

**МИНОБРНАУКИ РОССИИ**  
**Ярославский государственный университет им. П.Г. Демидова**

Кафедра компьютерной безопасности и математических методов обработки информации

УТВЕРЖДАЮ

Декан математического факультета



Нестеров П.Н.

21 мая 2024 г.

**Рабочая программа дисциплины**

**Информатика**

Направление подготовки (специальности)  
10.03.01 Информационная безопасность

Направленность (профиль)  
«Безопасность компьютерных систем (в сфере информационных технологий)»

Форма обучения очная

Программа рассмотрена  
на заседании кафедры  
от 26 апреля 2024 г., протокол № 8

Программа одобрена НМК  
математического факультета  
протокол № 9 от 3 мая 2024 г.

## 1. Цели освоения дисциплины

Дисциплина "Информатика" нацелена на подготовку студентов к деятельности, связанной с разработкой программного обеспечения для решения профессиональных задач, и является одним из основных предметов, способствующих развитию информационной и алгоритмической культуры и компьютерной грамотности студентов.

Курс способствует формированию фундаментальной базы, необходимой для успешного освоения как общепрофессиональных, так и специальных дисциплин, изучение которых связано с созданием информационных систем для различных предметных областей, их анализом, внедрением и сопровождением.

Задачи дисциплины:

- изучение основных понятий информатики;
- изучение свойств и способов записи алгоритмов;
- изучение способов представления чисел, символов, графики, аудио- и видеоинформации в персональном компьютере;
- ознакомление с логическими основами устройства ЭВМ;
- ознакомление с составом и назначением функциональных узлов компьютера;
- изучение основ построения операционных систем (ОС);
- изучение основ программирования;
- овладение навыками применения сервисных программных средств системного и прикладного назначения;
- изучение основ построения компьютерных сетей;
- овладение навыками поиска информации в глобальной информационной сети Интернет.

## 2. Место дисциплины в структуре образовательной программы

Дисциплина «Информатика» относится к обязательной части образовательной программы. Для успешного усвоения данной дисциплины необходимо, чтобы студент владел знаниями, умениями и навыками, сформированными в процессе изучения программы общеобразовательной школы.

Дисциплина «Информатика» является предшествующей для изучения следующих дисциплин: «Языки программирования», «Аппаратные средства вычислительной техники», «Основы информационной безопасности», «Операционные системы», «Компьютерные сети».

Знания и практические навыки, полученные в результате освоения дисциплины, используются студентами при разработке курсовых и дипломных работ, в научно-исследовательской работе.

## 3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Процесс изучения дисциплины направлен на формирование следующих компетенций в соответствии с ФГОС ВО, ООП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
Общепрофессиональные компетенции		

<b>ОПК-7</b> Способен использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности	<b>И-ОПК-7.1</b> Способен формализовать задачу, разработать алгоритм ее решения и реализовать его на языках программирования, применяя методы и инструментальные средства программирования	<b>Знать:</b> – средства описания алгоритмов; – различные парадигмы программирования. – общие принципы построения и использования современных языков программирования высокого уровня; – базовые структуры данных. – основы программирования на языке C#; <b>Уметь:</b> – формализовать поставленную задачу; – составлять программы на языке программирования высокого уровня для решения вычислительных задач и задач обработки информации; – проводить оценку сложности алгоритмов; – пользоваться инструментальными средствами для разработки прикладных приложений. – тестировать и отлаживать программы; <b>Владеть:</b> – навыками алгоритмического мышления. – методиками современной технологии программирования.
	<b>И-ОПК-7.2</b> Знает языки программирования высокого и низкого уровней и современные среды разработки программ	<b>Знать:</b> – современные технологии программирования <b>Уметь:</b> – работать с современными системами программирования, включая объектно-ориентированные; – оформлять программную документацию – оценивать качество готового программного продукта <b>Владеть навыками:</b> – разработки алгоритмов решения типовых профессиональных задач; – использования инструментальных средств для создания, отладки и тестирования программ; – работы с научно-технической литературой и технической документацией

#### 4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единиц, 144 акад. часов.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)		Формы текущего контроля успеваемости  Форма промежуточной аттестации (по семестрам)
			Контактная работа		

			лекции	практические	лабораторные	консультации	аттестационные испытания	самостоятельная работа	
1	Вводная лекция	1	2					2	
2	Архитектура и организация ЭВМ	1	2		1			2	
3	Арифметические и логические основы ЭВМ	1	2		4	1		4	Лабораторная работа № 1 Контест № 1
4	Технология решения задач на ЭВМ	1	14		18	4		4	Лабораторная работа № 2 – 4 Контест № 2-5
5	Кодирование, измерение и защита информации	1	2		4	2		4	Лабораторная работа № 5-6 Контест № 6
6	Программное обеспечение информационных систем	1	2		1			4	Лабораторная работа № 7 Контест № 7
7	Введение в операционные системы	1	2		1	1		4	Контест № 8
8	Введение в файловые системы	1	2		1			4	Контест № 9
9	Введение в компьютерные сети	1	2		1			4	Контест № 10
10	Основы Internet	1	2		1			4	
						2	0,5	33,5	экзамен
	<b>Всего</b>		<b>32</b>		<b>32</b>	<b>10</b>	<b>0,5</b>	<b>69,5</b>	

### Содержание разделов дисциплины:

#### Раздел 1. Основные понятия информатики

##### Тема 1. Вводная лекция

Задачи и программа курса. Подходы к определению понятий «информации» и «информатики». Виды и свойства информации. Информационные процессы, технологии, системы, ресурсы. Информатизация общества. Информационные войны.

