講義メモ

・オリエンテーション
・開発環境について

p.2 C#でなにができるのか（補足）

・C#は複数の開発形式を提供している
①コンソールアプリケーション：コマンドインターフェース（キーボードからの入力とコンソールへの表示）を用いる基本形態（この講座で利用）
②フォームアプリケーション：画像や動画や音声を用いて、マウスなども利用可能な形態（応用コースで学習）
③サーバアプリケーション：Webサーバ上で動作しインターネット経由でのリクエスト（例：ゲームサーバやメンバーや課金の管理）に対応できる形態
④他のシステムやゲームエンジンと連携する形態（例：Unity）
・C#はWindows、Macなどで利用可能で、Windowsでは.NETフレームワークの一部として提供されているため、実行効率が良く、安全性が高い

p.3 オブジェクト指向のメリット（補足）

・「グループ開発」「再利用」「部品化」を容易にするのがオブジェクト指向であり、現代のゲーム開発では必須のテクニック
・これが初めから言語に含まれているのがC#、Java、C++など

p.6 プログラムができるまで（補足）

・C#の開発環境であるVisual Studioでは、ソースファイルを読み込ませて「ビルド」すると実行用ファイルが生成される（コンパイル（翻訳）＋リンク（結合）＝ビルド）
・また、ビルドは必要時に自動的に行われるので、単に「実行」するだけでOK

p.8 .NET Framework（補足）

・基礎コースの学習においては、詳細を理解する必要はない
・ただし、Visual Studioの新規インストールにおいては「.NET Framework」のチェックをオンにしないとC#は使えないので注意

p.11 Visual Studio 2026 のインストール（補足）

・無料ダウンロード：https://visualstudio.microsoft.com/ja/downloads/
・Communityの「無料ダウンロード」をクリック
・VisualStudioSetup.exe ファイルがダウンロードされるので「ダウンロード」フォルダなどに「保存」
・このファイルを開くとインストール開始
※インストールにおいて「.NET Framework」のチェックをオンにすること
※マイクロソフトのIDを作成しサインインする必要がある

p.11 Visual Studio 2022 の使い方（補足）

①「スタート」「すべて」「V」「Visual Studio 2022」
②「新しいプロジェクトの作成」
③言語を「C#」に変更
④「空のプロジェクト（.NET Framework）」「次へ」
⑤プロジェクト名は自由（共用PCでUSBメモリが無い場合は、名前を含むプロジェクト名に）
⑥場所は自由（共用PCでUSBメモリがある場合はUSBメモリのドライブを、無い場合は「ドキュメント」）
　※デスクトップは推奨しません

⑦「作成」
⑧「新機能」が開いたらタブ内の「×」をクリックして閉じる
⑨(共用PCでは)「ウィンドウ」「ウィンドウレイアウトのリセット」「はい」
　※ソリューションエクスプローラが右側に開けばOK
⑩「プロジェクト(P)」「新しい項目の追加」
　※ここで「すべてのテンプレートの表示」ボタンが表示されたらクリック
⑪「コード」「コードファイル」
⑫「名前」を入力して「追加」
　※最初の1本目は「CodeFile1.cs」を「sample.cs」に書換
　※ソリューションエクスプローラに「sample.cs」が追加されてエディタが開く

p.15 Sample.csを用いて動作確認

①コードを入力

```
//sample.cs
using System;
class Sample
{
   public static void Main()
   {
      Console.WriteLine("Hello World!");
   }
}
```

※対になる文字が自動的に入力されたり、空白が自動的に挿入されたり、ヒントが表示されることがある
※画面上に「0個の参照」と表示されたら「ツール」「オプション」「テキストエディター」「CodeLens」で「CodeLensを有効にする」のチェックをクリックしてオフにすると良い
※赤い波線が表示されていたら、入力ミスがあるので確認を(ただし、ミスの位置にあるとは限らない)
②「ファイル」「すべて保存」、または「Ctrl + Shift+S」
※「ファイル」「sample.csの保存」、または「Ctrl + S」でもOK
③「▷(デバッグなしで開始)」をクリックするとビルドが行われ、コンソールが開いて結果「Hello World!」が表示される
④ 結果を確認したら、必ず、コンソールを閉じる(なにかキーを押すか「×」をクリック)

