cos(0.0)-b*cos(0.0)));
    int oy = (int)(180-K*((a+b)*sin(0.0)-b*sin(0.0)));
    for (double t=0; t<=2*pi*b; t += pi/80) {</pre>
        int nx = (int)(180+K*((a+b)*cos(t)-b*cos((a+b)*t/b)));
        int ny = (int)(180-K*((a+b)*sin(t)-b*sin((a+b)*t/b)));
        g->DrawLine(pen, ox, oy, nx, ny);
        ox = nx;
        oy = ny;
    }
}
  Form1.h ファイルの先頭に
#include <math.h>
  を追加する。
  コンパイルし、実行すると
```

📰 Form1	
	a
	3
	ь
	2
	[button1]

となる。button1 をクリックすると



```
となる。
```
```
int oy = (int)(180-K*((a+b)*sin(0.0)-b*sin(0.0)));
    for (double t=0; t<=2*pi*b; t += pi/80) {</pre>
        int nx = (int)(180+K*((a+b)*cos(t)-b*cos((a+b)*t/b)));
       int ny = (int)(180-K*((a+b)*sin(t)-b*sin((a+b)*t/b)));
       g->DrawLine(pen, ox, oy, nx, ny);
       ox = nx;
       oy = ny;
   }
}
の
    double a = System::Convert::ToDouble(textBox1->Text);
は textBox1 のデータを double の数値として変数 a に代入している。同様に
    double b;
    Double::TryParse(textBox2->Text, b);
も textBox2 のデータを double の数値として変数 b に代入している。
  この図を描く様子をアニメーションで見るには
import pygame
from math import *
pygame.init()
black = (0,0,0)
white = (255, 255, 255)
green = (0, 255, 0)
red= (255, 0, 0)
blue = (0, 0, 255)
pi = 3.141592653
size = (700, 500)
screen = pygame.display.set_mode(size)
pi = atan(1.0)*4
done = False
clock = pygame.time.Clock()
theta = 0
K = 20
a = 5
```

```
b = 2
```

```
while done == False:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done = True
    screen.fill(white)
    pygame.draw.ellipse(screen, green,
        [350-K*a, 250-K*a, 2*K*a, 2*K*a])
    x = (int)(K*((a+b)*cos(theta)-b))
    y = (int)(K*((a+b)*sin(theta)+b))
    pygame.draw.ellipse(screen, blue,[350+x,250-y, 2*K*b, 2*K*b])
    x1 = (int)(350+K*((a+b)*cos(theta)-b*cos((a+b)*theta/b)))
    y1 = (int)(250-K*((a+b)*sin(theta)-b*sin((a+b)*theta/b)))
    x2 = (int)(350+K*((a+b)*cos(theta)+b*cos((a+b)*theta/b)))
    y2 = (int)(250-K*((a+b)*sin(theta)+b*sin((a+b)*theta/b)))
    pygame.draw.line(screen, black, [x1, y1], [x2, y2], 2)
    x1 = (int)(350+K*((a+b)*cos(theta)-b*cos((a+b)*theta/b+pi/2)))
    y1 = (int)(250-K*((a+b)*sin(theta)-b*sin((a+b)*theta/b+pi/2)))
    x2 = (int)(350+K*((a+b)*cos(theta)+b*cos((a+b)*theta/b+pi/2)))
    y2 = (int)(250-K*((a+b)*sin(theta)+b*sin((a+b)*theta/b+pi/2)))
    pygame.draw.line(screen, black, [x1, y1], [x2, y2], 2)
    t = 0
    ox = (int)(350+K*((a+b)*cos(t)-b*cos((a+b)*t/b)))
    oy = (int)(250-K*((a+b)*sin(t)-b*sin((a+b)*t/b)))
    while t <= theta:
        nx = (int)(350+K*((a+b)*cos(t)-b*cos((a+b)*t/b)))
        ny = (int)(250-K*((a+b)*sin(t)-b*sin((a+b)*t/b)))
        pygame.draw.line(screen, red, [ox, oy], [nx, ny], 2)
        ox = nx
        oy = ny
        t += pi/100
    theta += pi/25
    if theta >= 2*pi*b:
        theta = 0
    pygame.display.flip()
    clock.tick(5)
pygame.quit()
とすればよい。
```



レポート問題:原点を中心とし半径が a の円 O の内側を半径が b a > b > 0)の円 C が円 O に内接しながら滑ることなく転がるとき、円 C 上の点 P の軌跡を考える。ただし、点 P のはじめの位置は、円 O と x 軸の正の部分との交点 A とする。

円 C が転がるとき、動径 OC が表す角を θ とし、そのときの外接する点を Q、点 P の座標 e(x,y) とすると、P の軌跡は θ を媒介変数として

$$x = (a - b)\cos\theta + b\cos\frac{a - b}{b}\theta, \quad y = (a - b)\sin\theta - b\sin\frac{a - b}{b}\theta$$

で表される。この曲線を ハイポサイクロイド という。ハイポサイクロイドを描くプログラムを 作れ。



レポート問題:極方程式

 $r = a + b\cos\theta$

で表される曲線を リマソン という。特に、a = bのとき、極方程式

 $r = a(1 + \cos\theta)$

で表される曲線を カージオイド という。リマソンを表示するように、上のプログラムを変更せよ。



レポート問題:

$$x = (a+b)\cos\theta - c\cos\frac{a+b}{b}\theta, \quad y = (a+b)\sin\theta - c\sin\frac{a+b}{b}\theta$$

で表される曲線を描くプログラムを作れ。



レポート問題:

$$x = (a-b)\cos\theta + c\cos\frac{a-b}{b}\theta, \quad y = (a-b)\sin\theta - c\sin\frac{a-b}{b}\theta$$

で表される曲線を描くプログラムを作れ。

🖷 Form1	- 🗆 🗙
	a 7 b 4 c 5

この二つの図は Windows8 での実行結果です。