#### Раздел 2. Основы построения ЭВМ

##### Тема 2. Арифметические и логические основы ЭВМ

Системы счисления, перевод из одной системы счисления в другую. Представление в оперативной памяти персонального компьютера числовой информации. Основные логические устройства компьютера.

##### Тема 3. Архитектура и организация ЭВМ

Обобщенная блок-схема персонального компьютера. Состав и назначение функциональных узлов компьютера. Интерфейсы персонального компьютера (ПК). Организация памяти ПК.

#### Раздел 3. Кодирование информация. Алгоритмы и программирование

##### Тема 4. Технология решения задач на ЭВМ

Понятие алгоритма. Свойства алгоритма. Способы записи алгоритмов. Основные алгоритмические конструкции. Методы разработки алгоритмов. Стандартные структуры данных, алгоритмы сортировки данных, основы программирования.

##### Тема 5. Кодирование, измерение и защита информации

Понятие кодирования информации. Кодирование текстовой, графической, звуковой, видеоинформации. Подходы к измерению информации. Основные понятия информационной безопасности. Основные методы защиты информации. Основы защиты от компьютерных вирусов.

## **Раздел 4. Программное обеспечение ЭВМ**

### **Тема 6. Программное обеспечение информационных систем**

Основные понятия и классификация программного обеспечения. Определение свободного программного обеспечения. Общественная лицензия GNU.

### **Тема 7. Введение в операционные системы**

Понятие операционной системы (ОС). Структура обобщенной операционной системы. Основные компоненты операционной системы. Понятие ядра ОС. Понятие прерывания и процедуры обработки прерывания. Взаимодействие пользователя с ОС. Использование программных средств системного и прикладного назначения (форматирования, дефрагментации, архивации, антивирусной защиты и т.д.). Резервное копирование и восстановление системы. Политики управления доступом в компьютерных системах.

### **Тема 8. Введение в файловые системы**

Понятие файловой системы (ФС). Именованное файлов и каталогов. Назначение основных системных каталогов. Типы файлов. Права доступа к файлам и каталогам. Физическая реализация файловой системы.

## **Раздел 5. Компьютерные сети**

### **Тема 9. Введение в компьютерные сети**

Назначение и классификация компьютерных сетей. Понятие модели взаимодействия открытых систем (OSI). Сетевые архитектуры. Понятие сети Ethernet. Настройка компьютера и сетевого оборудования на пользовательском уровне.

### **Тема 10. Основы Internet**

Протоколы Internet. Адресация в Internet. Система доменных имен. Основы использования Internet для поиска информации, обмена файлами и сообщениями. Основы информационной безопасности при работе в сети.

## **5. Образовательные технологии, в том числе технологии электронного обучения и дистанционные образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине**

В процессе обучения используются следующие образовательные технологии:

**Вводная лекция** – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

**Академическая лекция с элементами лекции - беседы** – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя. Элементы лекции-беседы обеспечивают контакт преподавателя с аудиторией, что позволяет привлекать внимание студентов к наиболее важным темам дисциплины, активно вовлекать их в учебный процесс, контролировать темп изложения учебного материала в зависимости от уровня его восприятия.

**Лабораторная работа** – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

**Консультации** – вид учебных занятий, являющийся одной из форм контроля самостоятельной работы студентов. На консультациях по просьбе студентов рассматриваются наиболее сложные моменты при освоении материала дисциплины, преподаватель отвечает на вопросы студентов, которые возникают у них в процессе самостоятельной работы.

**Контеcт** - cоpевнование по программированию, при котором участники отправляют код с решением данных задач на сайт, предназначенный для их автоматического тестирования.

## **6. Перечень лицензионного и (или) свободно распространяемого программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине**

В процессе осуществления образовательного процесса используются:  
для формирования материалов для текущего контроля успеваемости и проведения промежуточной аттестации, для формирования методических материалов по дисциплине:

- программы Microsoft Office;
- издательская система LaTeX;
- Adobe Acrobat Reader;
- для проведения мастер-классов и подготовки материалов лекций:
- Microsoft Visual Studio.
- для проведения лабораторных занятий:
- MS Windows,
- антивирусное ПО (ESET),
- системные утилиты,
- архиватор - 7Zip (GNU GPL v.3).

