

**АКАДЕМИЯ МАРКЕТИНГА
И СОЦИАЛЬНО-ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ – ИМСИТ**

г. Краснодар

**Факультет информатики и вычислительной техники
Кафедра математики и вычислительной техники**

УТВЕРЖДАЮ

Протокол № 1 от 29.08.2016 г.

Зав. кафедрой _____ Н.С.Нестерова

**Б3.В.ОД.7 ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ
ПРОГРАММИРОВАНИЕ**

Лабораторный практикум по дисциплине
для студентов направления подготовки
09.03.01 «Информатика и вычислительная техника»
Направленность (профиль) образовательной программы
«Автоматизированные системы обработки информации и управления»
Квалификация (степень выпускника) бакалавр

Краснодар, 2016

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

В соответствии с учебным планом студенты заочного отделения выполняют задания, состоящие из теоретических вопросов и решения задач. Задания построены таким образом, чтобы выяснить подготовку студента по различным разделам изучаемой дисциплины.

Целью настоящих методических рекомендаций является оказание практической помощи студенту в выполнении заданий по дисциплине. Выполнить работу следует после ознакомления со всеми темами дисциплины.

Темы, вынесенные в задание, требуют от студента углубленного изучения. Ответ на каждый вопрос должен излагаться подробно, с разъяснениями, со ссылкой на нормативно-правовые акты.

К каждому варианту прилагается список литературы, по своей инициативе студент может использовать другие дополнительные источники.

Контрольная работа выполняется письменно в тетради с условием ясного, читаемого почерка или машинописным способом до полного освещения вопросов. Вопросы задания и текст задачи должен быть переписан студентом с указанием варианта. Обязательной является ссылка на соответствующие статьи нормативных актов.

Студент выполняет одно задание по каждой из трёх тем:

Тема 1: Классы и объекты

Тема 2: Перегрузка

Тема 3: Наследование

Требования к оформлению контрольных работ следующие: на титульном листе необходимо написать свою фамилию, имя и отчество, шифр, название дисциплины и фамилию преподавателя.

Для всех контрольных заданий необходимо привести условие задачи, программу на языке программирования высокого уровня (если предусмотрено заданием), пояснения к основным конструкциям и операторам программ.

Первый лист работы – титульный. В нем указывается наименование образовательного учреждения, дисциплина, вариант, фамилия, имя и отчество студента, факультет, курс, группа. Далее следует содержательная часть работы, включающая перечисленные выше разделы.

В конце работы ставится дата и подпись студента. Самостоятельная работа по каждой теме отсылается на кафедру до начала экзаменационной сессии в оговорённые преподавателем сроки и оценивается преподавателем в баллах в соответствии с формой текущего контроля в межсессионный период. Работы не отправленные в срок теряют часть начисляемых баллов.

Ниже представлены методические указания и индивидуальные задания по каждой теме.

1 Классы и объекты

Ознакомиться с материалом по данной теме с помощью источников [1, с.100] или [2, с.178].

Пример выполнения задания

Создать класс для работы с точкой на плоскости. Класс должен состоять из двух полей – координат точки и одного метода, возвращающего расстояние между точками.

Реализуем класс в программе на языке C#, которая вводит координаты с консоли, вычисляет и выводит на консоль результат:

```
using System;
//класс точки на плоскости:
class Point2D
{
    //поля:
    double x, y;
    //конструктор:
    public Point2D(double x, double y)
    {
```

```

        this.x = x;
        this.y = y;
    }
    //метод:
    public double Metric(Point2D p)
    {
        return Math.Sqrt((x-p.x)*(x-p.x) + (y-p.y)*(y-p.y));
    }
}

//класс с точкой вызова:
class Program
{
    public static void Main()
    {
        double x, y;
        //ввод координат точек:
        Console.Write("x0 = ");
        x0 = Convert.ToDouble(Console.ReadLine());
        Console.Write("y0 = ");
        y0 = Convert.ToDouble(Console.ReadLine());
        Console.Write("x1 = ");
        x1 = Convert.ToDouble(Console.ReadLine());
        Console.Write("y1 = ");
        y1 = Convert.ToDouble(Console.ReadLine());
        //объявление экземпляра класса:
        Point2D p0 = new Point2D(x0, y0);
        Point2D p1 = new Point2D(x1, y1);
        //вычисление и вывод результата:
        Console.WriteLine("Расстояние равно: " + p1.Metric(p2));
    }
}

```

}
}

Задачи для самостоятельного решения

Задание 1.

