Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Кубанский государственный технологический университет»

(ФГБОУ ВО «КубГТУ»)

Институт компьютерных систем и информационной безопасности

Кафедра информационных систем и программирования

**ОТЧЕТ**

по лабораторной работе №5.

Тема работы: «№ 5. Интерфейсы».

Наименование дисциплины: «Объектно-ориентированное программирование».

Подготовил:

Студент группы 22-КБ-ПИ1

Мальцев М.С.

Проверил доцент

Мурлина В.А.

Краснодар

2023

**Цель работы:** Изучить основные принципы разработки интерфейсов в C#.

**Задание:**

1. **Общее задание для выполнения на паре**

Разработать приложение, реализующее логику компьютерной игры, в которой:

- Имеется базовый класс "Существо" для любого персонажа/существа (с полями "Количество здоровья", "Переносимый вес", "Кошелек", "Инвентарь" и т.д);

- Имеется базовый класс "Предмет" для любого предмета (с полями "Вес", "Цена" и т.д.);

- Интерфейс «Оружие» для предметов, имеющий метод «Получить урон» и возвращающий количество снимаемого за одну атаку здоровья;

- Имеется интерфейс "Боец" с методом "Атаковать", принимающим в качестве аргументов существо, на которое направлена атака и предмет, реализующий интерфейс «Оружие», которым производится атака;

- Имеется интерфейс "Торговец" с методами "Получить"/"Отдать", принимающими в качестве аргументов другого торговца-участника сделки и предмет.

При помощи комбинаций обозначенных сущностей реализовать несколько классов-наследников.

Доработать приложение из общего задания, добавив:

- Интерфейс «Помощь» для предметов, имеющий метод «Получить эффект» и возвращающий количество восстанавливаемого за одну сессию лечения здоровья;

- Интерфейс «Лекарь» с методом «Лечить», принимающим в качестве аргумента существо, на которое направлено лечение, и предмет, реализующий интерфейс «Помощь», которым ведется лечение;

- Интерфейс «Вор» с методом «Получить», имеющий такое же название и сигнатуру, что и в интерфейсе «Торговец», но иную реализацию.

При помощи комбинаций обозначенных сущностей реализовать несколько классов-наследников.

**Код:**

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Imaging;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ConsoleApp1

{

public partial class Form1 : Form

{

private string state;

ArrayList Player\_Items = new ArrayList();

Weapon Sword = new Weapon(10, 200, 30);

Item torch = new Item(1, 30);

Warrior Player;

Weapon Fangs1 = new Weapon(1, 10, 20);

ArrayList Creature1\_items = new ArrayList();

Creature Creature1;

Weapon Fangs2 = new Weapon(1, 8, 15);

ArrayList Creature2\_items = new ArrayList();

Creature Creature2;

HealPotion potion = new HealPotion(30, 1, 30);

bool can\_attack;

bool[] inventory = new bool[6];

Creature Trader = new Creature(20, 1000, 1000, 1000, new ArrayList());

public Form1()

{

InitializeComponent();

TraderInteface.Visible = false;

Player\_Items.Add(Sword);

Player\_Items.Add(torch);

Player\_Items.Add(potion);

inventory[0] = true;

inventory[1] = true;

inventory[2] = true;

Player = new Warrior(20, 400, 300, 600, Player\_Items, 10);

PlayerHealthBar.Text = "\n " + Player.hp\_bar.ToString() + "/" + Player.max\_hp;

PlayerLevel.Text = "Level: " + Player.level.ToString();

PlayerAttack.Text = "Dmg: " + ((Weapon)Player\_Items[0]).damage.ToString();

PlayerGold.Text = "Gold: " + Player.gold;

can\_attack = true;

Creature1\_items.Add(Fangs1);

Creature1 = new Creature(5, 50, 0, 0, Creature1\_items);

NPC1HealthBar.Text = "\n" + Creature1.hp\_bar.ToString() + "/" + Creature1.max\_hp;

NPC1Level.Text = "Level: " + Creature1.level.ToString();

NPC1Attack.Text = "Dmg: " + ((Weapon)Creature1\_items[0]).damage.ToString();

Creature2\_items.Add(Fangs2);

Creature2 = new Creature(6, 70, 0, 0, Creature2\_items);