## **7. Перечень современных профессиональных баз данных и информационных справочных систем, используемых при осуществлении образовательного процесса по дисциплине (при необходимости)**

В процессе осуществления образовательного процесса по дисциплине используются:

- Автоматизированная библиотечно-информационная система «БУКИ-NEXT»  
[http://www.lib.uniyar.ac.ru/opac/bk\\_cat\\_find.php](http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php)
- Электронная библиотечная система «Лань» <https://e.lanbook.com>
- Электронная библиотечная система «Юрайт» <https://urait.ru>
- Электронная библиотечная система «Консультант студента»  
<https://www.studentlibrary.ru>

## **8. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет» (при необходимости), рекомендуемых для освоения дисциплины**

### **а) основная литература**

1. О. В. Власова, Н. П. Федотова, О. П. Якимова Основы программирования: учебное пособие - Ярославль: ЯрГУ, 2019. <http://www.lib.uniyar.ac.ru/edocs/iuni/20190204.pdf>
2. В. Л. Дольников, О. П. Якимова Основы программирования на языке C#. Примеры лабораторных работ: лабораторный практикум - Ярославль, ЯрГУ, 2010. <http://www.lib.uniyar.ac.ru/edocs/iuni/20100478.pdf>
3. Подбельский В. В. Программирование. Базовый курс C#: учебник для вузов — Москва: Издательство Юрайт, 2020. <https://urait.ru/viewer/programmirovanie-bazovyy-kurs-s-450868>

### **б) дополнительная литература**

1. Е. В. Кудрина, М. В. Огнева. Основы алгоритмизации и программирования на языке С#: учебное пособие для вузов — Москва: Издательство Юрайт, 2022. <https://urait.ru/viewer/osnovy-algoritmizacii-i-programmirovaniya-na-yazyke-c-494874>
2. Волк В. К. Информатика: учебное пособие для вузов — Москва: Издательство Юрайт, 2021. <https://urait.ru/viewer/informatika-467779>
3. Толстобров А. П. Архитектура ЭВМ: учебное пособие для вузов — Москва: Издательство Юрайт, 2022. <https://urait.ru/viewer/arhitektura-evm-496167>
4. Казанский А. А. Программирование на Visual С#: учебное пособие для вузов — Москва: Издательство Юрайт, 2020. <https://urait.ru/viewer/programmirovanie-na-visual-c-451467>
5. О. В. Власова, Н. П. Федотова, О. П. Якимова Основы программирования: учебное пособие - Ярославль: ЯрГУ, 2021.

## **9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине**

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

- учебные аудитории для проведения занятий лекционного типа;
- учебные аудитории для проведения занятий лабораторных работ (лаборатория информационных технологий): компьютерные классы, оборудованные ПЭВМ класса не ниже Intel Pentium IV, 1 Gb RAM, 100G HDD с установленным программным обеспечением: MS Windows 7/10, MS Office, Microsoft Visual Studio 2017,2019. Из расчета одна ПЭВМ на одного человека.
- учебные аудитории для проведения групповых и индивидуальных консультаций,
- учебные аудитории для проведения текущего контроля и промежуточной аттестации;
- помещения для самостоятельной работы;
- помещения для хранения и профилактического обслуживания технических средств обучения.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.

### **Автор(ы):**

доцент кафедры компьютерной безопасности  
и математических методов обработки информации, к. ф.-м. н.,

О. П. Якимова

Старший преподаватель  
кафедры компьютерной безопасности и математических  
методов обработки информации

О. В. Власова

**Приложение № 1 к рабочей программе дисциплины  
«Информатика»**

**Фонд оценочных средств  
для проведения текущего контроля успеваемости  
и промежуточной аттестации студентов  
по дисциплине**

**1. Типовые контрольные задания и иные материалы,  
используемые в процессе текущего контроля успеваемости**

**Лабораторная работа № 1.**

Типы данных. Условный оператор. Циклы  
Лабораторная работа состоит из четырех заданий.

Задание 1. Работа с вещественными числами. Математические функции.

Для выполнения задания необходимо знать:

- диапазоны различных вещественных типов;
- описание и инициализацию переменных вещественных типов;
- форматы представления вещественных чисел: с фиксированной и с плавающей точкой;
- операторы ввода-вывода вещественных чисел в нужном формате, с нужной точностью;
- арифметические операции над переменными вещественных типов;
- принципы сравнения вещественных чисел с учетом погрешности вычислений;
- различные функции для округления вещественных чисел;
- основные математические функции, определенные в классе Math.

В этом задании необходимо ввести значения необходимых переменных и вычислить два выражения A и B, зависящих от этих переменных, вывести значения обоих выражений и разности C между ними в формате с фиксированной точкой с точностью 0,000001.

Выражения могут содержать дроби и корни, поэтому важно помнить об ОДЗ. Поскольку оператор ветвления в этой лабораторной работе не используется, необходимо вывести информацию об ОДЗ на экран.

Следует также помнить о читаемости кода и не записывать вычисление выражения в один оператор.

При составлении системы тестов к заданию может потребоваться ввести углы в градусах, а потом перевести их в радианы.

$$1. A = \left( \frac{4 - 2x + x^2}{4 - 2x} + \frac{6x^2 + 8 + 12x}{4 - x^2} - \frac{x^2 + 2x + 4}{2x + 4} \right)^{-\frac{1}{3}} \cdot (x + 2);$$
$$B = \sqrt[3]{4 - x^2};$$

Задание 2. Решение уравнений. Проверка принадлежности точки области

Для выполнения задания необходимо знать:

- ввод, вывод, инициализацию целых и вещественных переменных;
- правила преобразования переменных различного типа;
- стандартные логические функции: отрицание, конъюнкция, дизъюнкция;
- приоритет логических операций.

Решение уравнений

Необходимо аккуратно рассмотреть все возможные принципиально разные варианты значений коэффициентов или параметров.

