



Practice 4-1

FGO-alike console program

Introduction

- Practice Class with simple FGO-alike console program
- What you need to learn before this practice
 - C# Class(Teacher taught in class)
 - UML diagram
 - https://www.tutorialspoint.com/uml/uml_basic_notations.htm

Introduction

- Create a class **Character** with

- Property:

- protected type(string)
 - protected typesStrongTo(string[])
 - protected typesWeakTo(string[])
 - protected health(double)
 - protected attack(double)
 - protected attackBuff(double)
 - protected hitRate(double)

: store character type(ex. Saber, Archer...)

: store list of enemies that this character is strong to(**attack *= 2**)

: store list of enemies that this character is weak to(**attack *= 0.5**)

: store health of this character

: store default attack value of this character

: store attack buff ratio(**attack *= attackBuff**)

: store hitRate, if a **Random.nextDouble > hitRate**, then **this attack count as zero**

Introduction

- Create a class **Character** with
 - Method:
 - `public double attackWithRatio(string enemyType)` : Calculate the attack this character generates
 - `public string getType()` : Get this character's type
 - `public double getHealth()` : Get this character's health
 - `public void setHealth(double health)` : Set this character's health

Introduction

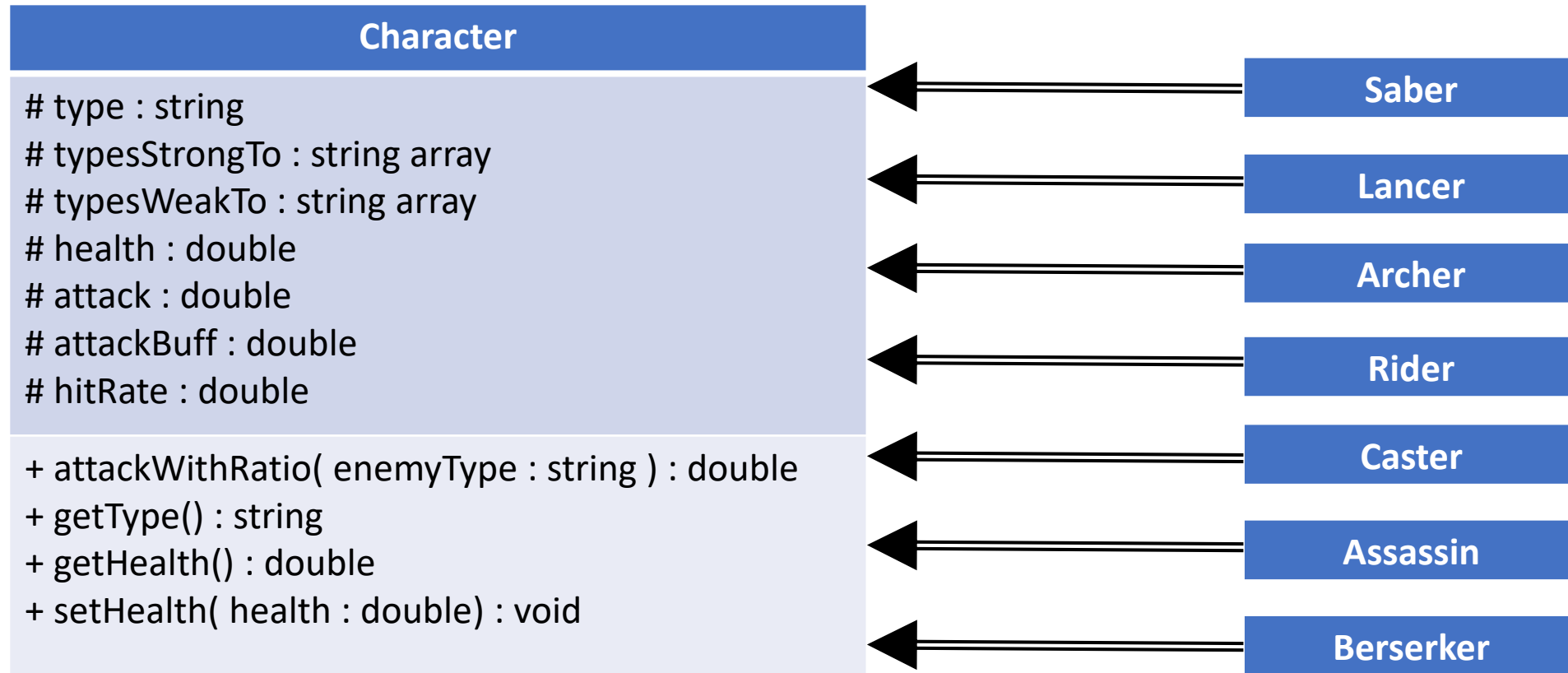
- **Character** can create 7 kinds of object in following: **Saber, Archer, Lancer, Rider, Caster, Assassin, Berserker**
- You should put the following **status in to Character's constructor**, to create one in the 7 kinds of object **when needed**(Next page)

