Bài 1

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MainData;

namespace AnimalList
{
    class Program
    {
        // static void Main(string[] args)
        // {
        //     Dog cho = new Dog ("Pin",5,30,12);
        //     cho.Displayinfo();
        //     Console.WriteLine(cho.ToString());

        //     Cat meow = new Cat ("monleo",3,20,7);
        //     meow.Displayinfo();
        //     Console.WriteLine(meow.ToString());

        //     Console.ReadLine();
        // }

        static void Main(string[] args)
        {
            int nDog = 3;
            int nCat = 2;

            Dog cho;
            Cat meo;
            object[] obj = new object[nDog + nCat];

            for (int i = 0; i < nDog; i++)
            {
                cho = new Dog();
                cho.InputInfo();
                obj[i] = (Dog)cho;
            }

            for (int i = 0; i < nCat; i++)
            {
                meo = new Cat();
                meo.InputInfo();
                obj[nDog + i] = (Cat)meo;
            }

            for (int i = 0; i < nDog + nCat; i++)
            {
                if (obj[i].GetType() == typeof(Dog))
                {
                    Console.WriteLine("Information of Dog: ");
                    cho = (Dog)obj[i];
                    cho.Displayinfo();
                }
```

```
                else
                {
                    Console.WriteLine("Information of Cat: ");
                    meo = (Cat)obj[i];
                    meo.Displayinfo();
                }
            }
            Console.ReadLine();
        }
    }
}

namespace MainData
{
    public class Dog
    {
        private string Name = "Not Assigned";
        private int Age = 0;
        private float Height = 0;
        private float Weight = 0;


        public Dog(string Name = "", int Age = 0, float Height = 0, float Weight = 0)
        {
            this.Name = Name;
            this.Age = Age;
            this.Height = Height;
            this.Weight = Weight;
        }

        public void Displayinfo()
        {
            Console.WriteLine("Name = {0}, Age = {1}, Height = {2}, Weight = {3}\n",
Name, Age, Height, Weight);
        }

        public void InputInfo()
        {
            Console.Write("Input Dog Name: ");
            Name = Console.ReadLine();
            InputAge();
            InputHeight();
            InputWeight();
        }
        public void InputAge()
        {
            bool isCompleted = false;

            Console.Write("Input Dog Age: ");
            string strAge = Console.ReadLine();
            try
            {
                Age = int.Parse(strAge);
                if (Age <= 0 || Age > 20)
                {
                    throw new NegativeNumException();
                }
                isCompleted = true;
```

```
        }
        catch (FormatException)
        {
            Console.Write("Not input a Number. Please reinput a Number\n");
        }
        catch (NegativeNumException)
        {
            Console.Write("Negative is not accepted. Please reinput a Number\n");
        }
        finally
        {
            if (!isCompleted)
                InputAge();
        }
    }

    public void InputHeight()
    {
        bool isCompleted = false;
        Console.Write("Input Dog Height: ");
        string strHeight = Console.ReadLine();
        try
        {
            Height = int.Parse(strHeight);
            if (Height <= 0)
            {
                throw new NegativeNumException();
            }
            isCompleted = true;
        }
        catch (FormatException)
        {
            Console.Write("Not input a Number. Please reinput a Number\n");
        }
        catch (NegativeNumException)
        {
            Console.Write("Negative is not accepted. Please reinput a Number\n");
        }
        finally
        {
            if (!isCompleted)
                InputHeight();
        }
    }

    public void InputWeight()
    {
        bool isCompleted = false;
        Console.Write("Input Dog Weight: ");
        string strWeight = Console.ReadLine();
        try
        {
            Weight = int.Parse(strWeight);
            if (Weight <= 0)
            {
                throw new NegativeNumException();
            }
            isCompleted = true;
```

```
            }
            catch (FormatException)
            {
                Console.Write("Not input a Number. Please reinput a Number\n");
            }
            catch (NegativeNumException)
            {
                Console.Write("Negative is not accepted. Please reinput a Number\n");
            }
            finally
            {
                if (!isCompleted)
                    InputWeight();
            }
        }
    }

    public class Cat
    {
        private string Name = "Not Assigned";
        private int Age = 0;
        private float Height = 0;
        private float Weight = 0;


        public Cat(string Name = "", int Age = 0, float Height = 0, float Weight = 0)
        {
            this.Name = Name;
            this.Age = Age;
            this.Height = Height;
            this.Weight = Weight;
        }

        public void Displayinfo()
        {
            Console.WriteLine("Name = {0}, Age = {1}, Height = {2}, Weight = {3}\n",
Name, Age, Height, Weight);
        }
        public void InputInfo()
        {
            Console.Write("Input Cat Name: ");
            Name = Console.ReadLine();
            InputAge();
            InputHeight();
            InputWeight();
        }
        public void InputAge()
        {
            bool isCompleted = false;