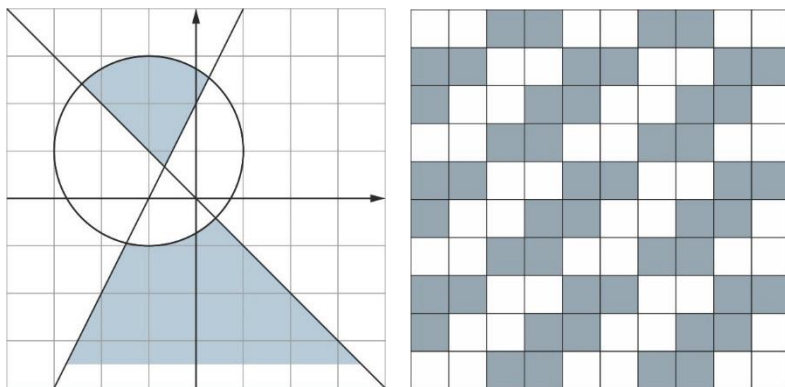
Даны два целых числа: *a* и *b* Решите уравнение  $ax + b = 0$ .

Необходимо вывести все решения, если их число конечно; NO, если решений нет; и INF, если решений бесконечно много.



### Принадлежность точки области

Это задание состоит из двух пунктов. В обоих случаях нужно проверить, принадлежит ли точка закрашенной области на плоскости. Точка задается двумя вещественными координатами. В первом случае - любыми, во втором пункте - неотрицательными координатами. Во втором пункте считаем началом координат левый нижний угол картинки. В заданиях второго пункта, если область ограничена не прямой линией, то это дуга окружности. Часто на рисунке изображена не вся область, а только ее часть, достаточная для обобщения на всю плоскость. Масштаб: 1 - одна клеточка по обеим осям.



### Задание 3. Цикл for. Перебор. Ребусы

В этом задании необходимо написать программу, которая перебором вариантов решает числовой ребус.

В ребусе разным буквам соответствуют разные цифры, и наоборот. Числа не могут начинаться с нуля. Решений может быть несколько, тогда нужно вывести их все.

В некоторых случаях можно ограничить перебор вариантов (когда цифра может быть только четной или заведомо не превосходит какого-то числа).

ДА + ДА = НЕТ

### Задание 4 Цикл for. Печать таблицы истинности булевой функции

Для вычисления логических выражений потребуются логические функции от двух переменных. К основным будем относить:

отрицание (не, !);

конъюнкция (и, &, \*) логическое умножение в формуле может быть опущено;

дизъюнкция (или, +)

Другие логические функции можно выразить через основные. Для каждой из них необходимо реализовать отдельный метод вычисления.

- $x \oplus y = \bar{x}y + x\bar{y}$ ;

- $x \rightarrow y = \bar{x} + y$ ;

- $x \leftrightarrow y = xy + \bar{x}\bar{y}$ ;

- $x | y = \overline{\bar{x}\bar{y}}$ ;

- $x \downarrow y = \overline{x + y}$ ;

Определим следующий приоритет логических операций:

$\neg$ ,  $\wedge$ ,  $\vee(+)$ ,  $\oplus$ ,  $\rightarrow$ ,  $\leftrightarrow$ ,  $|$ ,  $\downarrow$ .

В этом задании необходимо построить таблицы истинности для двух логических выражений А и В.

1.  $A = \bar{x}\bar{z} + xy + x\bar{z}$ ;  $B = z \rightarrow xy$ ;

### Лабораторная работа № 2.

Лабораторная работа 2. Циклы. Последовательности

Лабораторная работа состоит из трех заданий.}

### Задание 1. Цикл while. Сумма ряда

Дана функция, заданная бесконечным рядом (по вариантам).

Ее нужно протабулировать в N точках отрезка [A, B]. Если некоторые точки отрезка не принадлежат указанной в задании области сходимости ряда, необходимо вывести сообщение об этом. Значения функции вычисляются с заданной точностью epsilon.

Таким образом на вход поступают 4 числа: N, A, B, epsilon.

Функцию нужно вычислить тремя способами, а результат вывести на экран в виде таблицы:

x	f1(x)	f2(x)	f3(x)
0.753	0.753	0.753	0.753

Количество знаков после запятой должно соответствовать заданной точности.

Способы вычисления:

1. вызвать стандартную функцию, определенную в пространстве имен Math;
2. вычисление каждого слагаемого по-отдельности, используя самостоятельно написанные для этого функции возведения в степень, вычисления модуля, факториала и, возможно, других функций;
3. слагаемые надо вычислять эффективно, используя результаты, полученные на предыдущем шаге. Без использования вспомогательных функций (за исключением вариантов с числами Бернулли).

Сами функции  $f_i(x)$  необходимо реализовать в виде отдельных методов `double fi(double x, double e)`.

$$1. f(x) = \sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!}, x \in R$$

### Задание 2. Обработка последовательностей

Цель работы. Придумать алгоритм и решить задачу на применение простейших конструкций языка - цикла и условия, осуществляя пошаговый ввод и вывод данных (массив использовать нельзя).

Требуется ввести с клавиатуры натуральное число n, далее в цикле обеспечить ввод и обработку остальных данных.

Даны натуральные числа n,  $a_1, \dots, a_n$ . Определить количество членов  $a_i$  последовательности  $a_1, \dots, a_n$  кратных 3 и не кратных 5.

### Задание 3. Обработка последовательностей

Цель работы. Придумать алгоритм и решить задачу на применение простейших конструкций языка - цикла и условия, осуществляя пошаговый ввод и вывод данных.