Создать класс Круг. Его поля должны включать радиус и координаты центра круга. Реализовать операции: инициализация полей данных, изменения радиуса круга, вычисления длины окружности и площади круга.

Задание 2.

Создать класс Шар. Его поля должны включать радиус и координаты центра шара. Реализовать операции: инициализация полей данных, изменения радиуса шара и вычисления объёма шара.

Задание 3.

Создать класс Прямоугольник. Его поля должны включать стороны прямоугольника. Реализовать операции: инициализация полей данных с проверкой на их соответствие прямоугольнику, вычисления периметра прямоугольника и его площади.

Задание 4.

Создать класс Треугольник. Его поля должны включать углы и стороны треугольника. Реализовать операции: инициализация полей данных с проверкой на их соответствие треугольнику, вычисления периметра треугольника и его площади.

Задание 5.

Создать класс Угол для работы с углами на плоскости, задаваемыми величиной в градусах. Реализовать операции перевода угла в радианы, приведение к диапазону 0–360, увеличение/уменьшение угла на заданную величину.

Задание 6.

Рациональная (несократимая) дробь представляется парой целых чисел: (a, b) . Создать класс для работы с рациональными дробями. Реализовать операции сложения, умножения и деления рациональных дробей.

Задание 7.

Реализовать класс Товар. В классе должны быть представлены поля: наименование товара (тип string), цена товара, количество единиц товара и номер накладной. Реализовать методы изменения цены товара, его количества, вычисление стоимости товара.

Задание 8.

Создать класс Зарплата. В классе должны быть представлены поля: оклад, процент надбавки, подоходный налог. Реализовать методы вычисления начисленной суммы, вычисления удержанной суммы, вычисления суммы, выдаваемой на руки.

Задание 9.

Создать класс Money для работы с денежными суммами. Денежная сумма должна быть представлена двумя полями целого типа: одно – для рублей, другое – для копеек. Реализовать в классе Money операции сложения и вычитания денежных сумм, а также операцию вывода на консоль денежной суммы таким образом, чтобы копейки отделялись от рубля запятой.

Задание 10.

Комплексное число задаётся парой действительных чисел (a, b) . Разработать класс комплексных чисел. Реализовать выполнение операций сложения: $(a_1, b_1) + (a_2, b_2) = (a_1 + a_2, b_1 + b_2)$ и вычитания: $(a_1, b_1) - (a_2, b_2) = (a_1 - a_2, b_1 - b_2)$ комплексных чисел. Написать программу, демонстрирующую работу с этим классом.

Задание 11.

Разработать класс объектов-векторов, задаваемых координатами концов в трёхмерном пространстве. Реализовать операции сложения и вычитания векторов с получением нового вектора, вычисления скалярного произведения двух векторов и длины вектора. Написать программу, демонстрирующую работу с этим классом.

Задание 12.

Разработать класс объектов-прямых линий, заданных с помощью уравнения $y = kx + b$, $k, b = \text{const}$. Класс должен содержать конструктор, создающий объекты заданием значений атрибутов k, b и вычисляющий значение вертикальной координаты по заданному значению x . Написать программу, демонстрирующую работу с классом.

Задание 13.

Разработать класс конусов, заданных радиусом их основания и высотой. Класс должен содержать конструктор класса и методы, вычисляющие объём и площадь поверхности фигуры.

Задание 14.

Разработать класс эллипсоидов вращения, заданных своими полуосями. Класс должен содержать конструктор класса и методы, вычисляющие объём и площадь поверхности фигуры.

Задание 15.

