



**Автономная некоммерческая образовательная организация
высшего образования Центросоюза Российской Федерации
«Сибирский университет потребительской кооперации»**

Методические указания и задания к практическим
и лабораторным занятиям, самостоятельной работе
по дисциплине:

ОП.10 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

по специальности

**09.02.13 Интеграция решений с применением
технологий искусственного интеллекта**

(направленность программы: Применение искусственного интеллекта)

квалификация выпускника:

Специалист по работе с искусственным интеллектом

Новосибирск
2025

Методические указания и задания к практическим и лабораторным занятиям, самостоятельной работе по дисциплине «*Основы алгоритмизации и программирования*» для обучающихся по специальности 09.02.13 Интеграция решений с применением технологий искусственного интеллекта / [сост. И.Д. Колдунова, канд. пед. наук];– Новосибирск, 2025

РЕЦЕНЗЕНТ: Т.А. Брякотнина, ст. преподаватель кафедры информатики

Методические указания и задания к практическим и лабораторным занятиям, самостоятельной работе рекомендованы к использованию в учебном процессе на заседании кафедры информатики, протокол 28 мая 2025 г., № 9

СОДЕРЖАНИЕ

1. ОБЩИЕ ПОЛОЖЕНИЯ	4
2. ТЕМЫ И ИХ КРАТКОЕ СОДЕРЖАНИЕ	4
3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ И ЗАДАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ (ЛАБОРАТОРНЫХ) И САМОСТОЯТЕЛЬНЫХ РАБОТ	4
4. СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ	18
5. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ РЕСУРСОВ	18
6. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ	19

1. ОБЩИЕ ПОЛОЖЕНИЯ

Дисциплина ОП.10 «Основы алгоритмизации и программирования» относится к общепрофессиональному циклу учебного плана подготовки специалистов по направлению *09.02.13 Интеграция решений с применением технологий искусственного интеллекта*.

Методические указания и задания по выполнению лабораторных и самостоятельных работ по дисциплине «Основы алгоритмизации и программирования» предназначены для работы обучающихся на лабораторных занятиях, самостоятельного изучения дисциплины на основе литературных источников.

В методической разработке предложены вопросы для самоподготовки, задачи и методические указания по их выполнению.

Основной целью изучения дисциплины «Основы алгоритмизации и программирования» является формирование у обучающихся компетенций в области программирования, в том числе с понятием алгоритма, основными алгоритмическими структурами и способами составления алгоритмов.

Задачи освоения дисциплины «Основы алгоритмизации и программирования»:

- формирование у студента навыка перевода конкретной задачи на алгоритмический язык;
- развитие способностей составления кода программы и ее отладки.

2. ТЕМЫ И ИХ КРАТКОЕ СОДЕРЖАНИЕ

Раздел 1.	<i>Введение в программирование</i>
Тема 1.1. Языки программирования	1. Развитие языков программирования.
	2. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы.
	3. Жизненный цикл программы. Программа. Программный продукт и его характеристики.
	4. Основные этапы решения задач на компьютере.
Тема 1.2. Типы данных	1. Типы данных.
	2. Простые типы данных.
	3. Производные типы данных.
	4. Структурированные типы данных.
Раздел 2.	<i>Операторы языка программирования</i>

Тема 2.1. Операторы языка про- граммиро- вания	1. Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор.
	2. Условный оператор. Оператор выбора.
	3. Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные циклы.
	4. Массивы. Двумерные массивы. Строки. Стандартные процедуры и функции для работы со строками.
	5. Структурированный тип данных – множество. Операции над множествами.
	6. Комбинированный тип данных – запись. Файлы последовательного доступа. Файлы прямого доступа
Раздел 3.	<i>Основы структурного и модульного программирования</i>
Тема 3.1. Процедуры и функции	1. Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация функций.
	2. Рекурсия. Программирование рекурсивных алгоритмов.
Тема 3.2. Структури- зация в програм- мировании	1. Основы структурного программирования.
	2. Методы структурного программирования.
Тема 3.3. Модульное програм- мирование	1. Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы.
	2. Стандартные модули.
Раздел 4	<i>Основные конструкции языков программирования</i>
Тема 4.1 Указатели.	1. Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных.
	2. Структуры данных на основе указателей.
	3. Задача о стеке.