Решение задачи должно быть эффективным как по времени работы, так и по используемой памяти. Программу будем считать эффективной по памяти, если используемая память не зависит от размера входных данных. Программу будем считать эффективной по времени, если при увеличении размера входных данных N в t раз (t- любое число) время её работы увеличивается не более чем в t раз.

Дан список точек плоскости с целочисленными координатами. Необходимо определить:

- 1) номер координатной четверти K, в которой находится больше всего точек;
- 2) точку A в этой четверти, наименее удаленную от осей координат;
- 3) расстояние R от этой точки до ближайшей оси.

Если в нескольких четвертях расположено одинаковое количество точек, следует выбрать ту четверть, в которой величина R меньше. При равенстве и количества точек, и величины R необходимо выбрать четверть с меньшим номером K. Если в выбранной четверти несколько

точек находятся на одинаковом минимальном расстоянии от осей координат, нужно выбрать первую по списку. Точки, хотя бы одна из координат которых равна нулю, считаются не принадлежащими ни одной четверти и не рассматриваются. Напишите эффективную, в том числе по памяти, программу, которая будет решать эту задачу.

Описание входных данных

В первой строке вводится одно целое положительное число – количество точек  $N$ . Каждая из следующих  $N$  строк содержит координаты очередной точки – два целых числа (первое – координата  $x$ , второе – координата  $y$ ).

Описание выходных данных

Программа должна вывести номер выбранной четверти  $K$ , количество точек в ней  $M$ , координаты выбранной точки  $A$  и минимальное расстояние  $R$  по образцу, приведённому ниже в примере.

Пример входных данных:

```
7
-3 4
1 2
1 1
0 4
-2 -3
-6 8
-12 1
```

Пример выходных данных для приведённого выше примера входных данных:

```
K = 2
M = 3
A = (-12, 1)
R = 1
```

### **Лабораторная работа № 3.**

Лабораторная работа 3. Одномерные массивы. Датчик случайных чисел. Файлы.

Лабораторная работа состоит из четырех заданий.

Задание 1. Обработка одномерных массивов.

Цель работы. Решить задачу на обработку одномерного массива. Требуется ввести с клавиатуры размерность массива  $n$ , далее в цикле ввести элементы массива. Полученный ответ вывести на экран.

Варианты задания

Даны натуральные  $n$ ,  $m$ , целые числа  $a_1, \dots, a_n$  и  $b_1, \dots, b_m$ . Внутри каждой из данных последовательностей нет повторяющихся членов. Построить объединение данных последовательностей.

Задание 2. Обработка серий подряд идущих элементов.

Цель работы. Решить задачу на обработку одномерного массива. Требуется ввести с клавиатуры размерность массива  $n$ , далее в цикле ввести элементы массива. Полученный ответ вывести на экран. Если в результате работы размер Вашего массива может измениться в сторону увеличения, то следует заранее это учесть, создав массив большего размера.

Определение: Назовем серией группу подряд идущих одинаковых элементов, а длиной серии — количество этих элементов (длина серии может быть равна 1).

Дан целочисленный массив  $A$  размера  $N$ . Сформировать два новых целочисленных массива  $B$  и  $C$  одинакового размера, записав в массив  $B$  длины всех серий исходного массива, а в массив  $C$  — значения элементов, образующих эти серии.

Задание 3. Методы, принимающие и возвращающие массивы в качестве параметров. Файлы.

При выполнении этого задания нужно использовать одномерные массивы и написать методы, принимающие и возвращающие массивы в качестве параметров. Кроме перечисленных в заданиях методов необходимо реализовать методы ввода и вывода массивов. Не следует объединять в один метод реализацию нескольких задач. Массивы и их длины в условии задачи записаны заглавными буквами для лучшей читаемости условия. Если в результате работы размер Вашего массива может измениться в сторону увеличения, то следует заранее это учесть, создав массив большего размера. Исходные данные считываются из файла. Результат записывается в файл.

В соответствии с вариантом Вам необходимо выполнить следующее:

Из целочисленного массива  $X(N)$  все четные элементы записать в массив  $Y(K)$ . Удалить в массивах максимальные элементы (их может быть несколько). В программе написать методы: формирования массива  $Y$ , поиска максимального элемента, удаления элемента, удаления максимальных элементов.

#### Задание 4. Датчик случайных чисел.

Цель работы. Научиться формировать с помощью датчика случайных чисел наборы тестовых данных для проверки корректности работы программы.

Пользователь задает имя тестового набора (NameSet, например test), количество файлов в наборе (N), тип данных в наборе (int, double), диапазон значений элементов в наборе (Min, Max), максимальное и минимально количество элементов в файле (LenMin, LenMax), допустимость повторения элементов (yes, no), необходимость сортировки (asc, desc, none), тип вывода элементов (row, col).

Вам необходимо написать метод, который по заданным параметрам сформирует набор тестовых файлов.

А затем на сформированном наборе тестов проверить работу Вашего кода из заданий 1, 2 и 3.

