**Module 03 – Upping Your Game**

Tim Mastarone

Rasmussen University

Microsoft C# Programming

Instructor: Jim Barringer

Module 03 - Lab 2

April 17, 2024

For testing purposes, I moved all the default player creation into a function. I then created 2 new functions for creating a fighter and a wizard to test the new additions of inherited classes. The user is asked to enter the ‘default’, inherited, properties first, then asked to enter the sub-class specific property values. First they are asked to create a ‘Fighter’:

A screenshot of a computer program

Description automatically generated

After entry of the fighter character is completed, the fighter’s information is printed.

The user is then prompted to enter a wizard character:

A screenshot of a computer

Description automatically generated

The program is looping while looking for user input, and checking that the entered information is acceptable before moving on. Below is a screenshot showing several examples of error handling in regard to user input:

A screenshot of a computer

Description automatically generated

Below is the code from my C# application’s Program.cs file (I would like to modularized this a bit more to reduce the re-used code):

class Program

{

static void Main(string[] args)

{

string? userInput;

printGreeting();

createFighter();

createWizard();

void createDefaultCharacter()

{

//instantiate character object

Character playerOne = new Character();

//get information from the user

Print("Enter your character's name");

playerOne.PlayerName = (Console.ReadLine());

playerOne.Age = CheckInt("How old is " + playerOne.PlayerName + "?");

playerOne.Level = CheckInt("Enter the level of " + playerOne.PlayerName);

Print("Enter your character's gender (m/f)");

playerOne.Gender = (Console.ReadLine());

Print("Enter your character's race (Alien, Human, etc.)");

playerOne.PlayerRace = (Console.ReadLine());

Print("Enter your charcater's class (fighter, wizard, etc.)");

playerOne.PlayerClass = (Console.ReadLine());

//output player information to the user

Print("");

Print("Default Player entry completed!");

Print("Player Information:");

Print("Name: " + playerOne.PlayerName);

Print("Age: " + playerOne.Age);

Print("Level: " + playerOne.Level);

Print("Gender: " + playerOne.Gender);

Print("Race: " + playerOne.PlayerRace);

Print("Class: " + playerOne.PlayerClass);

Print("");

}

void createFighter()

{

//Test code to use the sub classes fighter and wizard

Fighter fighterOne = new Fighter();

fighterOne.PlayerClass = "Fighter";

//healing types for fighters

string[] healingTypes = { "Spell", "Hands-On", "Item" };

Print("");

Print("Enter the information for your fighter:");

Print("");

//get information from the user

Print("Enter your fighter's name");

fighterOne.PlayerName = (Console.ReadLine());

fighterOne.Age = CheckInt("How old is " + fighterOne.PlayerName + "?");

fighterOne.Level = CheckInt("Enter the level of " + fighterOne.PlayerName);

Print("Enter your character's gender (m/f)");

fighterOne.Gender = (Console.ReadLine());

Print("Enter your character's race (Alien, Human, etc.)");

fighterOne.PlayerRace = (Console.ReadLine());

Print("Enter the deity the fighter follows:");

fighterOne.Deity = (Console.ReadLine());

Print("Enter your fighter's first Divine Spell:");

fighterOne.DivineSpellOne = (Console.ReadLine());

Print("Enter your fighter's second Divine Spell:");

fighterOne.DivineSpellTwo = (Console.ReadLine());

Print("Enter your fighter's Healing Type:");

for (int i = 0; i < healingTypes.Length; i++)

{

Print(((i + 1).ToString()) + ": " + healingTypes[i]);

}

bool exitLoop = false;

while (!exitLoop)

{

userInput = Console.ReadLine();

if (userInput == "1" || userInput == "2" || userInput == "3")

{

int intUserInput = Int32.Parse(userInput);

fighterOne.HealingType = healingTypes[intUserInput];

exitLoop = true;

}

else

{

Print("Invalid Entry, choose 1, 2, or 3.");

}

}

//output player information to the user

Print("");

Print("Fighter entry completed!");