NPC2HealthBar.Text = "\n" + Creature2.hp\_bar.ToString() + "/" + Creature2.max\_hp;

NPC2Level.Text = "Level: " + Creature2.level.ToString();

NPC2Attack.Text = "Dmg: " + ((Weapon)Creature2\_items[0]).damage.ToString();

}

private void NPC2Picture\_Click(object sender, EventArgs e)

{

if ((state == "Attack")&&can\_attack)

{

var temp = Creature2.hp\_bar;

Player.Hit(Creature2, Sword);

DialogWindow.Text = "Log:\nYou did " + (temp - Creature2.hp\_bar) + " dmg to creature.";

NPC2HealthBar.Text = "\n" + Creature2.hp\_bar.ToString() + "/" + Creature2.max\_hp;

NPC2Level.Text = "Level: " + Creature2.level.ToString();

NPC2Attack.Text = "Dmg: " + ((Weapon)Creature2\_items[0]).damage.ToString();

if (Creature2.hp\_bar < 0)

{

NPC2Interface.Visible = false;

DialogWindow.Text += "\nThis attack was lethal for Wolf. You got levelup!";

Player.level++;

}

else

NPC2HP.Value = (int)((double)(Creature2.hp\_bar) / (double)(Creature2.max\_hp) \* 100);

PlayerHealthBar.Text = "\n " + Player.hp\_bar.ToString() + "/" + Player.max\_hp;

PlayerLevel.Text = "Level: " + Player.level.ToString();

PlayerAttack.Text = "Dmg: " + ((Weapon)Player\_Items[0]).damage.ToString();

PlayerGold.Text = "Gold: " + Player.gold;

can\_attack = false;

state = "";

}

}

private void AttackButton\_Click(object sender, EventArgs e)

{

state = "Attack";

}

private void SkipButton\_Click(object sender, EventArgs e)

{

can\_attack = true;

DialogWindow.Text = "";

if (!(NPC2Interface.Visible) && !(NPC1Interface.Visible))

{

DialogWindow.Text = "\nAfter the fight was over, a stranger approached you. He turned out to be a potions dealer.";

TraderInteface.Visible = true;

TraderHealth.Text = "Health: " + Trader.max\_hp;

TraderAttack.Text = "Dmg: 100";

TraderGold.Text = "Gold: " + Trader.gold;

}

if (NPC1Interface.Visible)

{

var temp = Player.hp\_bar;

Creature1.Hit(Player, Fangs1);

DialogWindow.Text = "\nCreature1 did " + (temp - Player.hp\_bar) + " dmg to you.";

if (Player.hp\_bar > 0)

{

PlayerHealthBar.Text = "\n " + Player.hp\_bar.ToString() + "/" + Player.max\_hp;

PlayerHealth.Value = (int)((double)(Player.hp\_bar) / (double)(Player.max\_hp) \* 100);

}

else

{

PlayerHealthBar.Text = "\n " + Player.hp\_bar.ToString() + "/" + Player.max\_hp;

DialogWindow.Text += "\nIt was lethal. You died.";

MessageBox.Show("You lose");

}

}

if ((Player.hp\_bar > 0) && (NPC2Interface.Visible))

{

var temp = Player.hp\_bar;

Creature2.Hit(Player, Fangs2);

DialogWindow.Text += "\nCreature2 did " + (temp - Player.hp\_bar) + " dmg to you.";

if (Player.hp\_bar > 0)

{

PlayerHealthBar.Text = "\n " + Player.hp\_bar.ToString() + "/" + Player.max\_hp;

PlayerHealth.Value = (int)((double)(Player.hp\_bar) / (double)(Player.max\_hp) \* 100);

}

else

{

PlayerHealthBar.Text = "\n " + Player.hp\_bar.ToString() + "/" + Player.max\_hp;

DialogWindow.Text += "\nIt was lethal. You died.";

MessageBox.Show("You lose");

}

}

}

private void NPC1Picture\_Click(object sender, EventArgs e)

{

if ((state == "Attack") && can\_attack)

{

var temp = Creature1.hp\_bar;

Player.Hit(Creature1, Sword);

DialogWindow.Text = "Log:\nYou did " + (temp - Creature1.hp\_bar) + " dmg to creature.";