Раздел 5	<i>Объектно-ориентированное программирование</i>
Тема 5.1 Основные принципы объектно-ориентированного программирования (ООП)	1. История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.
	2. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.
	3. Классы объектов. Компоненты и их свойства.
	4. Событийно-управляемая модель программирования. Компонентно-ориентированный подход.
Тема 5.2 Интегрированная среда разработчика.	1. Требования к аппаратным и программным средствам интегрированной среды разработчика.
	2. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов.
	3. Панель компонентов и их свойства. Окно кода проекта.
	4. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.
Тема 5.3. Визуальное событийно-управляемое программирование	1. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.
	2. Дополнительные элементы управления. Свойства компонентов. Виды свойств. Синтаксис определения свойств. Назначения свойств и их влияние на результат. Управление объектом через свойства.
	3. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий.
Тема 5.4 Разработка оконного приложения	1. Разработка функционального интерфейса приложения. Создание интерфейса приложения.
	2. Разработка функциональной схемы работы приложения.
	3. Разработка игрового приложения.
Тема 5.5 Этапы разработки	1. Разработка приложения.
	2. Проектирование объектно-ориентированного приложения.
	3. Создание интерфейса пользователя.

приложе- ний	4. Тестирование, отладка приложения.
Тема 5.6 Иерархия классов.	1. Классы ООП: виды, назначение, свойства, методы, события.
	2. Перегрузка методов.
	3. Тестирование и отладка приложения.
	4. Решение задач

3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ И ЗАДАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ (ЛАБОРАТОРНЫХ) И САМОСТОЯТЕЛЬНЫХ РАБОТ

Самостоятельная работа является одним из видов учебной деятельности обучающихся, способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня. Самостоятельная работа проводится с целью:

- систематизации и закрепления полученных теоретических знаний и практических умений, обучающихся;
- углубления и расширения теоретических знаний;
- формирования умений использовать специальную литературу;
- формирования самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации.

Аудиторная самостоятельная работа по учебной дисциплине проводится на учебных занятиях под непосредственным руководством преподавателя и по его заданию (перечень заданий приведен ниже). Внеаудиторная самостоятельная работа выполняется по заданию преподавателя без его непосредственного участия для обучающихся очной и заочной форм обучения.

Основными видами аудиторной самостоятельной работы являются:

- обсуждение теоретических вопросов и решение практических задач по темам дисциплины;
- работа с литературой и другими источниками информации, в том числе электронными.

Решение задач осуществляется на практических занятиях в соответствии с графиком учебного процесса. Для подготовки к практическому занятию обучающиеся должны изучить соответствующую главу основного учебного пособия и рекомендованную специальную литературу.

Практическая задача должна иметь четкую формулировку, к ней должны быть поставлены вопросы, ответы на которые необходимо найти и обосновать. Критерии оценки правильности решения задачи должны быть известны всем обучающимся.

Внеаудиторная самостоятельная работа выполняется по заданию преподавателя.

давателя, но без его непосредственного участия. Перед выполнением внеаудиторной самостоятельной работы преподаватель проводит консультацию с определением цели задания, его содержания, сроков выполнения, ориентировочного объема работы, основных требований к результатам работы, критериев оценки, форм контроля и перечня литературы. В процессе консультации преподаватель предупреждает о возможных типичных ошибках, встречающихся при выполнении задания.

Видами заданий для внеаудиторной самостоятельной работы могут быть:

- конспектирование текста; выписки из текста; работа со словарями и справочниками, нормативно-правовой документацией;
- решение задач.

При выполнении внеаудиторной самостоятельной работы обучающийся имеет право обращаться к преподавателю за консультацией с целью уточнения задания, формы контроля выполненного задания. Контроль результатов внеаудиторной самостоятельной работы студентов может проводиться в письменной, устной или смешанной форме, а также в виде контрольной работы.

3.1. Вопросы для подготовки по темам дисциплины

1. Алгоритмические языки.
2. Аппарат математической логики.
3. Аппликативные языки программирования.
4. Бинарный оператор.
5. Виды кванторов.
6. Визуальное проектирование меню.
7. Возможности VisualStudio.
8. Возможности в VisualStudio по созданию меню.
9. Графический интерфейс разработки и спектр сервисов.
10. Запуск программы в среде разработки VisualStudio.
11. Резервированное или ключевое слово в коде, которое выполняет работу в программе.
12. Иерархическое построение программы.
13. Иерархическое построение программы.
14. Имена объектов формы.
15. Инструменты в среде разработки VisualStudio представленные в виде закладок вдоль границ среды разработки или в нижней части окон инструментов.
16. Команда Run (Запустить) из Debug (Отладка).
17. Дайте определение следующим понятиям: алфавит языка, лексемы, идентификаторы, константы и переменные
18. Что такое типы данных
19. Какие существуют типы данных
20. Основные действия с переменной
21. Способы преобразования данных из одного типа в другой