Value of each object

	Saber	Archer	Lancer	Rider	Caster	Assassin	Berserker
type	Saber	Archer	Lancer	Rider	Caster	Assassin	Berserker
typesStrongTo	Lancer, Berserker	Saber, Berserker	Archer, Berserker	Caster, Berserker	Assassin, Berserker	Rider, Berserker	all
typesWeakTo	Archer	Lancer	Saber	Assassin	Rider	Caster	none
health	100	50	75	150	200	50	60
attack	50	100	75	50	25	100	100
attackBuff	1	0.95	1.05	1	0.9	0.9	1.1
hitRate	1	0.5	0.75	0.8	1	1	1

UML Diagram

Put the status in previous page into Character's constructor
To create one in the 7 kinds of object when needed



Procedure

- Enter y to continue the game, n to end the game
- Create two object, player & enemy, both is random from 7 kinds of object
(Hint: Character player = new Character(<one of 7 status from above>))
- Player and enemy attack each other
 - If one's **Random.nextDouble > hitRate**, that one's attack value count as 0
 - Else one's **attack value = attack * (strong or weak) * attackBuff**
- Then show
 - Player & enemy's type
 - Player & enemy's health before attack
 - Player & enemy's attack value
 - Player & enemy's health after attack

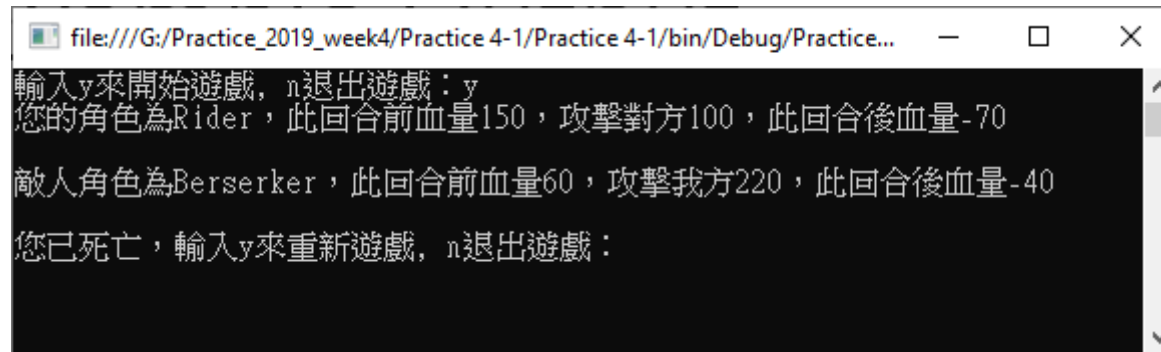
Procedure

- After Attack, depends on these three conditions
 - **Player is alive, enemy is dead :**
 - Player's health > 0 , enemy's health ≤ 0
 - Print 敵人倒下，此回合獲勝，輸入y來繼續遊戲, n退出遊戲：
(Print "Enemy down, you've won this round." Enter y to continue, n to end the game:)
 - **Reset enemy's character, keep the player's character**
 - **Player is dead, don't care about enemy is alive or dead:**
 - Player's health ≤ 0 , enemy's health don't care
 - Print 您已死亡，輸入y來重新遊戲, n退出遊戲：
(Print "You are dead." Enter y to restart, n to end the game :)
 - **Reset both enemy & player's character**
 - **Else player & enemy both alive:**
 - Player's health > 0 , enemy's health > 0
 - Print 雙方均存活，輸入y來進行下一回合，n退出遊戲：
(Print "Both sides survive" Enter y for the next round, n to end the game :)
 - **Keep both enemy & player's character**
- Then go back to the head of previous page

Hint

- Read the PPT carefully before you ask question to TA.
- Understand the Class before this practice.
- Google as much as you can if you have question, so you can improve yourself .
- Discuss with your classmates.
- Actually only about 170 lines codes, don't be afraid.