Print("Fighter Information:");

Print("Name: " + fighterOne.PlayerName);

Print("Age: " + fighterOne.Age);

Print("Level: " + fighterOne.Level);

Print("Gender: " + fighterOne.Gender);

Print("Race: " + fighterOne.PlayerRace);

Print("Class: " + fighterOne.PlayerClass);

Print("Deity: " + fighterOne.Deity);

Print("Divine Spell #1: " + fighterOne.DivineSpellOne);

Print("Divine Spell #2: " + fighterOne.DivineSpellTwo);

Print("Healing Type: " + fighterOne.HealingType);

Print("");

}

void createWizard()

{

//power types for wizards

string[] powerTypes = { "Magic", "Nature", "Demonic" };

Wizard wizardOne = new Wizard();

wizardOne.PlayerClass = "Wizard";

//get information from the user

Print("Enter your wizard's name");

wizardOne.PlayerName = (Console.ReadLine());

wizardOne.Age = CheckInt("How old is " + wizardOne.PlayerName + "?");

wizardOne.Level = CheckInt("Enter the level of " + wizardOne.PlayerName);

Print("Enter your character's gender (m/f)");

wizardOne.Gender = (Console.ReadLine());

Print("Enter your character's race (Alien, Human, etc.)");

wizardOne.PlayerRace = (Console.ReadLine());

Print("Enter your wizard's first Arcane Spell:");

wizardOne.ArcaneSpellOne = (Console.ReadLine());

Print("Enter your wizard's second Arcane Spell:");

wizardOne.ArcaneSpellTwo = (Console.ReadLine());

Print("Enter your wizard's Power Type:");

for (int i = 0; i < powerTypes.Length; i++)

{

Print(((i + 1).ToString()) + ": " + powerTypes[i]);

}

bool exitLoop = false;

while (!exitLoop)

{

userInput = Console.ReadLine();

if (userInput == "1" || userInput == "2" || userInput == "3")

{

int intUserInput = Int32.Parse(userInput);

wizardOne.PowerType = powerTypes[intUserInput];

exitLoop = true;

}

else

{

Print("Invalid Entry, choose 1, 2, or 3.");

}

}

//output player information to the user

Print("");

Print("Wizard entry completed!");

Print("Wizard Information:");

Print("Name: " + wizardOne.PlayerName);

Print("Age: " + wizardOne.Age);

Print("Level: " + wizardOne.Level);

Print("Gender: " + wizardOne.Gender);

Print("Race: " + wizardOne.PlayerRace);

Print("Class: " + wizardOne.PlayerClass);

Print("Arcane Spell #1: " + wizardOne.ArcaneSpellOne);

Print("Arcane Spell #2: " + wizardOne.ArcaneSpellTwo);

Print("Power Source: " + wizardOne.PowerType);

}

//function that return an integer, pass a quesion to the user an a parameter

int CheckInt(string message)

{

while (true)

{

Print(message);

userInput = Console.ReadLine();

if (int.TryParse(userInput, out int result))

{

return result;

}

else

{

Print("Invalid age, enter an integer");

continue;

}

}

}

void Print(string message)

{

Console.WriteLine(message);

}

void printGreeting()

{

Print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Print("Welcome to the Tim's D and D program!");

Print("You will create a character to use in the game.");

Print("The program with gather the player information including what type of character they are.");

Print("When you are done, the information will be output to the terminal.");

Print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

}

}

}

public class Character

{

public int? Age { get; set; }

public string? PlayerName { get; set; }

public int? Level { get; set; }

public string? Gender { get; set; }

public string? PlayerRace { get; set; }

public string? PlayerClass { get; set; }

}

public class Fighter : Character

{

public string? Deity { get; set; }

public string? DivineSpellOne { get; set; }

public string? DivineSpellTwo { get; set; }

public string? HealingType { get; set; }

}

public class Wizard : Character

{

public string? ArcaneSpellOne { get; set; }

public string? ArcaneSpellTwo { get; set; }

public string? PowerType { get; set; }

}