- 22.Перечислите основные операции.
- 23.Перечислите основные операторы.
- 24.Структура программы на языке программирования
- 25.Команды для ввода и вывод данных
- 26.Перечислите основные алгоритмические структуры
- 27.Операции инкремента и декремента
- 28.Указатели и операции с ними
- 29.Понятие препроцессора
- 30.Трансляторы, определение, виды и назначение
1. Синтаксис и принцип работы операторов выбора
2. Определение и виды циклов
3. Синтаксис и принцип работы цикла с предусловием
4. Синтаксис и принцип работы цикла с постусловием
5. Назначение директивы `#include`
6. Назначение директивы `#define`
7. Работа с массивами
8. Особенности ввода и вывода символов и строк
9. Функции работы со строками
- 10.Структуры: назначение, объявление, определение
- 11.Конкатенация (сцепление).
- 12.Линейный порядок следования операторов в программе.
- 13.Логические операторы.
- 14.Массивы переменных.
- 15.Мерность массива. Основные принципы ООП.
- 16.Создание пользовательских классов.
- 17.Методы и свойства класса.
- 18.Спецификаторы доступа.
- 19.Интерфейс и инициализация класса.
- 20.Обзор средств разработки.
- 21.Объявление, определение, реализация.
- 22.Указатель `this`.
- 23.Принципы наследования и композиции при конструировании новых классов.
- 24.Преимущества и недостатки наследования по сравнению с композицией.
- 25.Инициализация данных класса. Конструктор по умолчанию, конструктор с параметрами.
- 26.Вызов конструкторов объектов, являющихся членами класса.
- 27.Статические методы и свойства класса. Константные методы.
- 28.Назначение деструктора. Порядок вызова деструкторов при наследовании.
- 29.Виртуальные деструкторы.
- 30.Объектно-ориентированный подход к программированию.
- 31.Определение инкапсуляции, полиморфизма и наследования
- 32.Класс как тип данных
- 33.Создание объектов (экземпляров) класса

34. Доступ к членам класса
35. Назначение и свойства Конструкторов
36. Применение конструкторов копирования и преобразования
37. Назначение и свойства Деструктора
38. Когда вызываются конструкторы и деструкторы
39. Передача объектов класса в функции
40. Иерархия классов
41. Вызов конструкторов и деструкторов
42. Перегрузка оператора присваивания
43. Дружественные функции и дружественные классы.
44. Перегрузка операторов. Перегрузка операторов с помощью добавления методов класса. Перегрузка операторов с помощью дружественных функций.
45. Виртуальные функции, понятие полиморфизма. Чистые виртуальные функции и абстрактные классы. Виртуальный деструктор.
46. Множественное наследование. Виртуальное наследование.
47. Обобщенное программирование. Шаблоны классов.
48. Обобщенное программирование. Шаблоны функций.
49. Понятие исключения. Механизм генерации и обработки исключений.
50. Визуально-событийный подход к программированию.
51. Компоненты видимые и невидимые
52. Структура класса WindowsForm
53. Основные объекты управления, их свойства и события.
54. Элемент управления Label, его свойства, методы, события
55. Элемент управления Button, его свойства, методы, события
56. Элемент управления TextBox, его свойства, методы, события
57. Элемент управления CheckBox, его свойства, методы, события
58. Этапы создание приложение из нескольких форм
59. Способы взаимодействия форм
60. Какой класс (относится к пространству имен System.Windows.Forms) позволяет создавать диалоговые окна различных видов
61. Создание в Microsoft Visual Studio главной кнопочной формы
62. Создание простых ленточных форм для работы с данными
63. Какой метод закрывает приложение по окончании его использования сессий OfficeAutomation
64. Обработка текстовых строк
65. Обработчик события
66. Свойства множеств.
67. Свойство элемента, выбранного в списке.
68. Сколько подмножеств у множества
69. Обзор языка XML.
70. Способы работы с XML. Схемы, валидация.