#### Пример 1:

NameSet = 'test', N=2, Type = 'int' , Min = 2, Max = 10, LenMin=4, LenMax = 10, Repeat = 'yes' , Sort = 'asc', TypeOut = 'row'

Может быть сформирован набор из 2-х файлов

```
test01
4
2 4 7 9
test02
8
3 3 3 6 6 7 8 9
```

#### Пример 2:

NameSet = 'input', N=3, Type = 'double' , Min = 0, Max = 1, LenMin=2, LenMax = 10, Repeat = 'no' , Sort = 'none', TypeOut = 'col'

Может быть сформирован набор из 3-х файлов

```
input01
3
0,24
0,01
0,0123
input02
2
1
0
```

input03

4

0,1

0,2

0,3

0,4

#### *Лабораторная работа № 4*

##### Задание 1.

Цель работы. Решить задачу на обработку текста, используя функции работы со строками. Всюду ниже, если это не оговорено особо, предполагается, что исходным является текстовый файл. Если не оговорено условием, то в текст могут входить слова из латинских букв, цифры, знаки арифметических операций, точка, запятая, пробел.

Требуется считать текст из файла, вывести его на экран, после решения задачи вывести на экран результат. Решение должно быть эффективным!!!

Словом, называется группа букв, за которой и перед которой стоит не буква, а другой символ – цифра, знак, пробел и т.п.

Например, в строке «abc2a, to be, 3red... » ПЯТЬ слов: abc, a, to, be, red.

Варианты.

1. Текстовый файл содержит строку из десятичных цифр, всего не более чем из  $10^6$  символов. Файл образовался в результате последовательной записи «таймкодов» некоторых событий в формате ННММ (часы и минуты слитно по две цифры, т.е. всего 4 цифры на «таймкод», от 0000 до 2359) и прочих случайных данных. Найдите максимально возможное количество подряд идущих «таймкодов» между фрагментами случайной информации. Например, в строке 4212231135414447 можно выделить таймкоды тремя способами: 4[2122]3[1135]4[1444]7, 42[1223,1135]4[1444]7 или 421[2231,1354,1444]7. В последнем случае получилось наибольшее количество таймкодов подряд (3), это число и нужно ввести в ответе.

##### Задание 2.

Цель работы. Реализовать шифр простой замены по заданному алгоритму.

Шифр простой замены — метод шифрования, который заключается в создании таблицы шифрования по определённому алгоритму. В такой таблице каждой букве открытого текста однозначно соответствует буква шифр-текста. Шифрование заключается в замене букв открытого текста согласно таблице. Для расшифровки нужно знать таблицу или алгоритм, по которому она создается. В задании описан алгоритм создания такой таблицы.

В данном задании необходимо выполнить следующее:

Зашифровать оригинальный текст с помощью алгоритма простой замены (по вариантам)

1. Для работы желательно взять осмысленный текст (любимую книгу или отрывок из нее), содержащий не менее 10 страниц,
2. текст нужно прочитать из файла,
3. зашифрованный текст записать в файл.

Расшифровать свой текст, убедиться, что дешифрованный текст совпадает с исходным.

Для того, чтобы облегчить дальнейшую расшифровку текста в задании 2, будем придерживаться следующих соглашений:

- кодируем шифром простой замены только буквы, все остальные символы
- оставляем как есть;
- букву Ё не кодируем, оставляем, как есть;
- регистр символов сохраняем;
- используем предварительно обработанный текст узнаваемого с помощью
- поисковых систем художественного произведения (или его значительной
- части).

Варианты.

1. Шифр Гая Юлия Цезаря. Циклический сдвиг алфавита влево на 3 позиции.

### **Лабораторная работа № 5.**

Цель работы. Решить задачу и отобразить решение графически на экране.

Исходные данные прочитать из текстового файла. Решение должно быть эффективным!!! Используйте структуры для представления геометрических объектов.

Задача решается в 2 этапа.

1 этап: Построение математической модели задачи, разработка алгоритма решения, тестирование.

2 этап: Отрисовка найденного решения на экране в приложении Windows Form.

Задание

Многоугольник на плоскости задается координатами своих  $N$  вершин в порядке обхода их по контуру по часовой стрелке (контур самопересечений не имеет). Для заданной точки  $Z(x, y)$  определить, принадлежит ли она стороне многоугольника или лежит внутри, или вне его.

### **Лабораторная работа № 6.**

#### **Задача 6а.**

Цель работы. Использование подпрограмм в задаче на обработку двумерного массива.

Требуется вывести на экран меню, состоящее из следующих пунктов:

1. ввод матрицы с клавиатуры,
2. ввод матрицы из файла,
3. вычисление характеристики,
4. преобразование матрицы,
5. печать матрицы,
6. выход.

и обеспечить его функционирование.

Внутри программы характеристика оформляется в виде метода с передачей параметров по значению. Этот метод возвращает значение булевского типа; преобразование в виде метода с передачей параметров по ссылке. Необходимо отслеживать, был ли произведен ввод данных до выбора пунктов меню, которые обрабатывают матрицу.

Варианты характеристика/ преобразование.

**Характеристика.** В матрице существует строка, элементы которой образуют симметричную последовательность.

**Преобразование.** Удалить из матрицы строку и столбец, на пересечении которых расположен максимальный по модулю элемент (если таких элементов несколько, то максимальный элемент с меньшими коэффициентами).

