```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Remoting.Messaging;
using System.Text;
using System.Threading.Tasks;

namespace TEST_MAP_SIMPLU_2025
{
    public class Person
    {
        private UInt32 cod_pers;
        private String num_pers;
        public Person() { } // inherited by "base"
        public Person(UInt32 cod_pers_1, String num_1)
        {
        this.cod_pers = cod_pers_1;
        this.num_pers = num_1;
        }
        public UInt32 Cod_pers //
        { get { return cod_pers; } set { cod_pers = value; } }

        public String Num_pers
        {
            get => num_pers; set=> num_pers = value;
        }
        public override string ToString()
        {
            return String.Format("Datele sunt {0} si {1}", this.cod_pers, this.num_pers );
        }
    }

    public class Student : Person
    {
        private UInt16 nr_matricol;
        private String Facultate;
        private String Specializare;
        private UInt16 an_studii;
        public Student(): base() { }

        public Student (UInt32 cod_pers, String numpers, ushort nr_matricol1, string facultate1, string specializare1, ushort an_studii1):
        base(cod_pers,numpers)
        {
            this.nr_matricol = nr_matricol1;
            Facultate = facultate1;
            Specializare = specializare1;
            this.an_studii = an_studii1;
        }

        // continuati cu proprietati pentru toate atributele din clasa Student
        // eu am sa fac doar pentru specializare

        public String Specializare1
        {
            get => this.Specializare; set => this.Specializare = value;
        }
    }
    internal class Program
    {
        static void Main(string[] args)
        {

            Person p1 = new Person();
            Person p2 = new Person(1, "Alex");

            ArrayList list_pers = new ArrayList();

            list_pers.Add(p1);
            list_pers.Add(new Person(2, "John"));
            list_pers.Add(new Person(3, "Mark"));


            Console.WriteLine("Datele sunt {0} si {1}", p2.Cod_pers, p2.Num_pers);

            foreach (Person p in list_pers) {
                Console.WriteLine(p.ToString());// ut. supraincarcarea metodei ToString()

            }
            Console.ReadKey();

            Student student1 = new Student();

            Student student2 = new Student(1, "Alex", 1000, "Inginerie", "IT", 4);

            ArrayList list_stud = new ArrayList();

            list_stud.Add(student2);
            list_stud.Add(new Student(2, "Geo", 1002, "Stiinte", "Info", 2));
            list_stud.Add(new Student(3, "John", 1003, "Inginerie", "IT", 4));

            foreach (Student student in list_stud)
            {
                Console.WriteLine("Student {0} {1}",student.Num_pers, student.Specializare1);
            }
            Console.ReadKey();
        }
    }
}

```