3.2. Задачи к лабораторным занятиям и самостоятельной работе

Задания по теме «Алгоритмы»

Задание 1. У исполнителя Утроитель две команды, которым присвоены номера:

1. вычти 2
2. умножь на три

Первая из них уменьшает число на экране на 2, вторая – утраивает его.

Запишите порядок команд в программе получения из **11** числа **13**, содержащей не более 5 команд.

Задание 2. Для составления цепочек разрешается использовать бусины 5 типов, обозначаемых буквами А, Б, В, Е, И.

Каждая цепочка должна состоять из трех бусин, при этом должны соблюдаться следующие правила:

- а) на первом месте стоит одна из букв: А, Е, И,
- б) после гласной буквы в цепочке не может снова идти гласная, а после согласной – согласная,
- в) последней буквой не может быть А.

Какая из цепочек построена по этим правилам?

- 1) **АИБ** 2) **ЕВА** 3) **БИВ** 4) **ИБИ**

Задание 3. Предлагается некоторая операция над двумя произвольными трехзначными десятичными числами:

1. Записывается результат сложения старших разрядов этих чисел.
2. К нему дописывается результат сложения средних разрядов по такому правилу: если он меньше первой суммы, то полученное число приписывается к первому слева, иначе – справа.
3. Итоговое число получают приписыванием справа к числу, полученному после второго шага, сумму значений младших разрядов исходных чисел.

Какое из перечисленных чисел могло быть построено по этому правилу?

- 1) **141819** 2) **171418** 3) **141802** 4) **171814**

Задание 4. Путешественник пришел в 08:00 на автостанцию поселка ЛЕСНОЕ и увидел следующее расписание автобусов:

Отправление из	Прибытие в	Время отправления	Время прибытия
ЛЕСНОЕ	ОЗЕРНОЕ	07:45	08:55
ЛУГОВОЕ	ЛЕСНОЕ	08:00	09:10
ПОЛЕВОЕ	ЛЕСНОЕ	08:55	11:25
ПОЛЕВОЕ	ЛУГОВОЕ	09:10	10:10
ЛЕСНОЕ	ПОЛЕВОЕ	09:15	11:45
ОЗЕРНОЕ	ПОЛЕВОЕ	09:15	10:30
ЛЕСНОЕ	ЛУГОВОЕ	09:20	10:30
ОЗЕРНОЕ	ЛЕСНОЕ	09:25	10:35
ЛУГОВОЕ	ПОЛЕВОЕ	10:40	11:40
ПОЛЕВОЕ	ОЗЕРНОЕ	10:45	12:00

Определите самое раннее время, когда путешественник сможет оказаться в пункте ПОЛЕВОЕ согласно этому расписанию.

Задание 5. Для 5 букв латинского алфавита заданы их двоичные коды (для некоторых букв - из двух бит, для некоторых - из трех). Эти коды представлены в таблице:

a	b	c	d	e
000	110	01	001	10

Определите, какой набор букв закодирован двоичной строкой **1100000100110**.

1) **baade** 2) **badde** 3) **bacde** 4) **bacdb**

Задание 6. Для кодирования сообщения, состоящего только из букв А, В, С, D и Е, используется неравномерный по длине двоичный код:

A	B	C	D	E
000	11	01	001	10

Какое (только одно!) из четырех полученных сообщений было передано без ошибок и может быть декодировано:

1) **110000010011110**

2) **110000011011110**

3) **110001001001110**

4) 110000001011110

Задание 7. Считая, что каждый символ кодируется одним байтом, оцените информационный объём следующего предложения в кодировке КОИ-8 (в байтах):

Сегодня метеорологи предсказывали дождь.

Задание 8. Считая, что каждый символ кодируется 16-ю битами, оцените информационный объём следующего предложения в кодировке Unicode (в байтах):

Каждый символ кодируется восемью битами.

Задания по теме «Структура программирования»:

1. Два автомобиля движутся навстречу друг другу. Известно расстояние между ними, а также их скорости (вводятся с клавиатуры). Определите время, через которое они встретятся.
2. Найдите длину окружности и площадь ограниченного ею круга, если известен радиус (вводится с клавиатуры).
3. Найдите расстояние между двумя точками с координатами (x1,y1) и (x2,y2). Координаты точек вводятся с клавиатуры.
4. Создайте программу, осуществляющую деление двух обыкновенных дробей. Например: ввод – 2/3, 3/5, результат = 10/9.