第2章 テキストを表示する

p.17 初めてのC#プログラム

・すでに作成したコードファイルをもとにして、p.17「myname01.cs」を作る手順
(sample.csを書き換えるのではなく残したい場合)
①「プロジェクト(P)」「新しい項目の追加」
②「コード」「コードファイル」 ※前回の直後であれば選択済
③ 名前を「myname01.cs」にして「追加」
　※名前を間違えて「追加」すると格納されてしまうので、ソリューションエクスプローラで、そのコードファイルを右クリックして「名前の変更」すると変更できる
④「sample.cs」タブをクリックして全文を選択して「編集」「コピー」
　※「ctrl + a」「ctrl + c」すると便利
⑤ ソリューションエクスプローラで「sample.cs」を右クリックして「プロジェクトから除外」
　※コードファイルは消えず、実行対象にならなくなるだけ
　※残っていると、どちらを実行すべきかわからなくなるので実行できない
　※実行用コードファイルが使う部品や画像やデータなどを並べることは可能
⑥「myname01.cs」で「編集」「ペースト」または「ctrl + v」
　※この講義メモからのコピペでもOK
⑦「sample.cs」から変更されている部分のみを書き換える

※「粂井康孝」は好きな名前に変更してOK
⑧「ファイル」「すべて保存」、または「Ctrl + Shift+S」
⑨「▷(デバッグなしで開始)」
⑩ 結果を確認したら、コンソールを閉じる(なにかキーを押すか「×」をクリック)

p.17 mymane01.cs

```
//mymane01.cs
using System;
class MyName01
{
   public static void Main()
  {
      Console.WriteLine("私の名前はシャアです");
   }
}
```

p.18 プログラムの構造①コメント

・1行目の「//mymane01.cs」はコメントと呼ばれるメモ書きで、コンパイルやビルドでは無視される
・よって記述不要だが、業務上は記述すべきコメントが決められていることが多い
・例えば、プログラム名、作成者や権利者、作成日と作成理由、改版日と改版理由(改版者)など
・なお、コメントがまったくないコードファイルは使い捨てのものとみなされる事が多く、他者に見せるコードファイルにはコメントを記述することが望ましい
・コメントは「//」を先頭に入れ、末尾までがコメントになる
・通常の記述の右側にコメントを記述することも可能で、短い説明や注意書きに使われることが多い
  例: class MyName01 //クラス名MyName01は変更しないこと
・C言語の「/* から */ まで」コメント形式も利用可能だが、「//」形式と併用するのは推奨されない
  ※ 業務によってはC言語の「/* から */ まで」コメント形式は非推奨の場合もある

p.18 プログラムの構造②using

・2行目の「using System;」は名前空間の利用を示す前置きで、Systemは名前空間の名前。
・この指定は「このプログラムではSystem名前空間にあるものを使うので、Systemという名前は省略できるようにしてほしい」という意味
・Systemという名前空間には、多くの便利なものが用意されているので、使うことが多い。よって「using System;」は決まり文句的にプログラムの前方に書いておくことが多い。
・なお、このプログラムでは「Console」がSystemという名前空間にあるので、「using System;」を書かなくても「System.Console」とすればエラーにならない
  ※ 業務によっては「using System;」を必須とする場合と、記述不可とする場合がある

p.18 プログラムの構造③クラス

・3行目の「class MyName01」は「クラス名をMyName01とする」という意味で、後続の「{」から「}」までがMyName01クラスの内容になる
・クラスはオブジェクト指向言語ではプログラムを構成する単位で、C#などでは処理はクラスの中に記述する
・通常、クラスの名前はコードファイルの名前に合わせることが多い(大文字小文字を替えることが多い)
・シンプルなプログラムでは、コードファイルが1つ、その中にクラスが1つという形式になる

p.18 プログラムの構造④メソッド

・5行目の「public static void Main()」はメソッドの定義で、後続の「{」から「}」までがMainメソッドの内容になる

・詳細は後述するが、Mainメソッドは「実行開始位置」を示す特別なメソッドで、コンソールアプリケーションの実行用コードファイルに1つだけ必ず記述する
・MainのMは大文字にすること
　※小文字にすると、Mainメソッドなどから呼び出される部品として扱われてしまうので

次回予告：「p.20 プログラムの構造⑤メソッドを呼ぶ」から