

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Казанский национальный исследовательский технический университет
им. А.Н. Туполева – КАИ»

Институт компьютерных технологий и защиты информации
Отделение СПО ИКТЗИ (Колледж информационных технологий)

ПРАКТИЧЕСКАЯ РАБОТА №1

по дисциплине

Основы алгоритмизации и программирования

Тема: «ООП. Наследование»

Работу выполнил

Студент гр.4237

Данилова Анастасия Александровна

Принял

Преподаватель Шмидт И.Р.

Казань 2024

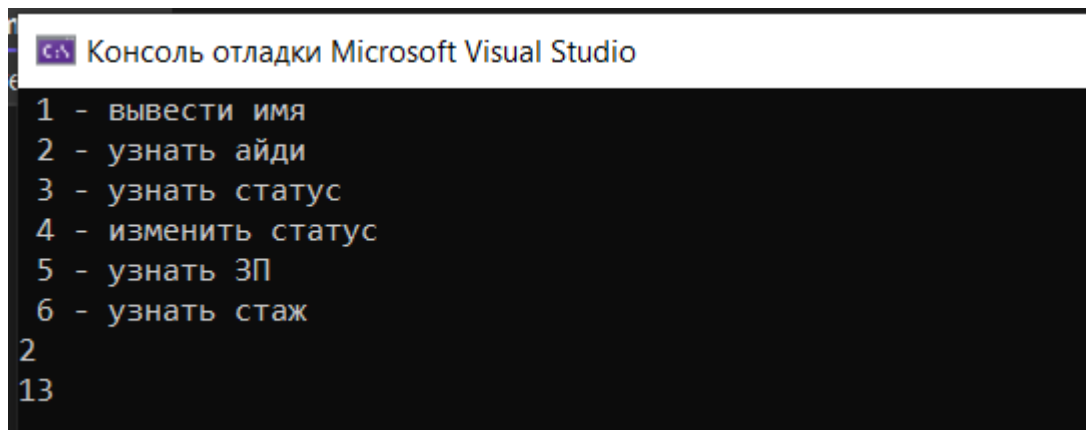
Цель работы

Изучить наследование (ООП).

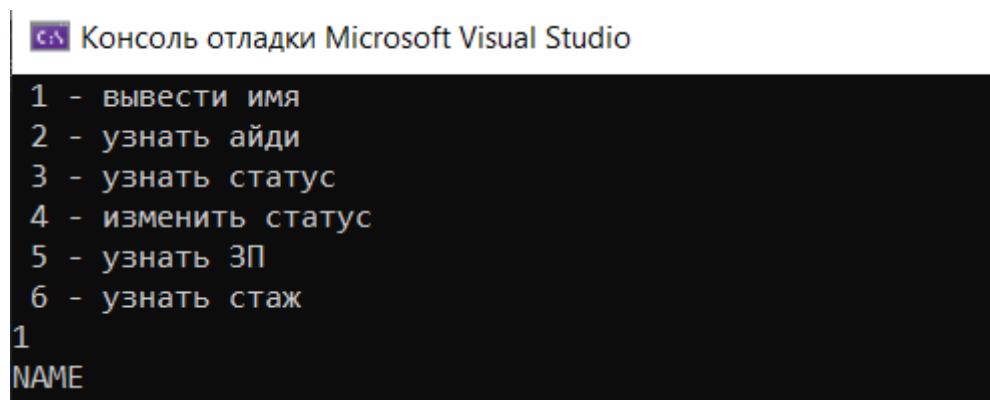
Задание на лабораторную работу

Создать 2 класса (родительский и дочерний). В родительском классе 4 поля и 2 метода. В дочернем классе создать ещё 2 поля и переопределить 2 метода и добавить 2 новых.

Результат выполнения работы



```
Консоль отладки Microsoft Visual Studio
1 - вывести имя
2 - узнать айди
3 - узнать статус
4 - изменить статус
5 - узнать ЗП
6 - узнать стаж
2
13
```



```
Консоль отладки Microsoft Visual Studio
1 - вывести имя
2 - узнать айди
3 - узнать статус
4 - изменить статус
5 - узнать ЗП
6 - узнать стаж
1
NAME
```

Листинг кода

```
using System.Data;

namespace inheretOOP
{
```

```

internal class Program
{
    static void Main(string[] args)
    {
        Employee employee = new Employee(); // создаём новый объект класса,
        отдельный-конкретный рабочий
        start:
        Console.Clear();
        Console.WriteLine(" 1 - вывести имя \n 2 - узнать айди \n 3 - узнать статус
        \n 4 - изменить статус \n 5 - узнать ЗП \n 6 - узнать стаж");
        int num = Convert.ToInt32(Console.ReadLine());
        switch (num)
        {
            case 1:
                employee.name_output("NAME");
                break;

            case 2:
                employee.show_id();
                break;

            case 3:

                break;

            case 4:
                break;

            case 5:
                break;

            case 6:
                break;

            default:
                Console.WriteLine("invalid number");
                goto start;
        }
    }
}

class Worker // базовый класс
{
    public string name = " ";
    protected int salary = 100;
    protected bool status = true;
    protected int experience = 0;

    protected virtual void work()
    {
        Console.WriteLine("they're working :)");
    }

    protected virtual void experience_gain()
    {
        Console.WriteLine("guess the number: ");
        int num = Convert.ToInt32(Console.ReadLine());
        if (num > 0)
        {
            experience++;
        }
    }
}

```

```

        public bool status_change()
        {
            status = false;
            Console.WriteLine($"{name} got laid off :(");
            return status;
        }
    }

class Employee : Worker // дочерний класс для конкретных рабочих
{
    int id;

    protected override void experience_gain() // перезапись 1 метода
    {
        if (experience == 1 )
        {
            Console.WriteLine("error");
        }
    }

    protected override void work() // перезапись 2 метода
    {
        if (status = true)
        {
            Console.WriteLine("correct. they're still working");
        }
        else
        {
            Console.WriteLine("incorrect");
        }
    }

    public void name_output(string name) // ввод имени
    {
        Console.WriteLine(name);
    }

    int id_generator()
    {
        Random rnd = new Random();
        id = rnd.Next(0, 20);
        return id;
    }

    public void show_id()
    {
        id_generator();
        Console.WriteLine(id);
    }
}
}

```