            Console.Write("Input Cat Age: ");
            string strAge = Console.ReadLine();
            try
            {
                Age = int.Parse(strAge);
                if (Age <= 0 || Age > 20)
                {
                    throw new NegativeNumException();
```

```
            }
            isCompleted = true;
        }
        catch (FormatException)
        {
            Console.Write("Not input a Number. Please reinput a Number\n");
        }
        catch (NegativeNumException)
        {
            Console.Write("Negative is not accepted. Please reinput a Number\n");
        }
        finally
        {
            if (!isCompleted)
                InputAge();
        }
    }

    public void InputHeight()
    {
        bool isCompleted = false;
        Console.Write("Input Cat Height: ");
        string strHeight = Console.ReadLine();
        try
        {
            Height = int.Parse(strHeight);
            if (Height <= 0)
            {
                throw new NegativeNumException();
            }
            isCompleted = true;
        }
        catch (FormatException)
        {
            Console.Write("Not input a Number. Please reinput a Number\n");
        }
        catch (NegativeNumException)
        {
            Console.Write("Negative is not accepted. Please reinput a Number\n");
        }
        finally
        {
            if (!isCompleted)
                InputHeight();
        }
    }

    public void InputWeight()
    {
        bool isCompleted = false;
        Console.Write("Input Cat Weight: ");
        string strWeight = Console.ReadLine();
        try
        {
            Weight = int.Parse(strWeight);
            if (Weight <= 0)
            {
                throw new NegativeNumException();
```

```
                }
                isCompleted = true;
            }
            catch (FormatException)
            {
                Console.Write("Not input a Number. Please reinput a Number\n");
            }
            catch (NegativeNumException)
            {
                Console.Write("Negative is not accepted. Please reinput a Number\n");
            }
            finally
            {
                if (!isCompleted)
                    InputWeight();
            }
        }
    }
    public class NegativeNumException : Exception
    {
        public NegativeNumException() { }
        public NegativeNumException(string message) : base(message) { }
    }
}
```

```
PS D:\HCMUS\Computer Science\.NET\Bài Tập\Assignments of Labs\Lab01\AnimalList> dotnet run
Input Dog Name: lulu
Input Dog Age: 23
Negative is not accepted. Please reinput a Number
Input Dog Age: -1
Negative is not accepted. Please reinput a Number
Input Dog Age: 7
Input Dog Height(cm): 28
Input Dog Weight(kg): 15
Input Dog Name: pin
Input Dog Age: 9
Input Dog Height(cm): 30
Input Dog Weight(kg): 17
Input Cat Name: pu
Input Cat Age: 2
Input Cat Height: 11
Input Cat Weight: 12
Input Cat Name: pu cha bu
Input Cat Age: 4
Input Cat Height: 4
Input Cat Weight: 14
Information of Dog:
Name = lulu, Age = 7, Height = 28, Weight = 15

