

**Задание.**

1) Для придуманной вами задачи (Пункт 1 из задания к л/р 2 + пункт 1 из задания к л/р 4) реализовать необходимые классы на языке C#;

2) Продемонстрировать их использование в консольном С# приложении;

3) Продемонстрировать сборку проекта и запуск приложения на платформе Mono под Linux, Mac OS и пр.;

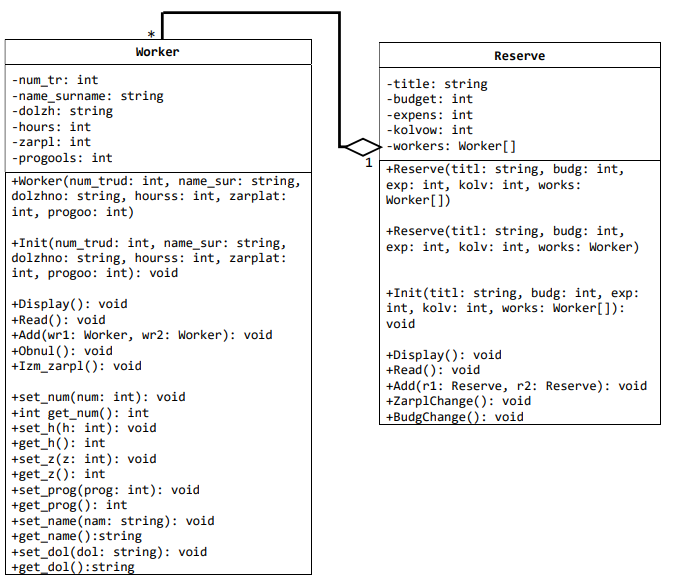
!) Использование GIT обязательно.

**Прикладная задача.**

Прикладной задачей данной работы является управление заповедниками.

Нужно вести учёт работников заповедника для хорошего управления, а также самих заповедников для эффективного распределения средств. Использовать классы.

**Диаграмма классов**



**Класс:** Worker (работник заповедника) (будет вызываться другим классом)

***Поля:***

private int num\_tr номер трудовой книжки

private string name\_surname имя и фамилия

private string dolzh должность

private int hours кол-во рабочих часов

private int zarpl зарплата в месяц в рублях

private int progools кол-во прогулов (в днях)

***Методы:***

public Worker(int num\_trud, string name\_sur, string dolzhno, int hourss, int zarplat, int progoo)

конструктор с параметрами

public void Init(int num\_trud, string name\_sur, string dolzhno, int hourss, int zarplat, int progoo)

инициализация

public void Display() вывод

public void Read() ввод

public void Add(Worker wr1, Worker wr2) сложение

public void Obnul() обнуление прогулов (прикладное)

public void Izm\_zarpl() изменение зарплаты (прикладное)

Получение и установление соответствующих полей

public void set\_num(int num)

public int get\_num()

public void set\_h(int h)

public int get\_h()

public void set\_z(int z)

public int get\_z()

public void set\_prog(int prog)

public int get\_prog()

public void set\_name(string nam)

public string get\_name()

public void set\_dol(string dol)

public string get\_dol()

**Класс:** Reserve (заповедник) (будет вызывать класс Worker)

***Поля:***

private string title название заповедника

private int budget бюджет заповедника

private int expens расходы

private int kolvow кол-во работников в заповеднике

private Worker workers[] работники заповедника

***Методы:***

public Reserve(string titl, int budg, int exp, int kolv, Worker[] works)

конструктор с параметрами

public void Init(string titl, int budg, int exp, int kolv, Worker[] works) Инициализация

public Reserve(string titl, int budg, int exp, int kolv, Worker works)

Конструктор с параметрами (выполняет инициализацию, вторая перегрузка)

public void Display() вывод

public void Read() ввод

public void Add(Reserve r1, Reserve r2) сложение

public void ZarplChange() изменение зарплаты всех работников (прикладное)

public void BudgChange() изменение бюджета (прикладное)

*Ссылка на работу в GitHub:* <https://github.com/GulinAlexey/lab6prog_sharp/tree/master>

**1) Код программы для задания**

**Program.cs**

//Гулин А. Н., ПИ-92. Лаб. 6 "Первая программа на С#"

using System;

namespace lab6prog\_sharp

{

class Worker //работник заповедника

{

private int num\_tr; //номер трудовой книжки

private string name\_surname; //имя и фамилия

private string dolzh; //должность

private int hours; //кол-во рабочих часов

private int zarpl; //зарплата в месяц в рублях

private int progools; //кол-во прогулов (в днях)

