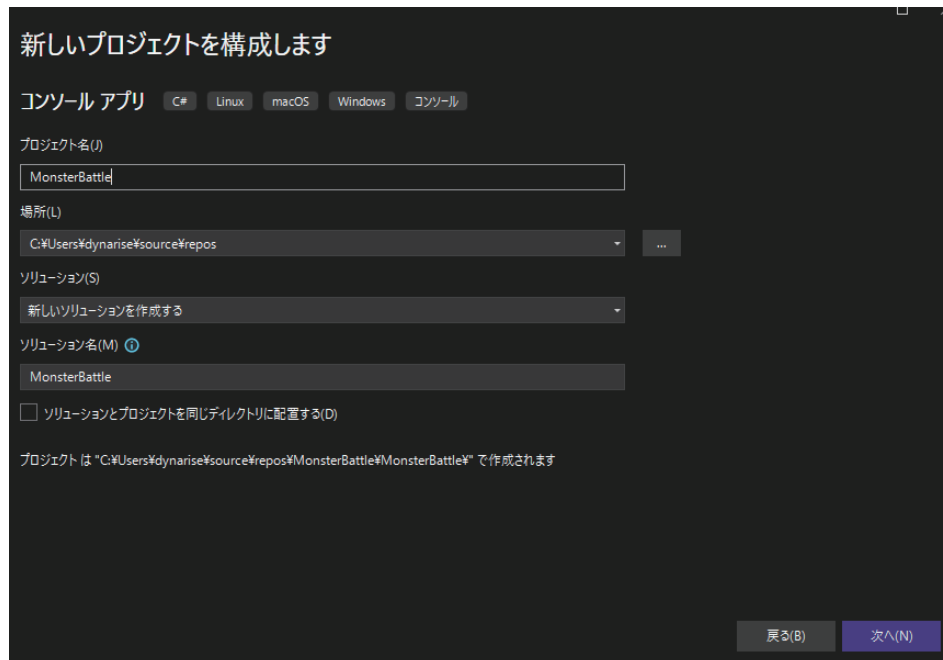


## C# 課題 レベル 4

**条件分岐**を用いて、設定した変数の値によって結果が変わるプログラムをつくります。  
まずは新規プロジェクト「MonsterBattle」を作成して、自動生成された「Program.cs」にプログラムが記述できる準備をしましょう。



Program.cs に **下記の配列データを記述**しましょう。

```
1 string[] charaName = { "ヒノタネ", "ミズトカゲ", "タネガメ", "ヒカッチュウ" }; //名前の整理
2 string[] type = { "ほのお", "みず", "くさ", "かみなり" }; //タイプの整理
3 string[] weakPoint = { type[1], type[2], type[0], type[1] }; //弱点の整理
4 float[] attack = { 10.0f, 15.0f, 20.0f }; //攻撃パワーの整理
5
```

## Program.cs の完成

### ①必要な変数を追加しましょう

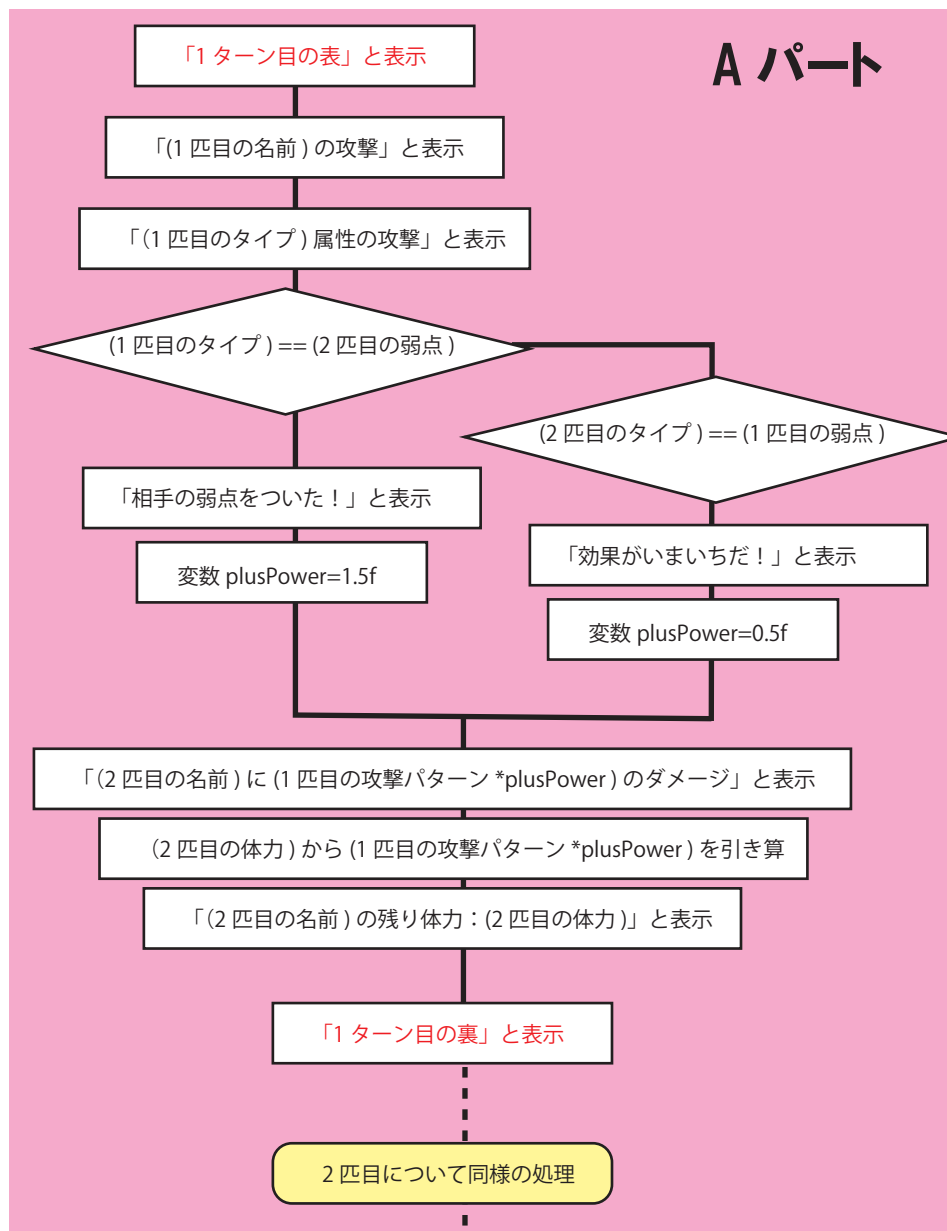
**int 型** monster1 = 0、monster2 = 1  
attackNum1 = 0、 attackNum2 = 0  
**float 型** hp1 = 50.0f、 hp2 = 50.0f、 plusPower = 1.0f