Information of Dog:
Name = na, Age = 4, Height = 20, Weight = 4

Information of Dog:
Name = pin, Age = 9, Height = 30, Weight = 17

Information of Cat:
Name = pu, Age = 2, Height = 11, Weight = 12

Information of Cat:
Name = pu cha bu, Age = 4, Height = 4, Weight = 14
```

Bài 2

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MainData;

namespace SportsList
{
    class Program
    {
        static void Main(string[] args)
        {
            // Tennis tns = new Tennis();
            // Football bongda = new Football();
            // Volleyball bongchuyen = new Volleyball();

            //int nTennis = 0, nFootball = 0, nVolleyball = 0;
            Tennis tns;
            Football bongda;
            Volleyball bongchuyen;

            Console.Write("Enter the number of clubs Tennis: ");
            int nTennis = Convert.ToInt16(Console.ReadLine());

            Console.Write("Enter the number of clubs Football: ");
            int nFootball = Convert.ToInt16(Console.ReadLine());
            Console.Write("Enter the number of clubs Volleyball: ");
            int nVolleyball = Convert.ToInt16(Console.ReadLine());
            int sum = nTennis + nFootball + nVolleyball;

            object[] obj = new object[sum];
            for (int i = 0; i < nTennis; i++)
            {
                tns = new Tennis();
                obj[i] = (Tennis)tns;
            }
            for (int i = 0; i < nFootball; i++)
            {
                bongda = new Football();
                obj[nTennis + i] = (Football)bongda;
            }
            for (int i = 0; i < nVolleyball; i++)
            {
                bongchuyen = new Volleyball();
                obj[nTennis + nFootball + i] = (Volleyball)bongchuyen;
            }
            Console.WriteLine("\n---------------------------");
            for (int i = 0; i < sum; i++)
            {
                if (obj[i].GetType() == typeof(Tennis))
                {
                    Console.WriteLine("\nInformation of Tennis: ");
                    tns = (Tennis)obj[i];
                    tns.Displayinfo();
```

```
                }
                else if (obj[i].GetType() == typeof(Football))
                {
                    Console.WriteLine("\nInformation of Football: ");
                    bongda = (Football)obj[i];
                    bongda.Displayinfo();
                }
                else
                {
                    Console.WriteLine("\nInformation of Volleyball: ");
                    bongchuyen = (Volleyball)obj[i];
                    bongchuyen.Displayinfo();
                }
            }
            Console.ReadLine();
        }
    }
}

namespace MainData
{
    public class Sport //lop cha
    {
        protected string namegr = "Not Assigned";
        protected int amongmember = 0;
        protected float time = 0;
        protected string typeball = "Not Assigned";

        public Sport(string namegr = "", int amongmember = 0, float time = 0, string
typeball = "")
        {
            this.namegr = namegr;
            this.amongmember = amongmember;
            this.time = time;
            this.typeball = typeball;
        }
        public void Displayinfo()
        {
            Console.WriteLine("Name Guild: {0}\nNumber of Members: {1}\nTime play:
{2}\nTypeball: {3}", namegr, amongmember, time, typeball);
        }
        public virtual void InputInfo()
        {
            //Console.Write(Environment.NewLine); //xuong dong moi
            Console.Write("Input Name Guild: ");
            namegr = Console.ReadLine();
            InputAmongMembers();
            InputTimePlay();
        }
        public virtual void InputAmongMembers()
        {
            bool isCompleted = false;
            Console.Write("Input Number of Members: ");
            string strAmongMember = Console.ReadLine();
            try
            {
                amongmember = int.Parse(strAmongMember);
                if (amongmember <= 0)
```

```
                {
                    throw new NegativeNumException();
                }
                isCompleted = true;
            }
            catch (FormatException)
            {
                Console.Write("Not input a Number. Please reinput a Number\n");
            }
            catch (NegativeNumException)
            {
                Console.Write("Negative is not accepted. Please reinput a Number\n");
            }
            finally
            {
                if (!isCompleted)
                    InputAmongMembers();
            }
        }
        public virtual void InputTimePlay()
        {
            bool isCompleted = false;
            Console.Write("Input Time Play: ");
            string strTimePay = Console.ReadLine();
            try
            {
                time = float.Parse(strTimePay);
                if (time <= 0)
                {
                    throw new NegativeNumException();
                }
                isCompleted = true;
            }
            catch (FormatException)
            {
                Console.Write("Not input a Number. Please reinput a Number\n");
            }
            catch (NegativeNumException)
            {
                Console.Write("Negative is not accepted. Please reinput a Number\n");
            }
            finally
            {
                if (!isCompleted)
                    InputTimePlay();
            }
        }
    }
    public class Tennis : Sport//cac lop con
