Министерство образования Республики Беларусь

Учреждение образования

Брестский государственный технический университет

Кафедра ИИТ

Лабораторная работа №5

За 5 семестр

По дисциплине «Современные платформы программирования»

Выполнил: студент 3 курса

Группы ПО-4(2)

Коташевич С.Н.

Проверил: Монтик Н. С.

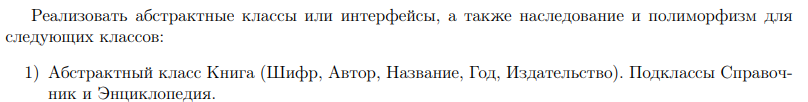
Брест 2021

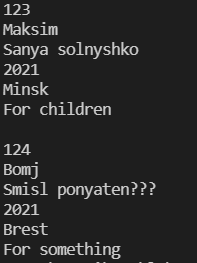
Лабораторная работа №5

Вариант 1

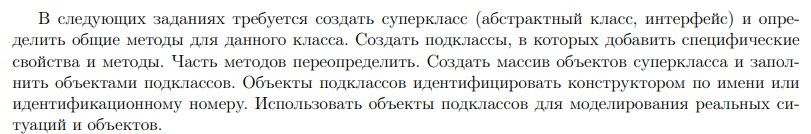
Цель: приобрести практические навыки в области объектно-ориентированного проектирования

Задание 1

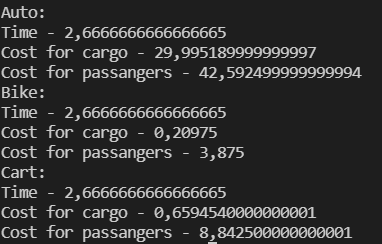




Задание 2







Задание 3





Текст программы:

Program.cs

using System;

using System.Collections.Generic;

using static System.Console;

namespace lab05

{

    class Program

    {

        static void Task1() {

            Encyclopedia encyclopedia = new Encyclopedia() {

                Code = 123,

                Author = "Maksim",

                Name = "Sanya solnyshko",

                Year = 2021,

                Publisher = "Minsk",

                ForWhome = "For children"

            };

            Guide guide = new Guide() {

                Code = 124,

                Author = "Bomj",

                Name = "Smisl ponyaten???",

                Year = 2021,

                Publisher = "Brest",

                ForWhat = "For something"

            };

            encyclopedia.ShowInfo();

            Console.WriteLine();

            guide.ShowInfo();

        }

        static void Task2() {

            List<Vehicle> vehicles = new List<Vehicle>();

            Auto auto = new Auto() {

                Rentcost = 12.1,

                Rashod = 14

            };

            WriteLine("Auto:");

            WriteLine($"Time - {auto.GetTime(75, 200)}");

            WriteLine($"Cost for cargo - {auto.GetCostCargo(16.78, 200)}");

            WriteLine($"Cost for passangers - {auto.GetCostPass(3, 200)}");

            Bike bike = new Bike() {

                Rentcost = 2.5,

                Wheel = 2

            };

            WriteLine("Bike:");

            WriteLine($"Time - {bike.GetTime(75, 200)}");

            WriteLine($"Cost for cargo - {bike.GetCostCargo(16.78, 200)}");

            WriteLine($"Cost for passangers - {bike.GetCostPass(3, 200)}");

            Cart cart = new Cart() {

                Rentcost = 7.86,

                MaxWeight = 20

            };

            WriteLine("Cart:");

            WriteLine($"Time - {cart.GetTime(75, 200)}");

            WriteLine($"Cost for cargo - {cart.GetCostCargo(16.78, 200)}");

            WriteLine($"Cost for passangers - {cart.GetCostPass(3, 200)}");

            vehicles.Add(auto);

            vehicles.Add(bike);

            vehicles.Add(cart);

        }

        static void Main(string[] args)

        {

            //Task1();

            Task2();

        }

    }

}

Books.cs

using System;

namespace lab05 {

    abstract class Book {

        public int Code { get; set; }

        public string Author { get; set; }

        public string Name { get; set; }

        public int Year { get; set; }

        public string Publisher { get; set; }

        public virtual void ShowInfo() {

            Console.WriteLine(Code);

            Console.WriteLine(Author);

            Console.WriteLine(Name);

            Console.WriteLine(Year);