Разработать класс эллипсов, заданных своими полуосями. Класс должен содержать конструктор класса и метод, вычисляющий площадь.

Задание 16.

Разработать класс прямоугольных тетраэдров, заданных длинами рёбер основания и высотой. Класс должен содержать конструктор и метод, вычисляющий объём фигуры.

2 Перегрузка

Ознакомиться с материалом по данной теме с помощью источников [1, с.161] или [2, с.189].

Пример выполнения задания

В примере из раздела 1 перегрузить операцию «-», которая вычисляет расстояние между двумя точками. Программа на языке C# будет следующей:

```
using System;
//класс точки на плоскости:
class Point2D
{
    //поля:
    double x, y;
    //конструктор:
    public Point2D(double x, double y)
    {
        this.x = x;
```

```

        this.y = y;
    }
    //перегрузка операции - :
    public static double operator - (Point2D p0, Point2D p1)
    {
        return Math.Sqrt((p0.x-p1.x)*(p0.x-p1.x) + (p0.y-p1.y)*(p0.y-p1.y));
    }
}

```

//класс с точкой вызова:

```

class Program
{
    public static void Main()
    {
        double x, y;
        //ввод координат точек:
        Console.Write("x0 = ");
        x0 = Convert.ToDouble(Console.ReadLine());
        Console.Write("y0 = ");
        y0 = Convert.ToDouble(Console.ReadLine());
        Console.Write("x1 = ");
        x1 = Convert.ToDouble(Console.ReadLine());
        Console.Write("y1 = ");
        y1 = Convert.ToDouble(Console.ReadLine());
        //объявление экземпляра класса:
        Point2D p0 = new Point2D(x0, y0);
        Point2D p1 = new Point2D(x1, y1);
        //вычисление расстояния:
        double d = p0 - p1;
        //вывод результата:
    }
}

```



```

        Console.WriteLine("Расстояние равно: " + d);
    }
}

```

Задачи для самостоятельного решения

В задачах из раздела 2 перегрузить следующие операции:

Задание 1. «*» на случай произведения числа на вектор (одномерный массив): $k(a_1, a_2, \dots, a_n) = (k a_1, k a_2, \dots, k a_n)$;

Задание 2. «-» на случай разности пар чисел (a_1, b_1) и (a_2, b_2) : $(a_1, b_1) - (a_2, b_2) = (a_1 - a_2, b_1 - b_2)$;

Задание 3. «+» на случай суммы пар чисел (a_1, b_1, c_1) и (a_2, b_2, c_2) : $(a_1, b_1, c_1) + (a_2, b_2, c_2) = (a_1 + a_2, b_1 + b_2, c_1 + c_2)$;

Задание 4. «*» на случай произведения пар чисел (a_1, b_1) и (a_2, b_2) : $(a_1, b_1) * (a_2, b_2) = (a_1 a_2 - b_1 b_2, a_1 b_2 + a_2 b_1)$;

Задание 5. «/» на случай частного пар чисел (a_1, b_1) и (a_2, b_2) : $(a_1, b_1) / (a_2, b_2) = ((a_1 a_2 + b_1 b_2) / (a_2^2 + b_2^2), (a_2 b_1 - a_1 b_2) / (a_2^2 + b_2^2))$;

Задание 6. «*» на случай произведения пары чисел (a, b) и числа k : $k(a, b) = (k a, k b)$;

Задание 7. унарную «-» на случай смены знака у пары чисел (a, b) : $-(a, b) = (-a, -b)$;

Задание 8. «*» на случай логического произведения (конъюнкции) двух высказываний A и B;

Задание 9. «+» на случай логического сложения (дизъюнкции) двух высказываний A и B;

Задание 10. «-» на случай отрицания (инверсии) высказывания A;

Задание 11. «^» на случай импликации (исключающего ИЛИ) двух высказываний A и B;

Задание 12. «>» и «<» на случай операции следования для двух высказываний A и B: $A \rightarrow B$ и $B \leftarrow A$, соответственно;

