|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **윈도우프로그래밍\_과제\_7주** | **이름 :** | **이갑성** | **학번 :** | **20135151** |

* **내용 점검**

|  |
| --- |
| 1. 스레드 |
| using System.Threading; //반드시 포함되어야 함  class ThreadProperty{  public void ThreadBody(){  Thread myself = Thread.CurrentThread;  Monitor.Enter(this); //현재 A쓰레드가 아래의 영역을 실행중일 때 다른 B,C쓰레드가 접근하지 못해서 사용할 수 없다.  for (int i = 1; i <= 3; i++){  Console.WriteLine("{0} is activated => {1}", myself.Name, i);  Thread.Sleep(1000);  }  Monitor.Exit(this); }  }  class Program{  public static void Main(){  ThreadProperty obj = new ThreadProperty();  ThreadStart ts = new ThreadStart(obj.ThreadBody); //멀티쓰레드를 사용한다는 의미  Thread t1 = new Thread(ts);  Thread t2 = new Thread(ts);  t1.Name = "Apple"; t2.Name = "Orange";  t1.Start(); t2.Start();  }  } |
| **[실행결과]** |

|  |
| --- |
| 1. 컬렉션 클래스 - ArrayList |
| static void Main(){  ArrayList list = new ArrayList();  list.Add("generic");  list.Add(45.3);  list.Add(true);  foreach (object s in list){  Console.WriteLine($"value = {s}, type={s.GetType()}");  }  list.Insert(1, "insert");  list.Remove(true);  foreach (object s in list){  Console.WriteLine($"value = {s}");  }  } |
| [실행결과] |

|  |
| --- |
| 1. 컬렉션 클래스 - Hashtable |
| static void Main(){  Hashtable t = new Hashtable();  bool flag = true;  string word, mean;  while (flag){  Console.Write("key >> ");  word = Console.ReadLine();  if (!t.Contains(word)){ //key를 포함하고 있지 않으면  Console.Write("Value >> ");  mean = Console.ReadLine();  t.Add(word, mean); //key, value 값을 해쉬테이블에 저장  //또는 t[word]=mean;  }  else{  flag = false;  }  }    foreach (string s in t.Keys){  Console.WriteLine($"key = {s}, value={t[s]}");  }  } |
| **[실행결과]** |

|  |
| --- |
| 1. 제네릭 컬렉션 클래스 – List & Dictionary |
| class Room{  public Room(int num, string pass){  Number = num;  Pass = pass;  }  public int Number { get; set; }  public string Pass { get; set; }  public override string ToString(){  return "Number = " + Number + ",\tPass = " + Pass;  }}  class Program{  static void Main(){  List<Room> list = new List<Room>();  Dictionary<int, Room> dic = new Dictionary<int, Room>();  list.Add(new Room(201, "s2345")); list.Add(new Room(304, "r345"));  Console.WriteLine("=== 제네릭 List 컬렉션 ===");  foreach (Room s in list){ Console.WriteLine(s.ToString()); }  dic.Add(10, new Room(1002, "e1002")); //인덱서를 사용하여 값을 저장 가능  dic.Add(11, new Room(1102, "e1102"));  dic.Add(12, new Room(1202, "e1202"));  // KeyValuePair : 키와 값을 같이 뽑아내는 기능을 가지는 함수  Console.WriteLine("\n=== key와 value 동시 접근 ===");  foreach (KeyValuePair<int, Room> kvp in dic){ //KeyValuePair은 구조체이다.  Console.WriteLine("Key) {0}, Value) {1}", kvp.Key, kvp.Value.ToString()); }  Console.WriteLine("\n=== key 사용 접근 ===");  foreach (int s in dic.Keys){  Console.WriteLine($"key) {s}, value) {dic[s]}"); }  }  } |
| **[실행결과]** |

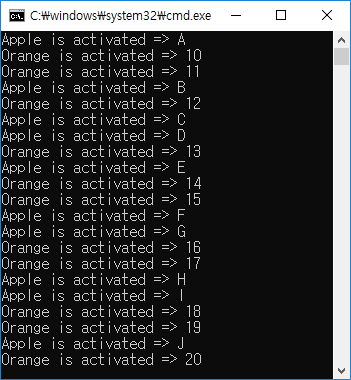
* 프로그램 과제

1. 다음과 같이 실행되는 스레드 프로그램을 작성하시오

* Apple 스레드는 문자 A ~ J 까지 출력 – alpha()
* Orange 스레드는 숫자 10~20까지 출력 – digit()

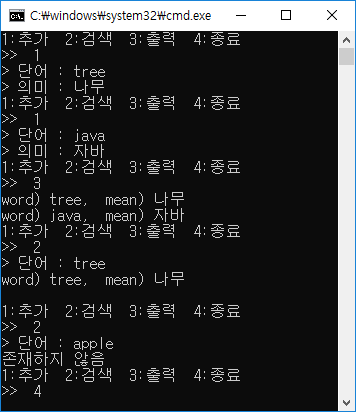
힌트) ThreadStart ts = new ThreadStart(obj.alpha);