    {
        public Tennis(string namegr = "", int amongmember = 0, float time = 0) :
base(namegr, amongmember, time)
        {
            Console.Write(Environment.NewLine);
            typeball = "Tennis";
            Console.WriteLine("Typeball: {0}", typeball);
            InputInfo();
            // Displayinfo();
```

```
        }
        public override void InputAmongMembers()
        {
            bool isCompleted = false;
            Console.Write("Input Number of Members: ");
            string strAmongMember = Console.ReadLine();
            try
            {
                amongmember = int.Parse(strAmongMember);
                if (amongmember <= 0 || amongmember > 2)
                {
                    throw new NegativeNumException();
                }
                isCompleted = true;
            }
            catch (FormatException)
            {
                Console.Write("Not input a Number. Please reinput a Number\n");
            }
            catch (NegativeNumException)
            {
                Console.Write("Negative is not accepted. Please reinput a Number\n");
            }
            finally
            {
                if (!isCompleted)
                    InputAmongMembers();
            }
        }
    }
    public class Football : Sport
    {
        public Football(string namegr = "", int amongmember = 0, float time = 0) :
base(namegr, amongmember, time)
        {
            Console.Write(Environment.NewLine);
            typeball = "Football";
            Console.WriteLine("Typeball: {0}", typeball);
            InputInfo();
            // Displayinfo();
        }
        public override void InputAmongMembers()
        {
            bool isCompleted = false;
            Console.Write("Input Number of Members: ");
            string strAmongMember = Console.ReadLine();
            try
            {
                amongmember = int.Parse(strAmongMember);
                if (amongmember < 11 || amongmember > 22)
                {
                    throw new NegativeNumException();
                }
                isCompleted = true;
            }
            catch (FormatException)
            {
                Console.Write("Not input a Number. Please reinput a Number\n");
```

```
            }
            catch (NegativeNumException)
            {
                Console.Write("Negative is not accepted. Please reinput a Number\n");
            }
            finally
            {
                if (!isCompleted)
                    InputAmongMembers();
            }
        }
    }
    public class Volleyball : Sport
    {
        public Volleyball(string namegr = "", int amongmember = 0, float time = 0) :
base(namegr, amongmember, time)
        {
            Console.Write(Environment.NewLine);
            typeball = "Volleyball";
            Console.WriteLine("Typeball: {0}", typeball);
            InputInfo();
            // Displayinfo();
        }
        public override void InputAmongMembers()
        {
            bool isCompleted = false;
            Console.Write("Input Number of Members: ");
            string strAmongMember = Console.ReadLine();
            try
            {
                amongmember = int.Parse(strAmongMember);
                if (amongmember < 6 || amongmember > 12)
                {
                    throw new NegativeNumException();
                }
                isCompleted = true;
            }
            catch (FormatException)
            {
                Console.Write("Not input a Number. Please reinput a Number\n");
            }
            catch (NegativeNumException)
            {
                Console.Write("Negative is not accepted. Please reinput a Number\n");
            }
            finally
            {
                if (!isCompleted)
                    InputAmongMembers();
            }
        }
    }
    public class NegativeNumException : Exception
    {
        public NegativeNumException() { }
        public NegativeNumException(string message) : base(message) { }
    }
}
```

```
PS D:\HCMUS\Computer Science\.NET\Bài Tập\Assignments of Labs\Lab01\SportsList> dotnet run
Enter the number of clubs Tennis: 1
Enter the number of clubs Football: 2
Enter the number of clubs Volleyball: 1

Typeball: Tennis
Input Name Guild: hcmus
Input Number of Members: 2
Input Time Play: 1

Typeball: Football
Input Name Guild: hcmus
Input Number of Members: 3
Negative is not accepted. Please reinput a Number
Input Number of Members: 19
Input Time Play: 2

Typeball: Football
Input Name Guild: uit
Input Number of Members: 12
Input Time Play: 1,7

Typeball: Volleyball
Input Name Guild: uel
Input Number of Members: 8
Input Time Play: 2

--------------------------

Information of Tennis:
Name Guild: hcmus
Number of Members: 2
Time play: 1
Typeball: Tennis
```

```
Typeball: Football
Input Name Guild: uit
Input Number of Members: 12
Input Time Play: 1,7

Typeball: Volleyball
Input Name Guild: uel
Input Number of Members: 8
Input Time Play: 2

--------------------------

Information of Tennis:
Name Guild: hcmus
Number of Members: 2
Time play: 1
Typeball: Tennis

Information of Football:
Name Guild: hcmus
Number of Members: 19
Time play: 2
Typeball: Football

Information of Football:
Name Guild: uit
Number of Members: 12
Time play: 17
Typeball: Football

Information of Volleyball:
Name Guild: uel
Number of Members: 8
Time play: 2
Typeball: Volleyball
```