Задание 13. «%» на случай операции эквивалентности двух высказываний А и В: $A \sim B$;

Задание 14. «+» для операции объединения двух символьных множеств (массивов);

Задание 15. «/» для операции деления двух символьных множеств (массивов);

Задание 16. «-» для операции пересечения двух символьных множеств (массивов);

Задание 17. «==» и «!=» для сравнения двух символьных множеств (массивов) по количеству и составу их элементов (символов). Результат первой операции – *истина*, если имеет место совпадение множеств и *ложь* – в противном случае. Результат второй операции – *истина*, если имеет место несовпадение множеств и *ложь* – в противном случае;

Задание 18. «<» и «>» для выяснения принадлежности заданного символа множеству (массиву) символов: *символ* < *множество* и *множество* > *символ*. Если символ является элементом множества, то результат – *истина*. Иначе – *ложь*;

Задание 19. «⊂» и «⊃» для выяснения включения заданного символьного множества (массива) в другое множество (массив) символов: *множество1* ⊂ *множество2* и *множество1* ⊃ *множество2*, соответственно. Результат операции – *истина*, если условие включения выполняется. Иначе – *ложь*;

Задание 20. «^» на случай операции возведения числа *a* в степень *b*: a^b .

3 Наследование

Ознакомьтесь с материалом по данной теме с помощью источников [1, с.172] или [2, с.200].

Пример выполнения задания

Создать класс для точки в пространстве, производный от класса Point2D из предыдущих разделов. Программа, реализующая данный классами будет следующей:

```
using System;
//класс точки на плоскости (базовый):
class Point2D
{
    //поля, которые видны в производных классах:
    protected double x, y;
    //конструктор:
    public Point2D(double x, double y)
    {
        this.x = x;
        this.y = y;
    }
    //перегрузка операции - :
    public static double operator - (Point2D p0, Point2D p1)
    {
        return Math.Sqrt((p0.x-p1.x)*(p0.x-p1.x) + (p0.y-p1.y)*(p0.y-p1.y));
    }
}
//класс точки в пространстве (производный):
class Point3D : Point2D
{
    //поле:
    double z;

    //конструктор производного класса:
    public Point3D(double x, double y, double z) : base (x, y)
    {
```

```

        this.z = z;
    }

    //метод производного класса:
    public double Metric(Point3D p)
    {
        return Math.Sqrt ((z - p.z) * (z - p.z) + (x - p.x) * (x - p.x) +
            (y - p.y) * (y - p.y));
    }
}

```

//класс с точкой вызова:

```

class Program
{
    public static void Main()
    {
        double x, y;
        //ввод координат точек:
        Console.Write("x0 = ");
        x0 = Convert.ToDouble(Console.ReadLine());
        Console.Write("y0 = ");
        y0 = Convert.ToDouble(Console.ReadLine());
        Console.Write("z0 = ");
        z0 = Convert.ToDouble(Console.ReadLine());
        Console.Write("x1 = ");
        x1 = Convert.ToDouble(Console.ReadLine());
        Console.Write("y1 = ");
        y1 = Convert.ToDouble(Console.ReadLine());
    }
}

```

```

Console.WriteLine("z1 = ");
z1 = Convert.ToDouble(Console.ReadLine());

//объявление экземпляра класса:
Point3D p0 = new Point3D(x0, y0, z0);
Point3D p1 = new Point3D(x1, y1, z1);

//вывод результата:
Console.WriteLine("Расстояние равно: " + p0.Metric(p1));
}
}

```

Задачи для самостоятельного решения:

Задание 1.

Создать базовый класс **Car** (машина), характеризуемый торговой маркой (строка), числом цилиндров, мощностью. Определить методы переназначения и изменения мощности. Создать производный класс **Lorry** (грузовик), характеризуемый также грузоподъемностью кузова. Определить функции переназначения марки и изменения грузоподъемности.

Задание 2.