#### **Задача 6б.**

В таблице из  $n$  строк и  $n$  столбцов некоторые клетки заняты шариками, другие свободны. Выбран шарик, который нужно переместить, и место, куда его нужно переместить. Выбранный шарик за один шаг перемещается в соседнюю по горизонтали или вертикали свободную клетку. Требуется выяснить, возможно ли переместить шарик из начальной клетки в заданную, и если возможно, то найти путь из наименьшего количества шагов.

### **Лабораторная работа № 7.**

Цель работы. Изучение принципов динамического программирования

Треугольник Паскаля строится следующим образом. Первая строка состоит из одного числа, равного единице. Каждая следующая содержит на одно число больше, чем предыдущая. Первое и последнее из этих чисел равны 1, а все остальные вычисляются как сумма числа, стоящего в предыдущей строке над ним и числа, стоящего в предыдущей же строке слева от него.

### **Входные данные**

Вводится одно число  $N$  ( $0 \leq N \leq 30$ ).

### **Выходные данные**

Вывести  $N$  строк треугольника Паскаля (числа выводятся через пробел).

Примеры

входные данные

5

выходные данные

1

1 1

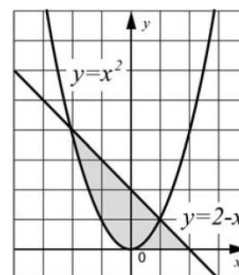
1 2 1

1 3 3 1

1 4 6 4 1

### **Контрольная работа №1**

1. Напишите программу, которая вводит с клавиатуры координаты точки на плоскости ( $x$ ,  $y$  – действительные числа) и определяет принадлежность точки заштрихованной области, включая ее границы.



2. Назовём натуральное число  $N$  ( $1000_8 \leq N \leq 777777_8$ ) счастливым, если суммы двух первых и двух последних цифр его восьмеричной записи равны. Найдите количество таких чисел.

Для проверки числа разработайте функцию **bool isHappy(int n);**

3. Вводится последовательность целых чисел (0 – конец последовательности), найти разность между наименьшим среди положительных и наибольшим среди отрицательных.

4. Дан целочисленный массив из  $N$  элементов, необходимо найти количество элементов в этом массиве, для которых последняя цифра в шестнадцатеричной записи и в восьмеричной записи одинаковая, и заменить все четные элементы массива на это количество. Гарантируется, что такой элемент есть.

Напишите программу для решения этой задачи. В качестве результата программа должна вывести изменённый массив, по одному элементу в строке.

Например, для исходного массива из 5 элементов

**22 38 14 23 11**

программа должна вывести числа **3 3 3 23 11**, по одному числу в строке.

В программе разработайте функцию вычисляющую характеристику, функцию изменения массива и функцию печати массива.

### **Контрольная работа №2**

1. Обработка двумерных массивов

Элементы исходной матрицы вводятся из текстового файла. Результаты выводить на экран и в результирующий текстовый файл. Матрицу выводить до и после преобразований.

Определить сумму и количество положительных чисел расположенных вне диагоналей матрицы  $B(n,n)$ . Если нет положительных чисел, то поменять местами элементы главной и побочной диагоналей.

2. Строки

Описать функцию *IsIdent(S)* целого типа, проверяющую, является ли строка  $S$  допустимым идентификатором (набор букв и цифр, начинающийся с буквы). При утвердительном ответе возвращается 0. Если  $S$  является пустой строкой, то возвращается  $(-1)$ , если строка начинается с цифры, то возвращается  $(-2)$ . Если  $S$  содержит недопустимые символы, то возвращается

номер первого недопустимого символа. Проверить с помощью этой функции пять данных строк.

## **2. Список вопросов и (или) заданий для проведения промежуточной аттестации**

1. Подходы к определению понятий «информации» и «информатики». Виды и свойства информации. Информационные процессы, технологии, системы, ресурсы. Информатизация общества. Информационные войны.
2. Понятие алгоритма. Свойства алгоритма. Способы записи алгоритмов. Методы разработки алгоритмов.
3. Понятие кодирования информации. Кодирование текстовой, графической, звуковой, видеоинформации.
4. Подходы к измерению информации. Основные понятия информационной безопасности.
5. Основные методы защиты информации. Основы защиты от компьютерных вирусов.
6. Системы счисления, перевод из одной системы счисления в другую. Представление в оперативной памяти персонального компьютера числовой информации.
7. Обобщенная схема персонального компьютера. Состав и назначение функциональных узлов компьютера. Оценка производительности ЭВМ. (состав персонального компьютера, назначение и характеристики основных элементов персонального компьютера: центрального процессора и системных шин, системной памяти: ОЗУ, ПЗУ, кэш, назначение и характеристики микропроцессорных систем)
8. Запоминающие устройства: классификация, принцип работы, основные характеристики
9. Устройства ввода/вывода данных, их разновидности и основные характеристики (клавиатура, координатные устройства ввода, видео- и звуковые адаптеры, сканеры, принтеры, плоттеры, мониторы)
10. Основные понятия и классификация программного обеспечения. Определение свободного программного обеспечения. Общественная лицензия GNU.
11. Понятие операционной системы (ОС). Структура обобщенной операционной системы. Основные компоненты операционной системы.
12. Процессы, файлы, потоки. Понятие ядра ОС. Понятие прерывания и процедуры обработки прерывания.
13. Использование программных средств системного и прикладного назначения (форматирования, дефрагментации, архивации, антивирусной защиты и т.д.). Резервное копирование и восстановление системы. Основы администрирования ОС.
14. Понятие файловой системы (ФС). Организация файловых систем на примере ОС WINDOWS. Именованное файлов и каталогов. Назначение основных системных каталогов. Типы файлов. Права доступа к файлам и каталогам. Физическая реализация файловой системы.
15. Назначение и классификация компьютерных сетей. Понятие модели взаимодействия открытых систем (OSI). Сетевые архитектуры. Понятие сети Ethernet.
16. Протоколы Internet. Адресация в Internet. Система доменных имен. Основы использования Internet для поиска информации, обмена файлами и сообщениями. Основы информационной безопасности при работе в сети.
17. Алфавит языка C#. Константы и переменные. Примеры. Стандартные типы данных – целые, действительные, логический, символьный. Операции, функции и процедуры над величинами стандартного типа. Примеры.
18. Структура программы. Пример. Оператор присваивания. Операторы ввода и вывода. Простой и составной оператор. Примеры.



