**Практическая работа**

**Классы. Наследование.**

1. ***Цель работы.*** Получение навыков составления программ с классами среде  Visual Studio.Net

***2. Литература.***  Т.А.Павловская “С#. Программирование на языке высокого уровня”, И.Г.Семакин, А.П.Шестаков «Основы программирования», Т.А.Павловская, Ю.А.Щупак «С/С++. Структурное программирование. Практикум», Е.Л.Шиманович «С/С++ в примерах и задачах»

***3. Подготовка к работе.***  Изучить конспект лекций по темам «Классы. Данные класса. Методы класса. Операции класса. Свойства.Static. Наследование».

***4. Перечень оборудования.*** Интегрированная среда Visual Studio.Net.

***5. Задание.*** Научиться составлять программы с классами в интегрированной среде Visual Studio.Net.

**Задание: Разработка системы классов для сотрудников компании**    
  
**Описание:**    
Реализуйте систему классов для учета сотрудников компании, используя принципы наследования в C#.  
  
**Требования:**    
  
1.**Создайте базовый класс Employee** с полями:   
   - string Name (имя сотрудника)   
   - int Age (возраст)   
   - double Salary (базовая зарплата)   
   - Метод virtual double CalculateBonus() для расчета премии (по умолчанию 5% от зарплаты).

   - Метод virtual void ShowInfo() для вывода информации о сотруднике.  
  
2.**Создайте производные классы:**    
   - Manager, добавьте поле int TeamSize (размер команды).   
     - Бонус = 10% от зарплаты + TeamSize \* 500.    
   - Developer, добавьте поле string ProgrammingLanguage (язык программирования).   
     - Бонус = 15% от зарплаты.   
   - Intern, добавьте поле string University (университет стажера).   
     - Бонус = 3% от зарплаты.   
  
3.**Переопределите ShowInfo() в каждом классе,** чтобы он выводил дополнительную информацию.

public class People

{

public string FullName { get; set; }

public string Gender { get; set; }

public People(string fullName, string gender)

{

this.FullName = fullName;

this.Gender = gender;

}

public virtual void GetInformation()

{

Console.WriteLine($"Привет! Я {FullName}! Мой пол {Gender}");

}

}

public class Student : People

{

public double Point { get; set; }

public Student(string fullName, string gender, double point): base(fullName, gender)

{

this.Point = point;

}

public override void GetInformation()

{

base.GetInformation();

Console.WriteLine($"Мой средний балл {Point}");

}

public class GraduateStudent : Student

{

public bool IsMagistr { get; set; }

public GraduateStudent(string fullName, string gender, double point, bool isMagistr) : base(fullName, gender, point)

{

this.IsMagistr = isMagistr;

}

public override void GetInformation()

{

base.GetInformation();

Console.WriteLine($"Я магистр: {(IsMagistr ? "да":"нет")}");

}

}

}

internal class Program

{

static void Main(string[] args)

{

People people = new People("Дунаева Оксана Николаевна", "Женский");

people.GetInformation();

Student student = new Student("Кумышбаева София Алексевна", "Женский", 5);

student.GetInformation();

GraduateStudent graduateStudent = new GraduateStudent("Сумбаев Иван Васильевич", "Мужской", 4, true);

graduateStudent.GetInformation(); }}