Создать класс **Pair** (пара чисел); определить методы изменения полей и сравнения пар: пара **p1** больше пары **p2**, если (**first.p1 > first.p2**) или (**first.p1 = first.p2**) и (**second.p1 > second.p2**). Определить класс-наследник **Fraction** с полями: целая часть числа и дробная часть числа. Определить полный набор методов сравнения.

Задание 3.

Создать класс **Liquid** (жидкость), имеющий поля названия и плотности. Определить методы переназначения и изменения плотности. Создать производный класс **Alcohol** (спирт), имеющий крепость. Определить методы переназначения и изменения крепости.

Задание 4.

Создать класс `Pair` (пара чисел); определить методы изменения полей и вычисления произведения чисел. Определить производный класс `Rectangle` (прямоугольник) с полями-сторонами. Определить методы вычисления периметра и площади прямоугольника.

Задание 5.

Создать класс `Man` (человек), с полями: имя, возраст, пол и вес. Определить методы переназначения имени, изменения возраста и изменения веса. Создать производный класс `Student`, имеющий поле года обучения. Определить методы переназначения и увеличения года обучения.

Задание 6.

Создать класс `Triad` (тройка чисел); определить методы изменения полей и вычисления суммы чисел. Определить производный класс `Triangle` с полями-сторонами. Определить методы вычисления углов и площади треугольника.

Задание 7.

Создать класс `Triangle` с полями-сторонами. Определить методы изменения сторон, вычисления углов, вычисления периметра. Создать производный класс `Equilateral` (равносторонний), имеющий поле площади. Определить метод вычисления площади.

Задание 8.

Создать класс `Triangle` с полями-сторонами. Определить методы изменения сторон, вычисления углов, вычисления периметра. Создать производный класс `RightAngled` (прямоугольный), имеющий поле площади. Определить метод вычисления площади.

Задание 9.

Создать класс `Pair` (пара чисел); определить методы изменения полей и вычисления произведения чисел. Определить производный класс `RightAngled` с полями-катетами. Определить методы вычисления гипотенузы и площади треугольника.

Задание 10.

Создать класс `Triad` (тройка чисел); определить метод сравнения триад (см. задание 2). Определить производный класс `Date` с полями: год, месяц и день. Определить полный набор методов сравнения дат.

Задание 11.

Создать класс `Triad` (тройка чисел); определить метод сравнения триад (см. задание 2). Определить производный класс `Time` с полями: час, минута и секунда. Определить полный набор методов сравнения моментов времени.

Задание 12.

Реализовать класс-оболочку **Number** для числового типа `float`. Реализовать методы сложения и деления. Создать производный класс **Real**, в котором реализовать метод возведения в произвольную степень, и метод для вычисления логарифма числа.

Задание 13.

Создать класс **Triad** (тройка чисел); определить методы увеличения полей на 1. Определить производный класс **Date** с полями: год, месяц и день. Переопределить методы увеличения полей на 1 и определить метод увеличения даты на n дней.

Задание 14.

Реализовать класс-оболочку **Number** для числового типа `double`. Реализовать методы умножения и вычитания. Создать производный класс **Real**, в котором реализовать метод, вычисляющий корень произвольной степени, и метод для вычисления числа π в данной степени.

Задание 15.

Создать класс **Triad** (тройка чисел); определить методы увеличения полей на 1. Определить класс-наследник **Time** с полями: час, минута, секунда. Переопределить методы увеличения полей на 1 и определить методы увеличения на n секунд и минут.

Задание 16.

Создать базовый класс **Pair** (пара целых чисел) с операциями проверки на равенство и перемножения полей. Реализовать операцию вычитания пар по формуле $(a, b) - (c, d) = (a - b, c - d)$. Создать производный класс **Rational**; определить новые операции сложения $(a, b) + (c, d) = (ad + bc, bd)$ и деления $(a, b) / (c, d) = (ad, bc)$; переопределить операцию вычитания $(a, b) - (c, d) = (ad - bc, bd)$.

Задание 17.