            Console.WriteLine(Publisher);

        }

    }

    class Encyclopedia : Book {

        public string ForWhome { get; set; }

        public override void ShowInfo()

        {

            base.ShowInfo();

            Console.WriteLine(ForWhome);

        }

    }

    class Guide : Book {

        public string ForWhat { get; set; }

        public override void ShowInfo()

        {

            base.ShowInfo();

            Console.WriteLine(ForWhat);

        }

    }

}

Ts.cs

namespace lab05 {

    abstract class Vehicle {

        public double Rentcost { get; set; }

        public virtual double GetTime(double speed, double distance) => distance / speed;

        public virtual double GetCostPass(double count, double distance) => count \* 75.0 / distance \* Rentcost;

        public virtual double GetCostCargo(double weight, double distance) => weight / distance \* Rentcost;

    }

    class Auto : Vehicle {

        public double Rashod { get; set; }

        public override double GetCostPass(double count, double distance)

        {

            if (count > 4) count = 4;

            return base.GetCostPass(count, distance)  + Rashod \* 2.07;

        }

        public override double GetCostCargo(double weight, double distance)

        {

            return base.GetCostCargo(weight, distance) + Rashod \* 2.07;

        }

    }

    class Bike : Vehicle {

        public int Wheel { get; set; }

        public override double GetCostPass(double count, double distance)

        {

            if (count > 2) count = 2;

            return base.GetCostPass(count, distance) + Wheel;

        }

    }

    class Cart : Vehicle {

        public double MaxWeight { get; set; }

        public override double GetCostCargo(double weight, double distance)

        {

            if (weight > MaxWeight) weight = MaxWeight;

            return base.GetCostCargo(weight, distance);

        }

    }

}

Cource.cs

using System;

using System.Collections.Generic;

namespace lab04 {

    abstract class Person {

        protected string \_name;

        public virtual void ShowName() {

            Console.WriteLine($"Name: {\_name}");

        }

        public override string ToString()

        {

            return \_name;

        }

    }

    class Teacher : Person {

        public Dictionary<Course, List<Student>> \_lst = new Dictionary<Course, List<Student>>();

        public Teacher(string name) {

            \_name = name;

        }

        public void AddCourse(string nameofcourse) {

            Course course = new Course(nameofcourse);

            \_lst.Add(course, new List<Student>());

        }

        public override string ToString()

        {

            return \_name.ToString();

        }

        public Course GetCourse(string name) {

            foreach (var sd in \_lst.Keys) {

                if (sd.ToString() == name){

                    return sd;

                }

            }

            return null;

        }

        public Mark SetMark(int mark, Student student, string course) {

            var n = GetCourse(course);

            var v = \_lst[n];

            v.Remove(student);

            \_lst[n] = v;

            Mark mar = new Mark(mark, this, student, n);

            Archive.AddMark(mar);

            return mar;

        }

    }

    class Course {

        private string \_nameofcourse;

        public Course (string coursename) {

            \_nameofcourse = coursename;

        }

        public override string ToString()

        {

            return \_nameofcourse;

        }

    }

    class Student : Person {

        private List<Course> \_courses = new List<Course>();

        public Student(string name) {

            \_name = name;

        }

        public void EnterToCourse(Teacher teacher, string coursename) {

            var crs = teacher.GetCourse(coursename);

            var vrs = teacher.\_lst[crs];

            vrs.Add(this);

            teacher.\_lst[crs] = vrs;

        }

    }

    static class Archive {

        private static List<Mark> lst = new List<Mark>();

        public static void AddMark(Mark mark) {

            lst.Add(mark);

        }

        public static void ShowMarks() {

            foreach (Mark mark in lst) {

                Console.WriteLine(mark.ToString());

            }

        }

    }

    class Mark {

        public int mark;

        public Teacher teacher;

        public Student student;

        public Course course;

        public Mark(int mark1, Teacher teacher1, Student student1, Course course1) {

            mark = mark1;

            teacher = teacher1;

            student = student1;

            course = course1;

        }

        public override string ToString()

        {

            return $"Student name: {student.ToString()}, Teacher name: {teacher.ToString()}, Course name: {course.ToString()}, Mark: {mark}";

        }

    }

}

Вывод: приобрёл практические навыки в области объектно-ориентированного проектирования.