### 説明：

モンスターを 2 匹チョイスして、バトルしようと思います。

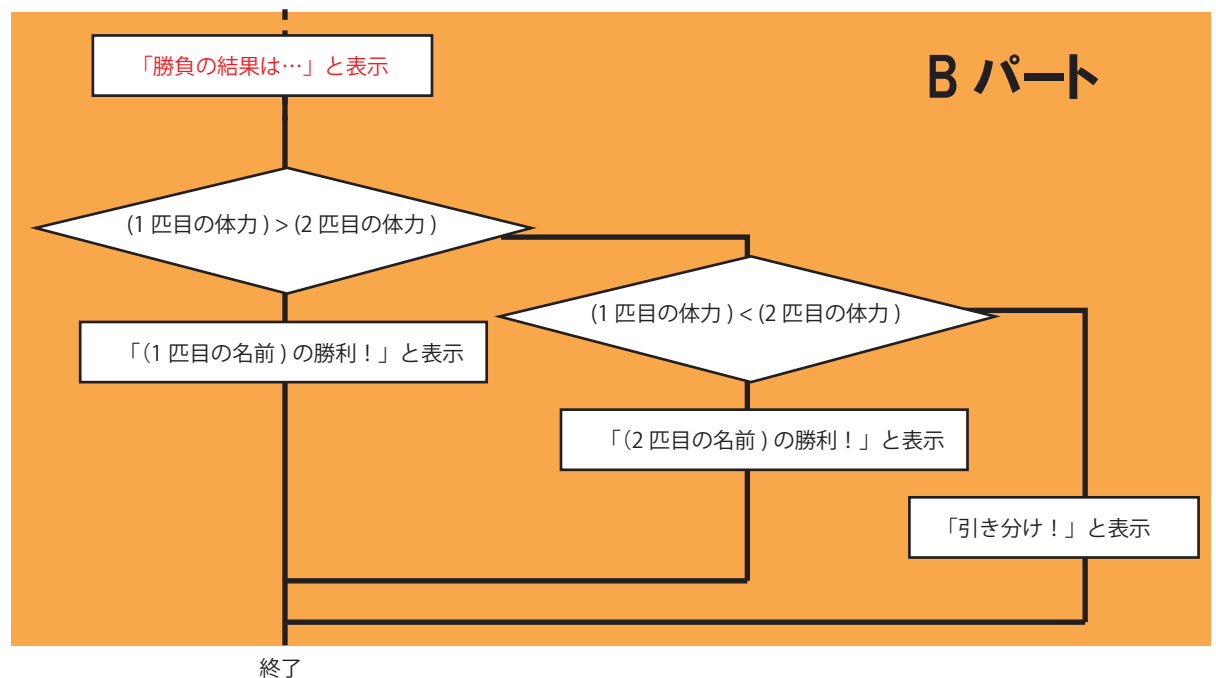
- monster1 は 1 匹目、monster2 は 2 匹目の  
それぞれの名前や属性を選ぶための番号  
(例 charaName[0] → ヒノタネ、その場合 type[0] → ほのおタイプ)
- attackNum1 は 1 匹目、attackNum2 は 2 匹目の攻撃パターンを選ぶための番号  
(例 attack[0] → 攻撃パターン 1 で攻撃力 10.0f)
- hp1、hp2 はモンスターそれぞれの体力
- plusPower は 2 匹のタイプの相性による攻撃力への掛け算  
(相手が得意なタイプなら 1.5 倍、相手が苦手なタイプなら 0.5 倍)