Создать класс **Pair** (пара чисел); определить метод перемножения полей и операцию сложения пар $(a, b) + (c, d) = (a + b, c + d)$. Определить производный класс **Complex** с полями: действительная и мнимая части числа. Определить методы умножения $(a, b) \times (c, d) = (ac - bd, ad + bc)$ и вычитания $(a, b) - (c, d) = (a - b, c - d)$.

Задание 18.

Создать класс **Pair** (пара целых чисел); определить методы изменения полей и операцию сложения пар $(a, b) + (c, d) = (a + b, c + d)$. Определить класс-наследник **Long** с полями: старшая часть числа и младшая часть числа. Переопределить операцию сложения и определить методы умножения и вычитания.

Задание 19.

Создать базовый класс `Triad` (тройка чисел) с операциями сложения с числом, умножения на число, проверки на равенство. Создать производный класс `vector3D`, задаваемый тройкой координат. Должны быть реализованы: операция сложения векторов, скалярное произведение векторов.

Задание 20.

Создать класс `Pair` (пара целых чисел); определить метод умножения на число и операцию сложения пар $(a, b) + (c, d) = (a + b, c + d)$. Определить класс-наследник `Money` с полями: рубли и копейки. Переопределить операцию сложения и определить методы вычитания и деления денежных сумм.

Учебно-методическое и информационное обеспечение дисциплины:

а) основная литература

1. Бужан В.В. Объектно-ориентированное программирование: учебно-методическое пособие для студентов очной и заочной форм обучения направления подготовки 230100.62 – Информатика и вычислительная техника / В.В. Бужан. – Краснодар: ИМСИТ, 2013. – 52 с.

2. Немцова Т.И. Программирование на языке высокого уровня. Программирование на языке Object Pascal: Учебное пособие / Т.И. Немцова; Под ред. Л.Г. Гагариной. - М.: ФОРУМ: ИНФРА-М, 2015. - 496 с

3. Фризен И.Г. Основы алгоритмизации и программирования (среда PascalABC.NET) : учеб. пособие / И.Г. Фризен. — М. : ФОРУМ : ИНФРА-М, 2017. — 392 с.

4. Голицына О.Л. Языки программирования: Учебное пособие / О.Л. Голицына, Т.Л. Партыка, И.И. Попов. - 3-е изд., перераб. и доп. - М.: Форум: ИНФРА-М, 2015. - 400 с

5. Колдаев В.Д. Основы алгоритмизации и программирования: Учебное пособие / В.Д. Колдаев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: ИНФРА-М, 2015. - 416 с.

б) дополнительная литература

6. Голицина О.Л., Попов И.И. Программирование на языках высокого уровня. М.: ФОРУМ, 2011. 496 с.

7. Васильев, А.Н. Java: Объектно-ориентированное программирование: базовый курс для магистров и бакалавров. – СПб: ПИТЕР, 2011. – 400 с.

8. Информатика и программирование. Компьютерный практикум: учеб. пособие для вузов / Гуда, А.Н., Бутаков, М.А., Нечитайло, Н.М., Чернов, А.В. – М.: Дашков и К, 2010. – 240 с.

9. Бужан В.В. Основы программирования в среде Borland C++ Builder. Учебное пособие. — Краснодар.: ЮИМ, 2005. — 109 с.

10. Страуструп Б. Язык программирования Си++: Пер. с англ.— М.: Радио и связь, 2001.— 352 с.

11. Алексеев Е.Р. Программирование на Visual C++ и Turbo C++ Explorer. М.:ИТ Пресс, 2007. – 352 с.

12. Пахомов Б.Н. C/C++ и MS Visual C++ 2005 для начинающих. – СПб.: БХВ-Петербург, 2007. – 464 с.

13. Сергеев А.П., Терен А.Н. Программирование в Microsoft Visual C++ 2005. – М.:Издательский дом «Вильямс», 2006. – 352 с.

14. Хортон А. Visual C++ 2005: Базовый курс – М.: И.Д. «Вильямс», 2007. – 1152 с.

15. Культин Н. С# в задачах и примерах. – СПб.: БХВ-Петербург, 2007. – 240 с.

Доцент, ведущий дисциплину

_____ В.В. Бужан