NPC1HealthBar.Text = "\n" + Creature1.hp\_bar.ToString() + "/" + Creature1.max\_hp;

NPC1Level.Text = "Level: " + Creature1.level.ToString();

NPC1Attack.Text = "Dmg: " + ((Weapon)Creature2\_items[0]).damage.ToString();

if (Creature1.hp\_bar < 0)

{

NPC1Interface.Visible = false;

DialogWindow.Text += "\nThis attack was lethal for Wolf. You got levelup!";

Player.level++;

PlayerLevel.Text = "Level: " + Player.level.ToString();

PlayerAttack.Text = "Dmg: " + ((Weapon)Player\_Items[0]).damage.ToString();

PlayerGold.Text = "Gold: " + Player.gold;

}

else

NPC1HP.Value = (int)((double)(Creature1.hp\_bar) / (double)(Creature1.max\_hp) \* 100);

can\_attack = false;

state = "";

}

}

private void UseButton\_Click(object sender, EventArgs e)

{

state = "Use";

}

private void Inventory3\_Click(object sender, EventArgs e)

{

if (state == "Use")

{

Player.Heal(Player, potion);

if (Player.hp\_bar > Player.max\_hp)

Player.hp\_bar = Player.max\_hp;

PlayerHealthBar.Text = "\n " + Player.hp\_bar.ToString() + "/" + Player.max\_hp;

PlayerHealth.Value = (int)((double)(Player.hp\_bar) / (double)(Player.max\_hp) \* 100);

inventory[2] = false;

Inventory3.BackgroundImage = ConsoleApp1.Properties.Resources.Empty;

}

}

private void TraderBuy\_Click(object sender, EventArgs e)

{

state = "Buy";

}

private void TradeSlot1\_Click(object sender, EventArgs e)

{

if (state == "Buy")

if (Player.gold > potion.GetPrice())

{

inventory[2] = true;

TradeSlot1.Image = ConsoleApp1.Properties.Resources.Empty;

Inventory3.BackgroundImage = ConsoleApp1.Properties.Resources.potion;

Player.Take(potion, Trader);

TraderGold.Text = "Gold: " + Trader.gold;

PlayerGold.Text = "Gold: " + Player.gold;

}

}

private void TraderSell\_Click(object sender, EventArgs e)

{

state = "Sell";

}

private void Inventory2\_Click(object sender, EventArgs e)

{

if (state == "Sell")

{

if (Trader.gold > torch.GetPrice())

{

Player\_Items.Remove(1);

inventory[1] = false;

Inventory2.BackgroundImage = ConsoleApp1.Properties.Resources.Empty;

TradeSlot2.Image = ConsoleApp1.Properties.Resources.torch;

Player.Give(torch, Trader);

TraderGold.Text = "Gold: " + Trader.gold;

PlayerGold.Text = "Gold: " + Player.gold;

}

}

}

private void StealButton\_Click(object sender, EventArgs e)

{

state = "Steal";

}

private void TraderPicture\_Click(object sender, EventArgs e)

{

if (state == "Steal")

{

Random rnd = new Random();

if (rnd.Next(0, 100) < 40)

{

int temp = Player.gold;

Player.Take(Trader);

DialogWindow.Text = "Log:\nYou stole " + (Player.gold - temp) + " from trader.";

TraderGold.Text = "Gold: " + Trader.gold;

PlayerGold.Text = "Gold: " + Player.gold;

}

else

{

DialogWindow.Text = "Log:\nYou attemp to steal money was spotted so trader left.";

TraderInteface.Visible = false;