Example Output



```
file:///G:/Practice_2019_week4/Practice 4-1/Practice 4-1/bin/Debug/Practice...
輸入y來開始遊戲, n退出遊戲: y
您的角色為Rider, 此回合前血量150, 攻擊對方100, 此回合後血量-70
敵人角色為Berserker, 此回合前血量60, 攻擊我方220, 此回合後血量-40
您已死亡, 輸入y來重新遊戲, n退出遊戲:
```

Player is dead

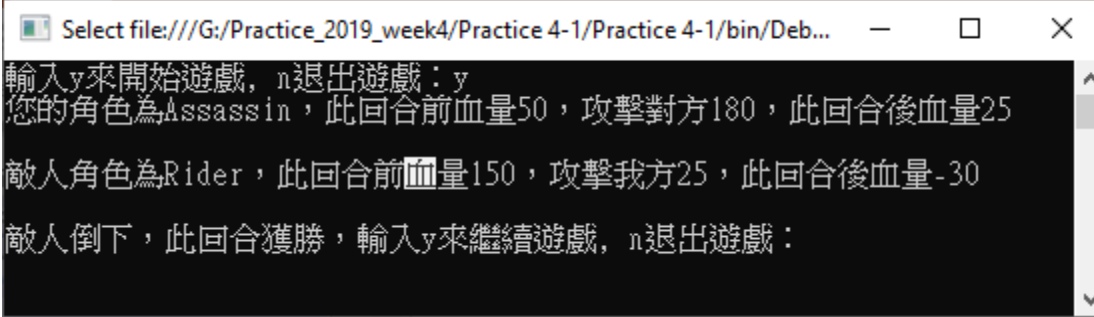
Enter y to start, n to end the game : y

Your character is Rider. Before this round, your health percent is 150. Attacked the other side with 100 . After this round, your health percent is -70.

The enemy is Berserker. Before this round, its health percent is 60. Attacked our side with 220. After this round, its health percent is -40.

You are dead. Enter y to restart, n to end the game :

Example Output

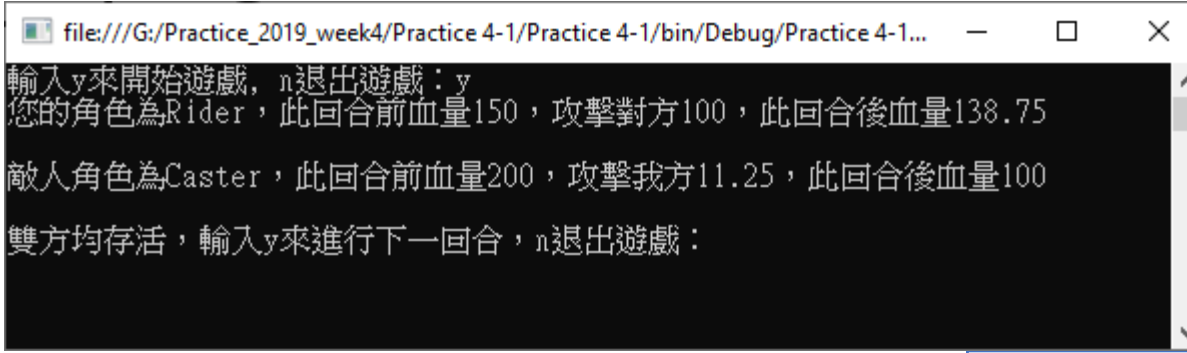


The screenshot shows a terminal window with a title bar that reads "Select file:///G:/Practice_2019_week4/Practice 4-1/Practice 4-1/bin/Deb...". The terminal content is as follows:

```
輸入y來開始遊戲, n退出遊戲: y  
您的角色為Assassin, 此回合前血量50, 攻擊對方180, 此回合後血量25  
敵人角色為Rider, 此回合前血量150, 攻擊我方25, 此回合後血量-30  
敵人倒下, 此回合獲勝, 輸入y來繼續遊戲, n退出遊戲:
```

Below the terminal window, the text "Enemy is Dead" is displayed in a blue-bordered box.