//конструктор с параметрами

public Worker(int num\_trud, string name\_sur, string dolzhno, int hourss, int zarplat, int progoo)

{

this.num\_tr = num\_trud;

this.name\_surname = name\_sur;

this.dolzh = dolzhno;

this.hours = hourss;

this.zarpl = zarplat;

this.progools = progoo;

}

//инициализация

public void Init(int num\_trud, string name\_sur, string dolzhno, int hourss, int zarplat, int progoo)

{

this.num\_tr = num\_trud;

this.name\_surname = name\_sur;

this.dolzh = dolzhno;

this.hours = hourss;

this.zarpl = zarplat;

this.progools = progoo;

}

//Получение и установление соответствующих полей

public void set\_num(int num)

{

this.num\_tr = num;

}

public int get\_num()

{

return num\_tr;

}

public void set\_h(int h)

{

this.hours = h;

}

public int get\_h()

{

return hours;

}

public void set\_z(int z)

{

this.zarpl = z;

}

public int get\_z()

{

return zarpl;

}

public void set\_prog(int prog)

{

this.progools = prog;

}

public int get\_prog()

{

return progools;

}

public void set\_name(string nam)

{

this.name\_surname = nam;

}

public string get\_name()

{

return name\_surname;

}

public void set\_dol(string dol)

{

this.dolzh = dol;

}

public string get\_dol()

{

return dolzh;

}

public void Read() //ввод

{

Console.WriteLine("\nInput info about worker.\n");

Console.WriteLine("Input num of workbook: ");

num\_tr=Int32.Parse(Console.ReadLine());

Console.WriteLine("Input name and surname: ");

name\_surname=Console.ReadLine();

Console.WriteLine("Input dolzhnost: ");

dolzh=Console.ReadLine();

Console.WriteLine("Input work hours: ");

hours=Int32.Parse(Console.ReadLine());

Console.WriteLine("Input zarplata: ");

zarpl=Int32.Parse(Console.ReadLine());

Console.WriteLine("Input progools: ");

progools=Int32.Parse(Console.ReadLine());

}

public void Display() //вывод

{

Console.WriteLine("\nOutput info about worker.\n");

Console.WriteLine("Num of workbook: {0}\n", num\_tr);

Console.WriteLine("Name and surname: {0}\n", name\_surname);

Console.WriteLine("Dolzhnost: {0}\n", dolzh);

Console.WriteLine("Work hours: {0}\n", hours);

Console.WriteLine("Zarplata: {0}\n", zarpl);

Console.WriteLine("Progools: {0}\n", progools);

}

public void Add(Worker wr1, Worker wr2) //сложение

{

Worker wrsum = new Worker(12345, "No Name", "No Prof", 0, 0, 0);

wrsum = wr1;

wrsum.hours += wr2.hours; //прибавить к имеющимся числовым переменным суммарного объекта значения из второго объекта (кроме номера трудовой)

wrsum.zarpl += wr2.zarpl;

wrsum.progools += wr2.progools;

this.Init(wrsum.num\_tr, wrsum.name\_surname, wrsum.dolzh, wrsum.hours, wrsum.zarpl, wrsum.progools); //вернуть итоговый объект как результат

}

//обнуление прогулов (прикладное)

public void Obnul()

{

this.progools = 0; //обнулить прогулы

}

public void Izm\_zarpl() //изменение зарплаты (прикладное)

{

Console.WriteLine("\nChanging zarplata of worker\n");

Console.WriteLine("Input changes of zarplata: ");

int izm; //переменная с прибавкой или убавкой зарплаты

izm = Int32.Parse(Console.ReadLine());

this.zarpl+=izm; //добавить изменение к текущей зарплате

}

}

class Reserve //заповедник

{

private string title; //название заповедника

private int budget; //бюджет заповедника

private int expens; //расходы

private int kolvow; //кол-во работников в заповеднике

private Worker[] workers = new Worker[100]; //работники заповедника

//конструктор с параметрами

public Reserve(string titl, int budg, int exp, int kolv, Worker[] works)

{

this.title = titl;

this.budget = budg;

this.expens = exp;

this.kolvow = kolv;

for (int i = 0; i < kolv; i++)

