

**МІНІСТЕРСТВО
ОСВІТИ ТА НАУКИ УКРАЇНИ
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Кафедра інформаційних систем та технологій**

**Звіт
з лабораторної роботи № 4
«Інтерфейси. Абстрактні класи»
з дисципліни
«Програмування – 2. Основи програмування»**

Варіант №17

Перевірив: _____

Виконав:

Коношевський Денис

Студент гр. ІС-11 ФІОТ

Посада, ПБ викладача

1 курс,

залікова книжка № ІС-1117

Лабораторна робота #4

Інтерфейси. Абстрактні класи

Завдання

17. Створити суперклас Співробітник підкласи Керівник, Менеджер, Інженер, Механік, Інженер-архітектор. За допомогою конструктора задати кількість досвіду співробітника в роках. Вивести зарплату кожного співробітника та кількість вихідних в році у інженерів. Реалізувати функцію підрахунку кількості підлеглих у менеджера

Р.с.про функцію підрахунку підлеглих:

По-ідеї, найвищий – керівник, потім – менеджер, а далі просто робітники, тому я рахував підлеглими менеджера інженерів, механіків та інженерів-архітекторів. Я робив це за допомогою зарплатні, оскільки найвищу зарплату повинен мати керівник, потім – менеджер, а далі просто робітники.

```
Employee 1 has experience 15 year, salary 50000 uah per month and 156 holidays per year
Employee 2 has experience 12 year, salary 35000 uah per month and 104 holidays per year
Employee 3 has experience 8 year, salary 20000 uah per month and 52 holidays per year
Employee 4 has experience 6 year, salary 15000 uah per month and 52 holidays per year
Employee 5 has experience 5 year, salary 12500 uah per month and 52 holidays per year

Manager has 3 subordinates
```

```
using Laba_4;
using System;
using System.Collections.Generic;

namespace Lab4
{
    class Program
    {
        static void Main(string[] args)
        {
            Head head_ex = new Head(15, 50000, 156);
            Manager manager_ex = new Manager(12, 35000, 104);
            Engineer engineer_ex = new Engineer(8, 20000, 52);
            Mechanic mechanic_ex = new Mechanic(6, 15000, 52);
            Engineer_Architect engineer_architect_ex = new Engineer_Architect(5,
12500, 52);
            ListOfEmployees list = new ListOfEmployees();
            list.Add(head_ex);
            list.Add(manager_ex);
            list.Add(engineer_ex);
            list.Add(mechanic_ex);
            list.Add(engineer_architect_ex);
            for (int i = 0; i < list.GetCount(); i++)
            {
                Console.WriteLine($"Employee {i+1} has experience "+
list.GExperience(i) + " " +
                " year, salary "+ list.GSalary(i) + " uah per month and " +
                ""+ list.GHoliday(i) + " holidays per year ");
            }
            Console.WriteLine();
            Console.WriteLine($"Manager has {list.GetSubordinates()} subordinates");
        }
    }
}

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

namespace Laba_4
```

```

{
    class ListOfEmployees
    {
        private List<Employee> MainList = new List<Employee>();
        public void Add(Employee item)
        {
            MainList.Add(item);
        }
        public int GExperience(int index)
        {
            return MainList[index].GetExperience();
        }
        public int GSalary(int index)
        {
            return MainList[index].GetSalary();
        }
        public int GHoliday(int index)
        {
            return MainList[index].GetHoliday();
        }
        public int GetCount()
        {
            return MainList.Count;
        }
        public int GetSubordinates()
        {
            int sub = 0;
            for(int i = 0; i < MainList.Count; ++i)
            {
                if (MainList[i].GetSalary() < 35000) ++sub;
            }
            return sub;
        }
    }
}

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

namespace Laba_4
{
    abstract class Employee
    {
        public abstract int Experience { get; set; }
        public abstract int Salary { get; set; }
        public abstract int Holiday { get; set; }
        public Employee(int experience, int salary, int holiday)
        {
            Experience = experience;
            Salary = salary;
            Holiday = holiday;
        }
        public abstract int GetExperience();
        public abstract int GetSalary();
        public abstract int GetHoliday();
    }
}

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

namespace Laba_4
{
    class Head : Employee

```

```

    {
        private int experience;
        private int salary;
        private int holiday;
        public Head(int experience, int salary, int holiday) : base(experience,
salary, holiday) { }
        public override int Experience
        {
            get => experience;
            set => experience = value;
        }
        public override int Salary
        {
            get => salary;
            set => salary = value;
        }
        public override int Holiday
        {
            get => holiday;
            set => holiday = value;
        }
        public override int GetExperience()
        {
            return Experience;
        }
        public override int GetSalary()
        {
            return Salary;
        }
        public override int GetHoliday()
        {
            return Holiday;
        }
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Laba_4
{
    class Manager : Employee
    {
        private int experience;
        private int salary;
        private int holiday;
        public Manager(int experience, int salary, int holiday) : base(experience,
salary, holiday) { }
        public override int Experience
        {
            get => experience;
            set => experience = value;
        }
        public override int Salary
        {
            get => salary;
            set => salary = value;
        }
        public override int Holiday
        {
            get => holiday;
            set => holiday = value;
        }
        public override int GetExperience()
        {
            return Experience;
        }
    }
}

```

```

    }
    public override int GetSalary()
    {
        return Salary;
    }
    public override int GetHoliday()
    {
        return Holiday;
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Laba_4
{
    class Engineer : Employee
    {
        private int experience;
        private int salary;
        private int holiday;
        public Engineer(int experience, int salary, int holiday) : base(experience,
salary, holiday) { }
        public override int Experience
        {
            get => experience;
            set => experience = value;
        }
        public override int Salary
        {
            get => salary;
            set => salary = value;
        }
        public override int Holiday
        {
            get => holiday;
            set => holiday = value;
        }
        public override int GetExperience()
        {
            return Experience;
        }
        public override int GetSalary()
        {
            return Salary;
        }
        public override int GetHoliday()
        {
            return Holiday;
        }
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Laba_4
{
    class Mechanic : Employee
    {
        private int experience;
        private int salary;
        private int holiday;

```

```

        public Mechanic(int experience, int salary, int holiday) : base(experience,
salary, holiday) { }
        public override int Experience
        {
            get => experience;
            set => experience = value;
        }
        public override int Salary
        {
            get => salary;
            set => salary = value;
        }
        public override int Holiday
        {
            get => holiday;
            set => holiday = value;
        }
        public override int GetExperience()
        {
            return Experience;
        }
        public override int GetSalary()
        {
            return Salary;
        }
        public override int GetHoliday()
        {
            return Holiday;
        }
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Laba_4
{
    class Engineer_Architect : Employee
    {
        private int experience;
        private int salary;
        private int holiday;
        public Engineer_Architect(int experience, int salary, int holiday) :
base(experience, salary, holiday) { }
        public override int Experience
        {
            get => experience;
            set => experience = value;
        }
        public override int Salary
        {
            get => salary;
            set => salary = value;
        }
        public override int Holiday
        {
            get => holiday;
            set => holiday = value;
        }
        public override int GetExperience()
        {
            return Experience;
        }
        public override int GetSalary()
        {
            return Salary;
        }
    }
}

```

```
    }  
    public override int GetHoliday()  
    {  
        return Holiday;  
    }  
}  
}
```