Example Output



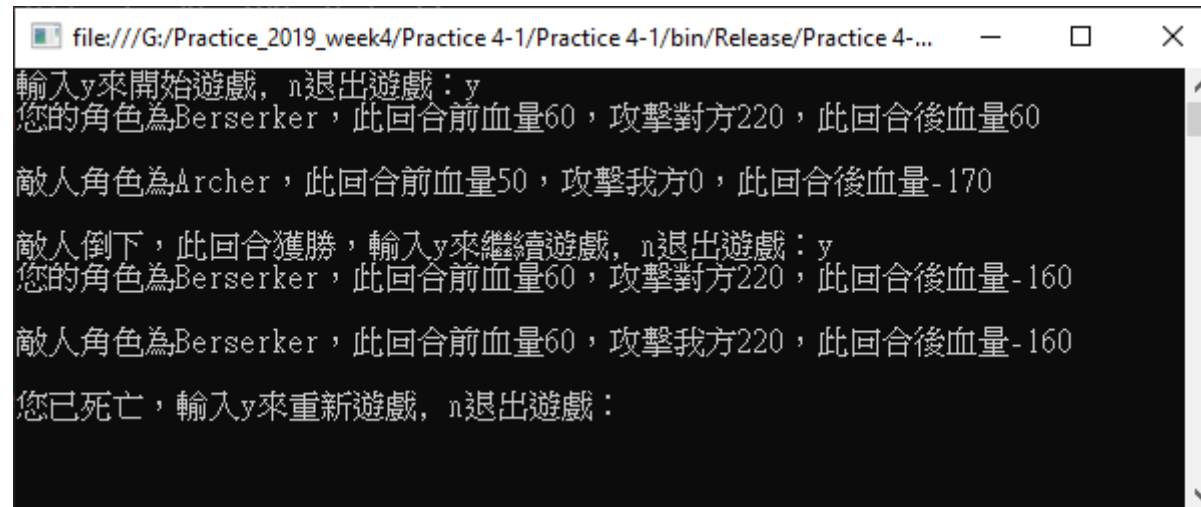
A screenshot of a Windows console window. The title bar shows the file path: file:///G:/Practice_2019_week4/Practice 4-1/Practice 4-1/bin/Debug/Practice 4-1... The console text is as follows:

```
輸入y來開始遊戲，n退出遊戲：y  
您的角色為Rider，此回合前血量150，攻擊對方100，此回合後血量138.75  
敵人角色為Caster，此回合前血量200，攻擊我方11.25，此回合後血量100  
雙方均存活，輸入y來進行下一回合，n退出遊戲：
```

Below the console window, the text "Both Alive" is displayed in a blue-bordered box.

Both Alive

Example Output

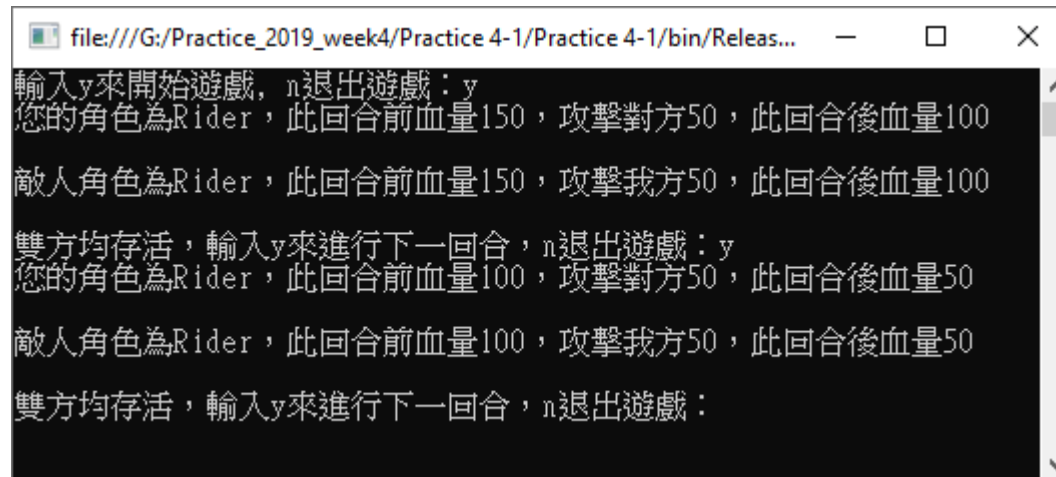
A screenshot of a Windows console window. The title bar shows the file path: file:///G:/Practice_2019_week4/Practice 4-1/Practice 4-1/bin/Release/Practice 4-... The console has a black background with white text. The text shows a game loop where a player character (Berserker) attacks an enemy (Archer). In the first round, the player wins. In the second round, the player wins again but the enemy's status is not reset, leading to a second round where the player's health drops to -160.

```
file:///G:/Practice_2019_week4/Practice 4-1/Practice 4-1/bin/Release/Practice 4-...
輸入y來開始遊戲，n退出遊戲：y
您的角色為Berserker，此回合前血量60，攻擊對方220，此回合後血量60
敵人角色為Archer，此回合前血量50，攻擊我方0，此回合後血量-170
敵人倒下，此回合獲勝，輸入y來繼續遊戲，n退出遊戲：y
您的角色為Berserker，此回合前血量60，攻擊對方220，此回合後血量-160
敵人角色為Berserker，此回合前血量60，攻擊我方220，此回合後血量-160
您已死亡，輸入y來重新遊戲，n退出遊戲：
```