19. Оператор цикла с постусловием, предусловием, с параметром. Операторы завершения цикла. Примеры.
20. Массивы. Сортировка массивов. Алгоритмы сортировки методом прямой выборки, прямого включения, обмена, быстрая сортировка, сортировка слиянием. Бинарный поиск, Бинарный поиск по ответу. Примеры.
21. Строковый тип данных. Операции над строками. Примеры.
22. Структуры. Применение структур при решении геометрических задач.
23. Методы. Область действия имен. Передача параметров по ссылке и по значению. Рекурсия. Примеры.
24. Двумерные массивы. Решение задач на обработку двумерных массивов.

### ***Пример типового экзаменационного билета***

1. Подходы к определению понятий «информации» и «информатики». Виды и свойства информации
2. Написать программу. Даны n, m, и две последовательности из n и из m целых чисел (ввести с клавиатуры). Внутри каждой из последовательностей числа не повторяются. Построить последовательность, которая является пересечением двух данных. Распечатать ее.
3. Написать метод, который переставляет строки матрицы по неубыванию суммы положительных четных элементов в строке. В качестве параметра в метод передается целочисленная квадратная матрица.
4. Какие значения выводит эта программа?

```
static void Swap(int[,] a, ref int[,] b)
{
    int[,] c = b;
    b = a;
    for (int i = 0; i < Math.Min(c.GetLength(0), a.GetLength(0)); i++)
        a[i,0] = c[i,0];
}
static void Print(int[,] mtrx)
{
    for (int i = 0; i < mtrx.GetLength(0); i++)
    {
        for (int j = 0; j < mtrx.GetLength(1); j++)
            Console.Write("{0} ", mtrx[i, j]);
        Console.WriteLine();
    }
}
public static void Main()
{
    int[,] a = new[,] { {6, 2, 3}, {7,4,0} };
    int[,] b = new int[,] { {0,1}, {9,7} };
    Swap(a, ref b);
    Print(a);
    Print(b);
    Console.ReadKey();
}
```

### **Методические указания для студентов по освоению дисциплины**

Основной формой изложения учебного материала по дисциплине «Информатика» являются лекции, причем в форме лекции-беседы или мастер-класса. Это связано с тем, что для формирования алгоритмического мышления необходим опыт формализации задач и их поэтапного решения. По большинству тем предусмотрены лабораторные занятия в смежном курсе «Практикум по основам программирования», на которых происходит закрепление лекционного материала путем применения его к конкретным задачам и отработка навыков работы по разработке алгоритмов, кодированию информации или работы с различным программным обеспечением.

Для успешного освоения дисциплины очень важно решение достаточно большого количества задач, требующих разработки алгоритма и написания программы, как в аудитории, так и самостоятельно в качестве домашних заданий. Примеры решения задач разбираются на лекциях и практических занятиях, при необходимости по наиболее трудным темам проводятся дополнительные консультации. Основная цель решения задач – сформировать алгоритмическое мышление. Для решения всех задач необходимо знать и понимать лекционный материал. Поэтому в процессе изучения дисциплины рекомендуется регулярное повторение пройденного лекционного материала. Материал, законспектированный на лекциях, необходимо дома еще раз прорабатывать и при необходимости дополнять информацией, полученной на консультациях, практических занятиях или из учебной литературы.

Большое внимание должно быть уделено выполнению домашней работы. В качестве заданий для самостоятельной работы дома студентам предлагаются задачи, аналогичные разобранным на лекциях и практических занятиях или немного более сложные, которые являются результатом объединения нескольких базовых задач. Задачи сдаются через систему Яндекс-контест, обязательны к решению три задачи из каждого тура.

Для более глубокого усвоения материала в течение обучения проводятся консультации (при необходимости) по разбору заданий для самостоятельной работы, которые вызвали затруднения.

В конце 1 семестра студенты сдают экзамен. Экзамен принимается по экзаменационным билетам, каждый из которых включает в себя один теоретический вопрос и три задачи. На самостоятельную подготовку к экзамену выделяется 3 дня, во время подготовки к экзамену предусмотрена групповая консультация.

Проверку решений задач по различной тематике можно выполнять на специализированных платформах:

1. <https://informatics.msk.ru/>
2. <https://codeforces.com/>

Рекомендуется дополнительно прослушать бесплатные обучающие курсы по языку C++ на платформе Stepik

<https://stepik.org/catalog/search?free=true>