② 作成した配列、変数を駆使して、次のフローチャートや出力結果の見本を頼りにプログラムを組みましょう。



### 出力結果

```
-----1ターン目の表-----
ヒノタネの攻撃
(ほのお属性の攻撃
効果がいまいちだ!
ミストカゲに5のダメージ
ミストカゲの残り体力: 45
-----1ターン目の裏-----
ミストカゲの攻撃
みず属性の攻撃
相手の弱点をついた!
ヒノタネに15のダメージ
ヒノタネの残り体力: 35
勝負の結果は...
ミストカゲの勝利!
```



- ③ ②の A パートを 3 回繰り返してから B パートに進みたいです。  
for 文をつかって繰り返しを実現しましょう。  
その際、ターン数が「1 ターン目」「2 ターン目」「3 ターン目」  
と変化するように注意する事  
モンスターの hp がそれぞれ 100.0f になるよう変更しておきます

#### 出力結果

```
-----1ターン目の表-----  
ヒノタネの攻撃  
ほのお属性の攻撃  
効果がいまいちだ！  
ミズトカゲに5のダメージ  
ミズトカゲの残り体力：45  
-----1ターン目の裏-----  
ミズトカゲの攻撃  
みず属性の攻撃  
相手の弱点をついた！  
ヒノタネに15のダメージ  
ヒノタネの残り体力：35  
-----2ターン目の表-----  
ヒノタネの攻撃  
ほのお属性の攻撃  
効果がいまいちだ！  
ミズトカゲに5のダメージ  
ミズトカゲの残り体力：40  
-----2ターン目の裏-----  
ミズトカゲの攻撃  
みず属性の攻撃  
相手の弱点をついた！  
ヒノタネに15のダメージ  
ヒノタネの残り体力：20  
-----3ターン目の表-----  
ヒノタネの攻撃  
ほのお属性の攻撃  
効果がいまいちだ！  
ミズトカゲに5のダメージ  
ミズトカゲの残り体力：35  
-----3ターン目の裏-----  
ミズトカゲの攻撃  
みず属性の攻撃  
相手の弱点をついた！  
ヒノタネに15のダメージ  
ヒノタネの残り体力：5  
勝負の結果は…  
ミズトカゲの勝利！
```

3 回繰り返し

#### ④ チャレンジ問題

値をランダムにする方法があります（授業では未紹介）

1. まずは Random クラスの変数 rnd を用意し、実体化
2. 変数 rnd の Next( 第一引数、第二引数 ) メソッドによって  
第一引数以上～第二引数未満のランダム値が取得できます。

### 例：monster1 のランダム値、 monster2 のランダム値

```
Random rnd = new Random();  
  
monster1 = rnd.Next(0, charaName.Length);  
monster2 = rnd.Next(0, charaName.Length);  
それぞれ 0 ～ 3 のランダム値を取得している
```

バトルが始まる前に一度だけ：monster1、monster2 のランダム取得  
バトルのくり返し中：attackNum1、attackNum2 が繰り返す度にランダム取得  
をすることで、実行する度に毎回結果が異なるプログラムを作ってみましょう

#### 出力結果

```
-----1ターン目の表-----  
ヒカッチュウの攻撃  
かみなり属性の攻撃  
効果がいまいちだ！  
ミストカゲに5のダメージ  
ミストカゲの残り体力：95  
-----1ターン目の裏-----  
ミストカゲの攻撃  
みず属性の攻撃  
相手の弱点をついた！  
ヒカッチュウに30のダメージ  
ヒカッチュウの残り体力：70  
-----2ターン目の表-----  
ヒカッチュウの攻撃  
かみなり属性の攻撃  
効果がいまいちだ！  
ミストカゲに7.5のダメージ  
ミストカゲの残り体力：87.5  
-----2ターン目の裏-----  
ミストカゲの攻撃  
みず属性の攻撃  
相手の弱点をついた！  
ヒカッチュウに22.5のダメージ  
ヒカッチュウの残り体力：47.5  
-----3ターン目の表-----  
ヒカッチュウの攻撃  
かみなり属性の攻撃  
効果がいまいちだ！  
ミストカゲに5のダメージ  
ミストカゲの残り体力：82.5  
-----3ターン目の裏-----  
ミストカゲの攻撃  
みず属性の攻撃  
相手の弱点をついた！  
ヒカッチュウに30のダメージ  
ヒカッチュウの残り体力：17.5  
勝負の結果は...  
ミストカゲの勝利！
```

**キャラクターは最初にランダムな  
2 匹が決まる**

**ダメージは繰り返しの度にランダム  
( attack のどれが選ばれるか  
attackNum1 と attackNum2 が  
for 文の繰り返しの度に毎回ランダム )**

時間が余ったら  
新しいプロジェクトにて、  
レベル 4 に近い内容を  
クラスを使って再現しましょう。  
( Monster.cs を作成、  
Program.cs で Monster クラスを  
実体化して使う )