First round player wins, keep player's status, reset enemy

Second round player use the same character and status, enemy use a one character

Example Output

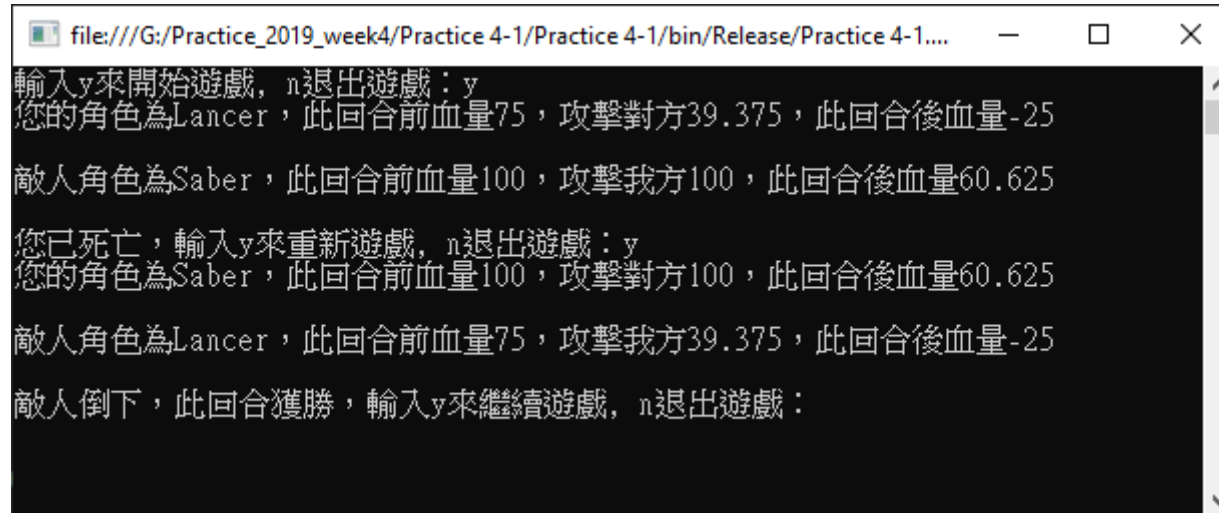
A screenshot of a console window with a black background and white text. The window title bar shows the file path: file:///G:/Practice_2019_week4/Practice 4-1/Practice 4-1/bin/Releas... The text in the console shows a game loop where both player and enemy are 'Rider' and have 100 HP after the first round. The player attacks the enemy, and the enemy attacks the player, but no damage is dealt in this specific example output.

```
file:///G:/Practice_2019_week4/Practice 4-1/Practice 4-1/bin/Releas...
輸入y來開始遊戲，n退出遊戲：y
您的角色為Rider，此回合前血量150，攻擊對方50，此回合後血量100
敵人角色為Rider，此回合前血量150，攻擊我方50，此回合後血量100
雙方均存活，輸入y來進行下一回合，n退出遊戲：y
您的角色為Rider，此回合前血量100，攻擊對方50，此回合後血量50
敵人角色為Rider，此回合前血量100，攻擊我方50，此回合後血量50
雙方均存活，輸入y來進行下一回合，n退出遊戲：
```

First round even, keep player & enemy's status

Second round player & enemy use the same character and status

Example Output



```
file:///G:/Practice_2019_week4/Practice 4-1/Practice 4-1/bin/Release/Practice 4-1....
輸入y來開始遊戲，n退出遊戲：y
您的角色為Lancer，此回合前血量75，攻擊對方39.375，此回合後血量-25
敵人角色為Saber，此回合前血量100，攻擊我方100，此回合後血量60.625
您已死亡，輸入y來重新遊戲，n退出遊戲：y
您的角色為Saber，此回合前血量100，攻擊對方100，此回合後血量60.625
敵人角色為Lancer，此回合前血量75，攻擊我方39.375，此回合後血量-25
敵人倒下，此回合獲勝，輸入y來繼續遊戲，n退出遊戲：
```

First round player is dead, no matter enemy is alive or dead, reset both player & enemy
Second round player & enemy use the same character and status