ThreadStart ts1 = new ThreadStart(obj.digit);



|  |
| --- |
| **프로그램 소스** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Threading;  namespace HomeWork01  {  class ThreadRun {  public void alpha() // Apple 스레드는 문자 A ~ J 까지 출력  {  Thread myself = Thread.CurrentThread;  for (int i = 65; i < 75; i++)  {  char ch = (char)i;  Console.WriteLine("{0} is activated => {1}", myself.Name, ch);  Thread.Sleep(1000);  }  }  public void digit() //Orange 스레드는 숫자 10~20까지 출력  {  Thread myself = Thread.CurrentThread;  for (int i = 10; i < 21; i++)  {  Console.WriteLine("{0} is activated => {1}", myself.Name, i);  Thread.Sleep(800);  }  }  }  class Program  {  static void Main(string[] args)  {  ThreadRun obj = new ThreadRun();  ThreadStart ts1 = new ThreadStart(obj.alpha);  ThreadStart ts2 = new ThreadStart(obj.digit);  Thread t1 = new Thread(ts1);  Thread t2 = new Thread(ts2);  t1.Name = "Apple";  t2.Name = "Orange";  t1.Start();  t2.Start();  }  }  } |
| **[실행결과]** |

1. 제시된 결과처럼 Dictionary 제네릭 컬렉션 클래스를 사용하여 사전을 구현하는 프로그램을 작성하시오

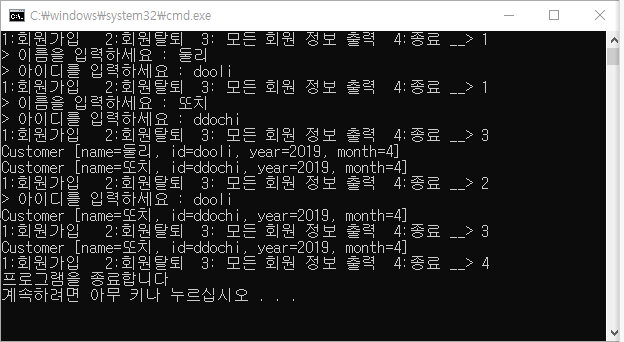


|  |
| --- |
| **프로그램 소스** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace HomeWork02  {  class Memo {  string word, mean; //단어와 뜻  public Memo(string word, string mean){  this.word = word;  this.mean = mean;  }  public string Word{ get { return word; } set { word = value; } }  public string Mean{ get { return mean; } set{ mean = value; } }  public override string ToString()  {  string str = "word) " + Word + ", mean) " + Mean;  return str;  }  }  class Program  {  static void Main(string[] args)  {  Dictionary<int, Memo> dic = new Dictionary<int, Memo>();  string word, mean;  bool flag = true;  int index = 1;    while (flag)  {  Console.Write("1:추가 2:검색 3:출력 4:종료 >>");  switch (Console.ReadLine())  {  case "1":  Console.Write("단어: ");  word = Console.ReadLine();  Console.Write("의미: ");  mean = Console.ReadLine();  dic.Add(index, new Memo(word, mean));  index++;  break;  case "2":    bool find = false;  Console.Write("찾고자하는 단어: ");  word = Console.ReadLine();  foreach (KeyValuePair<int, Memo> kvp in dic)  {  if (kvp.Value.Word == word)  {  Console.WriteLine("의미: {0}", kvp.Value.Mean);  find = true;  }  }  if (find == false)  Console.WriteLine("존재하지 않음");    break;  case "3":  foreach (KeyValuePair<int, Memo> kvp in dic) {  Console.WriteLine("{0}", kvp.Value.ToString());  }  break;  case "4":  flag = false;  break;  }  }  }  }  } |
| **[실행결과]** |

1. List 제네릭 컬렉션 클래스를 사용하여 동아리 회원 가입과 탈퇴를 처리하는 프로그램을 작성하시오.

* 날짜 정보 다루기 : DateTime.Now.Year(현재 년도를 정수형으로 반환하는 프로퍼티),

DateTime.Now.Month(현재 월을 정수형으로 반환하는 프로퍼티)



|  |
| --- |
| **프로그램 소스** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace HomeWork0  {  class Party {  string name, id;  int year, month;  public Party(string name, string id, int y, int m)  {  this.name = name;  this.id = id;  this.year = y;  this.month = m;  }  public string Name {  get {  return name;  }  set {  name = value;  }  }  public string Id {  get  {  return id;  }  set  {  id = value;  }  }  public int Year {  get  {  return year;  }  }  public int Month {  get  {  return month;  }  }  public override string ToString()  {  string str = "Cusomer [name=" + Name + ", id=" + Id + ", year=" + Year + ", month=" + Month + "]";  return str;  }  }  class Program  {  static void Main(string[] args)  {  List<Party> list = new List<Party>();  string name, id;  int m, y;  bool flag = true;    while (flag) {  Console.Write("1:회원가입 2:회원탈퇴 3:모든 회원 정보 출력 4:종료 >>");  switch (Console.ReadLine())  {  case "1":  Console.Write("이름을 입력하세요: ");  name = Console.ReadLine();  Console.Write("아이디를 입력하세요: ");  id = Console.ReadLine();  m = DateTime.Now.Month;  y = DateTime.Now.Year;  list.Add(new Party(name, id, m, y));  break;  case "2":  Console.Write("아이디를 입력하세요: ");  id = Console.ReadLine();  foreach (Party p in list)  {  if (p.Id == id)  {  list.Remove(p);  break;  }  }  break;  case "3":  foreach (Party p in list)  {  Console.WriteLine("{0}", p.ToString());  }  break;  case "4":  flag = false;  break;  }  }    }  }  } |
| **[실행결과]** |