5.
$$\frac{z = \frac{\sqrt{2}}{2} \sin x}{2}$$

Задания по теме «Базовые алгоритмические конструкции»:

1. Даны два натуральных числа **X** и **Y**. Найдите результат возведения числа **X** в степень **Y**.
2. Даны два целых числа **A** и **B**. Выведите в порядке возрастания все целые числа, расположенные между **A** и **B** (включая сами числа **A** и **B**), а также посчитайте количество этих чисел.
3. Найдите максимальное из введенных чисел (0 - признак окончания ввода).
4. Даны два целых числа **A** и **B**. Найдите произведение всех целых чисел, расположенных между **A** и **B**, включая их самих.
5. По заданному натуральному **N** найдите **N!**.

6. Определить наибольший общий делитель (НОД) двух натуральных чисел **M** и **N** используя алгоритм Евклида.
7. Вычислите сумму чисел кратных **3** и меньших заданного натурального числа.

Задания по теме «Функции и файлы»:

Задача 1

Дан файл, в котором через пробел записаны числа. Вывести на экран сумму всех чисел. `atoi`

Пример

Дано:	Результат:
21 35 456 7 89	Сумма=608

Задача 2 Найти сумму цифр чисел, записанных в файл

Дан файл, в котором через пробел записаны натуральные числа. Вывести на экран суммы цифр каждого числа.

Пример

Дано:	Результат:
21 35 456 7 89	21=3 35=8 456=15 7=7 89=17

Задача Определить, сколько раз в тексте встречаются заданные слова

Дан файл с учебным расписанием. Написать программу, которая определяет количество лекционных, практических и лабораторных занятий. Вывести их количество на экран. `strcmp, strstr`

Пример обрабатываемого файла:

Понедельник

Физика (лекц.)

Физика (лаб.)

Алгебра (практ.)

Вторник

Геометрия (лекц.)

Физика (практ.)

Физика (лаб.)

Физкультура (практ.)

Среда

Алгебра (лекц.)

Алгебра (практ.)
Геометрия (практ.)

Четверг

Теория струн (лекц.)
Моделирование (лаб.)
Психология (лекц.)
Физкультура (практ.)

Пример работы программы:

Лекций: 5
Практических: 6
Лабораторных: 3

Задача Подсчет количества символов, строк и слов в текстовом файле

Посчитать сколько содержится в файле символов, строк и слов. Разделителями слов считать пробелы и перенос строки.
strlen, getline

Задания по теме «Базовые понятия ООП»:

1. Создайте структуру с именем `student`, содержащую поля: фамилия и инициалы, номер группы, успеваемость (массив из пяти элементов). Создать массив из десяти элементов такого типа, упорядочить записи по возрастанию среднего балла. Добавить возможность вывода фамилий и номеров групп студентов, имеющих оценки, равные только 4 или 5.
2. Создайте структуру с именем `train`, содержащую поля: название пункта назначения, номер поезда, время отправления. Ввести данные в массив из пяти элементов типа `train`, упорядочить элементы по номерам поездов. Добавить возможность вывода информации о поезде, номер которого введен пользователем. Добавить возможность сортировки массив по пункту назначения, причем поезда с одинаковыми пунктами назначения должны быть упорядочены по времени отправления.
3. Создать класс с двумя переменными. Добавить функцию вывода на экран и функцию изменения этих переменных. Добавить функцию,

которая находит сумму значений этих переменных, и функцию которая находит наибольшее значение из этих двух переменных.

4. Описать класс, реализующий десятичный счетчик, который может увеличивать или уменьшать свое значение на единицу в заданном диапазоне. Предусмотреть инициализацию счетчика значениями по умолчанию и произвольными значениями. Счетчик имеет два метода: увеличения и уменьшения, — и свойство, позволяющее получить его текущее состояние. Написать программу, демонстрирующую все возможности класса.
5. Создать класс с двумя переменными. Добавить конструктор с входными параметрами. Добавить конструктор, инициализирующий члены класса по умолчанию. Добавить деструктор, выводящий на экран сообщение об удалении объекта.
6. Создать класс, содержащий динамический массив и количество элементов в нем. Добавить конструктор, который выделяет память под заданное количество элементов, и деструктор. Добавить методы, позволяющие заполнять массив случайными числами, переставлять в данном массиве элементы в случайном порядке, находить количество различных элементов в массиве, выводить массив на экран.

