|  |  |
| --- | --- |
| КГЭУ | МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования** **«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»** (ФГБОУ ВО «КГЭУ») |

**Институт цифровых технологий и экономики**

**Кафедра «Цифровые системы и модели»**

# **Дисциплина: Объектно-ориентированное программирование**

**Отчет о лабораторной работе № 2**

**Классы. Поля. Методы. Конструкторы.**

**Исполнитель:** Тазеев Р.Р.

**Группа:** ТРП-1-23

**Проверяющий:** Халидов А.А.

**Должность:** доцент кафедры ЦСМ

**КАЗАНЬ – 2024**

**Введение**

В данной лабораторной работе рассматриваются ключевые аспекты объектно-ориентированного программирования (ООП): классы, поля, методы и конструкторы. Эти элементы составляют основу ООП и обеспечивают модульность, повторное использование кода и более эффективное управление сложностью программ.

Цель работы — изучить процесс создания и взаимодействия объектов, исследовать механизм работы конструкторов, а также закрепить навыки использования методов для обработки данных, хранимых в полях объектов. Выполнение заданий лабораторной работы позволит студентам глубже понять, как разрабатываются и функционируют программные системы на основе классов.

Понимание принципов работы классов, полей, методов и конструкторов является важным шагом на пути освоения более сложных концепций ООП, таких как наследование, полиморфизм и инкапсуляция.

**Вариант 27**

**Задача 1.** Создать класс Numbers. Закрытые поля класса целого типа – а, b. Реализовать конструкторы с параметрами и без параметров, свойства и методы:  
- Метод NewK для задания новых значений полей.

- Метод Print для вывода значений полей на экран.

- Метод без параметров Nod, который вычисляет наибольший общий делитель этих чисел.

- Метод без параметров Nok, который вычисляет наименьшее общее кратное этих чисел.

Код:

class Numbers

    {

        private int a, b;

        public Numbers() { a = 0; b = 0; }

        public Numbers(int a, int b)

        {

            this.a = a;

            this.b = b;

        }

        public int A

        {

            get { return a; }

            set { a = value; }

        }

        public int B

        {

            get { return b; }

            set { b = value; }

        }

        public void NewK(int a, int b)

        {

            this.a = a;

            this.b = b;

        }

        public void Print()

        {

            Console.WriteLine($"A = {a}, B = {b}");

        }

        public int Nod()

        {

            int x = a, y = b;

            while (x != y)

            {

                if (x > y) x -= y;

                else y -= x;

            }

            return x;

        }

        public int Nok()

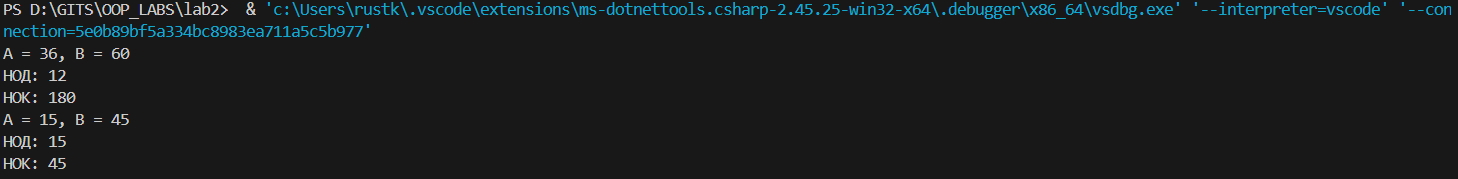
        {

            return a \* b / Nod();

        }

    }

Вывод:



**Задача 2.** Разработать класс для указанного объекта. Включить в класс свойства со средствами доступа **set**, **get**, в которых обеспечить корректную работу с полями. Определить метод вывода значений полей. **Student**: Фамилия, Имя, Отчество, Дата рождения, Адрес, Телефон, Факультет, Курс. В методе **Main()** создать массив объектов. Вывести:

a) список студентов заданного факультета;

б) списки студентов для каждого факультета и курса;

в) список студентов, родившихся после заданного года.

Код:

class Student

    {

        public string LastName { get; set; }

        public string FirstName { get; set; }

        public string MiddleName { get; set; }

        public DateTime BirthDate { get; set; }

        public string Address { get; set; }

        public string Phone { get; set; }

        public string Faculty { get; set; }

        public int Course { get; set; }

        public void Print()

        {

            Console.WriteLine($"{LastName} {FirstName} {MiddleName}, {BirthDate.ToShortDateString()}, {Faculty}, {Course}");

        }

        public static void FilterByFaculty(Student[] students, string faculty)

        {

            var filtered = students.Where(s => s.Faculty == faculty);

            foreach (var student in filtered) student.Print();

        }

        public static void FilterByYear(Student[] students, int year)

        {

            var filtered = students.Where(s => s.BirthDate.Year > year);

            foreach (var student in filtered) student.Print();

        }

    }

Вывод:

Ссылка на файлы лабораторных работ: <https://github.com/ironsast/OOP_LABS>

**Вывод:**

В ходе выполнения лабораторной работы были детально изучены такие основные элементы объектно-ориентированного программирования, как классы, поля, методы и конструкторы. Полученные знания и практический опыт позволили лучше понять принципы создания объектов, а также взаимодействие между элементами класса.

Закрепление этих навыков является важным шагом для освоения более сложных аспектов ООП, что открывает возможности для дальнейшего эффективного использования объектно-ориентированных технологий в разработке программного обеспечения.