}

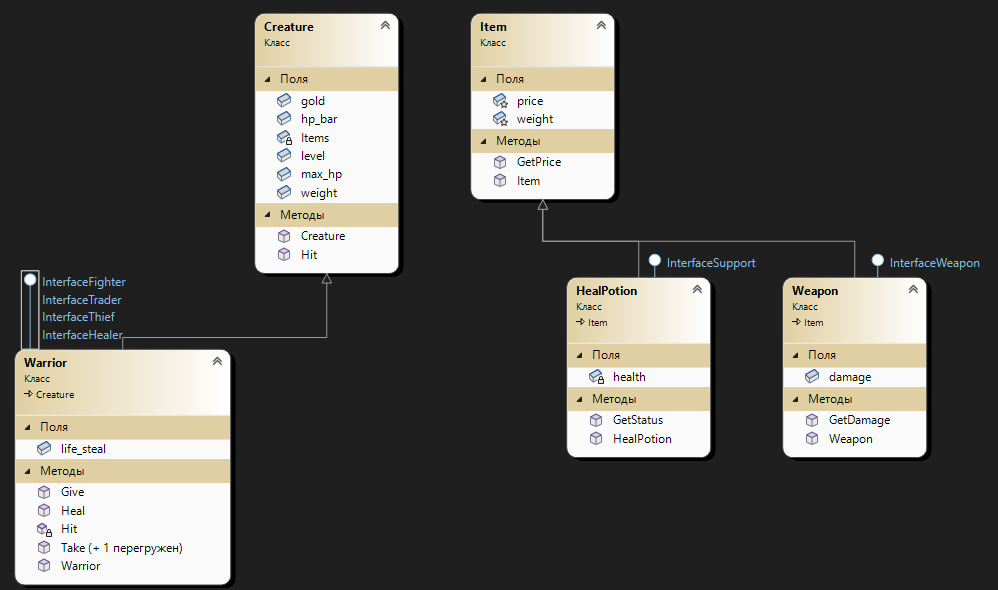
}

}

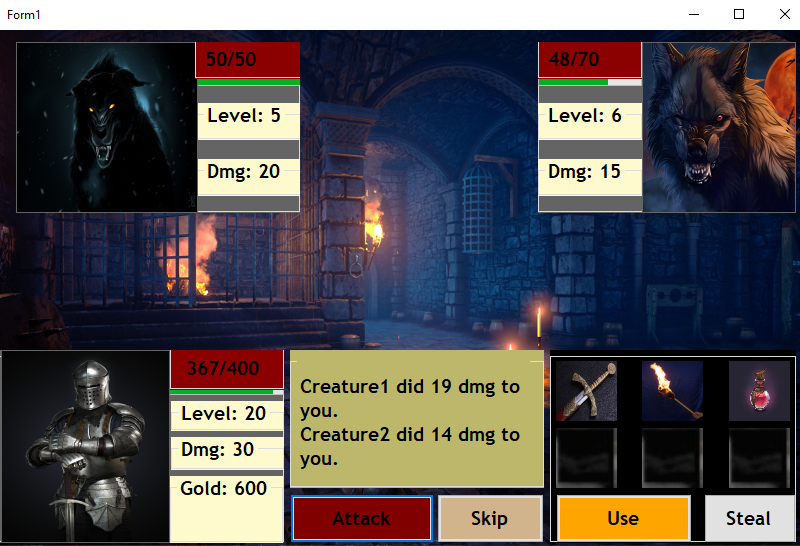
}

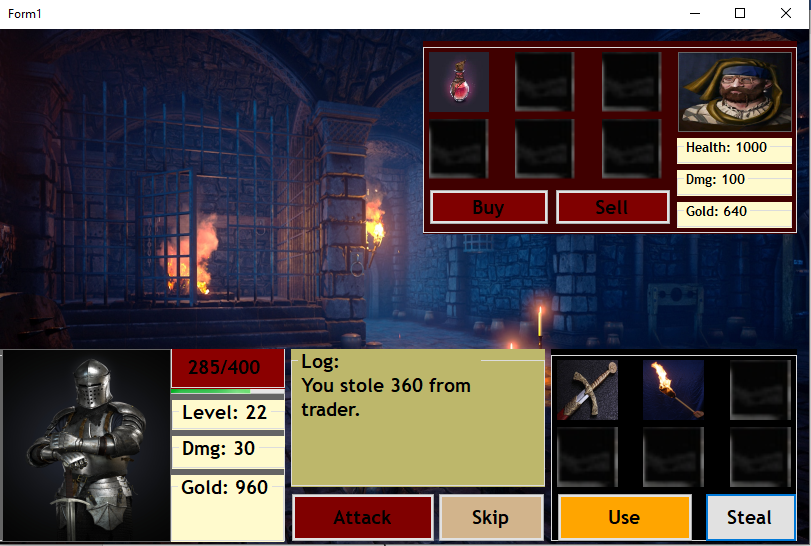
}

**UML-диаграмма:**

****

**Результат выполнения программы:**

****

****

****