Задания по теме «Интерфейс пользователя»:

ЗАДАНИЕ 1: Создание простой формы. В поле ввода должно вводиться имя, после нажатия клавиши Enter выводится окно сообщения с приветствием. По кнопке Закр^ыть форма закрывается.

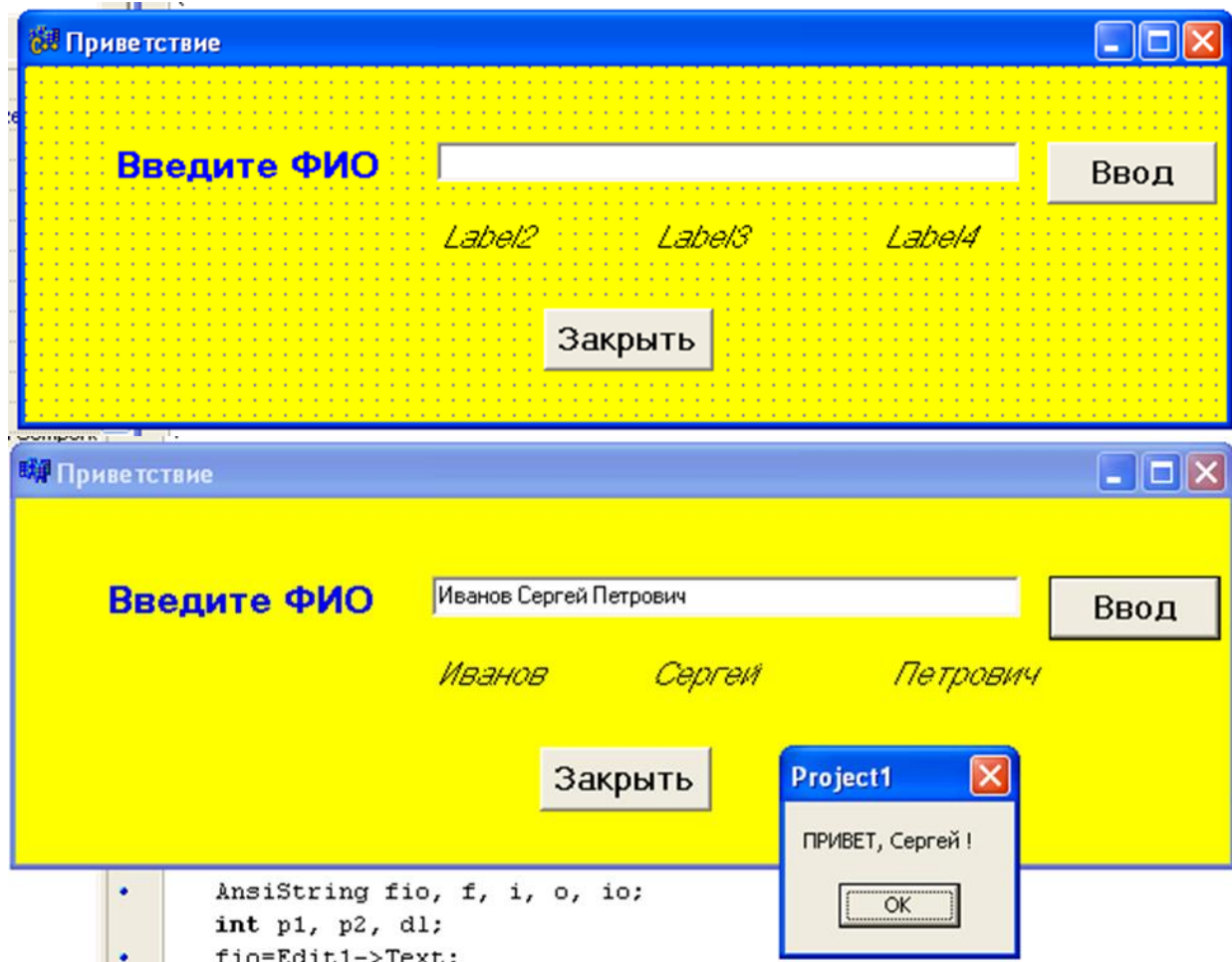


Диалоговое окно вызывается:

```
MessageBox::Show("текст"+textBox1->Text);
```

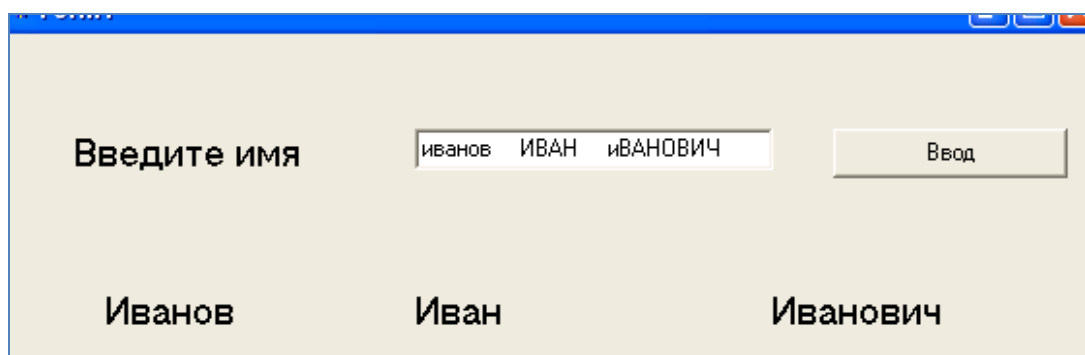
Код клавиши **Enter** 13.

ЗАДАНИЕ 2: Создание простой формы. В поле ввода вводится фамилия, имя и отчество. После нажатия кнопки *Ввод* данные поля *ФИО* «разносятся» в соответствующие надписи. Вызывается окно сообщений, в котором приветствуется человек только по имени. По кнопке *Заккрыть* форма закрывается.



ЗАДАНИЕ 3. Изменение простой формы.

1. Если в поле ввода ничего не введено, то выводится сообщение об отсутствии текста и функция завершает работу.
2. Не зависимо от регистра ввода текста (ФИО) данные в метках (надписях) должны отобразиться с большой буквы.



3. Изменить свойства надписей 2, 3, 4 чтобы при наведении мыши появлялось пояснение: Фамилия, Имя, Отчество, соответственно.

4. СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

4.1 Основная учебная литература

Голицына, О. Л. Основы алгоритмизации и программирования : учебное пособие / О.Л. Голицына, И.И. Попов. — 4-е изд., испр. и доп. — Москва : ФОРУМ : ИНФРА-М, 2021. — 431 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-570-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1150328>

2. Канцедаль, С. А. Алгоритмизация и программирование : учебное пособие / С. А. Канцедаль. — Москва : ФОРУМ : ИНФРА-М, 2021. — 352 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0727-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1189320>

3. Колдаев, В. Д. Основы алгоритмизации и программирования : учебное пособие / В. Д. Колдаев ; под ред. проф. Л. Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2021. — 414 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0733-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1151517>