{

this.workers[i] = works[i];

}

}

//конструктор с параметрами (вторая перегрузка)

public Reserve(string titl, int budg, int exp, int kolv, Worker works)

{

this.title = titl;

this.budget = budg;

this.expens = exp;

this.kolvow = kolv;

for (int i = 0; i < kolv; i++)

{

this.workers[i] = works;

}

}

//инициализация

public void Init(string titl, int budg, int exp, int kolv, Worker[] works)

{

this.title = titl;

this.budget = budg;

this.expens = exp;

this.kolvow = kolv;

for (int i = 0; i < kolv; i++)

{

this.workers[i] = works[i];

}

}

public void Display() //вывод

{

Console.WriteLine("\nOutput info about reserve.\n");

Console.WriteLine("Title: {0}\n", title);

Console.WriteLine("Budget: {0}\n", budget);

Console.WriteLine("Expenses: {0}\n", expens);

Console.WriteLine("Count of workers: {0}\n", kolvow);

int n = this.kolvow; //получить кол-во работников

for(int i=0; i<n; i++)

{

Console.WriteLine("\nWorker {0}\n", i + 1);

Console.WriteLine("Num of workbook: {0}\n", workers[i].get\_num());

Console.WriteLine("Name and surname: {0}\n", workers[i].get\_name());

Console.WriteLine("Dolzhnost: {0}\n", workers[i].get\_dol());

Console.WriteLine("Work hours: {0}\n", workers[i].get\_h());

Console.WriteLine("Zarplata: {0}\n", workers[i].get\_z());

Console.WriteLine("Progools: {0}\n", workers[i].get\_prog());

}

}

public void Read() //ввод

{

Console.WriteLine("\nInput info about reserve.\n");

Console.WriteLine("Input title: ");

this.title=Console.ReadLine();

Console.WriteLine("Input budget: ");

this.budget=Int32.Parse(Console.ReadLine());

Console.WriteLine("Input expenses: ");

this.expens=Int32.Parse(Console.ReadLine());

Console.WriteLine("Input count of workers: ");

this.kolvow = Int32.Parse(Console.ReadLine());

int n\_k = this.kolvow;

for(int i=0; i<n\_k; i++)

this.workers[i].Read();

}

public void Add(Reserve r1, Reserve r2) //сложение

{

Reserve rsum;

rsum = r1; //переписать первую структуру в суммарную структуру

rsum.budget += r2.budget; //прибавить к имеющимся числовым переменным суммарной структуры значения из второй структуры

rsum.expens += r2.expens;

rsum.kolvow += r2.kolvow;

this.Init(rsum.title, rsum.budget, rsum.expens, rsum.kolvow, rsum.workers); //вернуть итоговый объект как результат

}

public void ZarplChange() //изменение зарплаты всех (прикладное)

{

Console.WriteLine("\nChanging zarplata of all workers\n");

Console.WriteLine("Input changes of zarplata: ");

int izm; //переменная с прибавкой или убавкой

izm = Int32.Parse(Console.ReadLine());

int n = this.kolvow; //получить кол-во работников

for(int i=0; i<n; i++)

{

this.workers[i].set\_z(workers[i].get\_z()+izm); //добавить изменение к текущему

}

}

public void BudgChange() //изменение бюджета (прикладное)

{

Console.WriteLine("\nChanging budget of reserve\n");

Console.WriteLine("Input changes of budget: ");

int izm; //переменная с прибавкой или убавкой

izm = Int32.Parse(Console.ReadLine());

this.budget+=izm; //добавить изменение к текущему

}

}

class Program

{

static void Main(string[] args) //основная программа (главная)

{

const int LEN = 100;

Console.WriteLine("Start program for working with workers and reserves.\n");

Worker[] w0 = new Worker[LEN]; //начальные данные работника для инициализации заповедника

for (int i = 0; i < LEN; i++)

w0[i] = new Worker(12345, "No Name", "No Prof", 0, 0, 0);

Reserve res1= new Reserve("No Name", 0, 0, 100, w0);

res1.Read();

res1.Display();

res1.ZarplChange();

res1.Display();

res1.BudgChange();

res1.Display();

Console.WriteLine("\nEnd of program. Press any key to exit...");

Console.ReadKey(); //ожидание нажатия любой клавиши для выхода.

}

}

}

**2) Тест программы для задания**