Дополнительная учебная литература

4. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ФОРУМ : ИНФРА-М, 2021. — 343 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-553-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1217737>

5. Жуков, Р. А. Язык программирования Python. Практикум : учебное пособие / Р. А. Жуков. — Москва : ИНФРА-М, 2021. — 216 с. — (Среднее профессиональное образование). - ISBN 978-5-16-015638-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1190676>

6. Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке C++ : учебное пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2021. — 512 с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - ISBN 978-5-8199-0699-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1172261>

5. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ РЕСУРСОВ

- Интернет-университет информационных технологий: www.intuit.ru;
- Лекции по теории алгоритмов: www.twirpx.com/files/informatics;
- Поисковая система Google: www.google.ru;
- Портал Центра Информационных Технологий: www.citforum.ru.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

№ п/п	Темы дисциплины	Перечень учебно-методических материалов
Тема 1.1.	Языки программирования	1,2,4,5
Тема 1.2.	Типы данных	2,3
Тема 2.1.	Операторы языка программирования	1,4,5
Тема 3.1.	Процедуры и функции	1,2,6
Тема 3.2.	Структуризация в программировании	1,4,5
Тема 3.3.	Модульное программирование	1,2,4,5
Тема 4.1	Указатели.	1,2,3,5,6
Тема 5.1	Основные принципы объектно-ориентированного программирования (ООП)	1,2,3,5,6
Тема 5.2	Интегрированная среда разработчика.	1,2,3,5,6
Тема 5.3.	Визуальное событийно-управляемое программирование	1,2,4,5
Тема 5.4	Разработка оконного приложения	1,3,4,5
Тема 5.5	Этапы разработки приложений	1,3,4,5
Тема 5.6	Иерархия классов.	1,3,4,5