**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Кузбасский гуманитарно-педагогический институт федерального государственного бюджетного образовательного учреждения высшего образования «Кемеровский государственный университет»**

Факультет информатики, математики и экономики

Кафедра информатики и вычислительной техники
им. В.К. Буторина

*А.В. Маркидонов*

# ИНФОРМАТИКА

## Методические указания к выполнению курсовой работы

*для обучающихся по направлению подготовки*

*09.03.01 Информатика и вычислительная техника, профиль «Автоматизированные системы обработки информации и управления»*

*Форма обучения – очная, заочная*

Новокузнецк, 2024

УДК 621.391
ББК 32.811

**Маркидонов А.В.**

Информатика: метод. указ. к выполнению курсовой работы по направлению подготовки 09.03.01 Информатика и вычислительная техника для очной и заочной форм обучения / А. В. Маркидонов; Кузбасский гуманитарно-педагогический институт ФГБОУ ВО «Кемеровский государственный университет». – Электрон. текст. дан. – Новокузнецк: КГПИ КемГУ, 2024. – 60 с.

В методических указаниях приводятся общие положения, определяющие порядок выполнения, представления результатов и защиты курсовой работы, критерии ее оценки, варианты заданий, требования к структуре и содержанию, перечень и характеристика сведений, приводимых в разделах курсовой работы, список рекомендуемой литературы, а также пример выполнения расчетной части работы.

Указания предназначены для студентов очной и заочной формы обучения, обучающихся по направлению подготовки 09.03.01 Информатика и вычислительная техника, профиль «Автоматизированные системы обработки информации и управления».

Рекомендовано
на заседании кафедры
информатики и вычислительной
техники им. В.К. Буторина
«30» августа 2024 г.
Заведующий кафедрой

А.В. Маркидонов

Утверждено
методической комиссией
факультета информатики, математики
и экономики
«12» сентября 2024 г.
Председатель комиссии

И.А. Жибинова



Текст представлен в авторской редакции

# СОДЕРЖАНИЕ

# ВВЕДЕНИЕ

Учебная дисциплина «Информатика» является обязательной дисциплиной комплексного модуля «Математические и общетехнические основы профессиональной деятельности» основной профессиональной образовательной программы вуза для направлений 09.03.01 Информатика и вычислительная техника, профиль «Автоматизированные системы обработки информации и управления». В результате освоения дисциплины у обучающегося должны быть сформирована компетенция ОПК-1 «Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности», ОПК-2 «Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности» и ОПК-3 «Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности».

На заключительном этапе изучения дисциплины студентом осуществляется выполнение курсовой работы, в ходе которого применяются полученные знания и умения при решении комплексных задач, связанных со сферой профессиональной деятельности будущих специалистов. Элемент исследования – неотъемлемая часть курсовой работы.

Целью курсовой работы является моделирование и расчет характеристик дискретных источников сообщений и каналов связи.

В методических указаниях приводятся общие положения, определяющие порядок выполнения, представления результатов и защиты курсовой работы, критерии ее оценки, требования к структуре и содержанию, перечень и характеристика сведений, приводимых в разделах курсовой работы, а также основные теоретические сведения, варианты заданий и примеры их выполнения,

# I. ПОРЯДОК ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

## 1. Общие положения

Руководитель курсовой работы выдает задание на работу (2-я неделя семестра), методические указания по выполнению и оформлению курсовой работы, оказывает студенту помощь в разработке графика и календарного плана на весь период выполнения работы, рекомендует студенту основную литературу, справочные и методические материалы, проводит регулярные консультации по расписанию, проверяет ход выполнения работы.

Ответственность за результаты работы несет студент.

Курсовая работа оформляется в виде пояснительной записки и демонстрационной части (компьютерной презентации), состоящей из набора слайдов и их копий на бумажных носителях. Демонстрационные материалы используются во время защиты работы.

**Обратите внимание!** Общие требования к содержанию и оформлению типовых структурных элементов пояснительной записки к курсовой работе и демонстрационной части приведены в учебно-методическом пособии «Правила оформления учебных работ студентов»[1] и являются обязательными.

Пояснительная записка представляется руководителю на проверку в электронном и распечатанном виде в соответствии с графиком самостоятельной работы студента.

Курсовая работа подлежит защите. Защита курсовой работы осуществляется в назначенное руководителем время. К защите допускаются студенты, представившие оформленную в соответствии с установленными требованиями пояснительную записку к курсовой работе.

На защите заслушивается:

- устный доклад студента о выполненной работе и ее результатах;
- ответы на вопросы присутствующих на защите (руково-

[1] Правила оформления учебных работ студентов [Текст] : учебно-методическое пособие / И. А. Жибинова [и др.] ; Новокузнец. ин-т (фил.) Кемеров. гос. ун-та ; под ред. И. А. Жибиновой. – Новокузнецк: НФИ КемГУ, 2018. – 124 с. – URL: skado.dissw.ru/indicationsvkr/842/ (дата обращения 31.01.2024). – Текст: электронный.

дителя курсовой работы, студентов группы и приглашенных преподавателей) по представленной пояснительной записке и докладу;

- отзыв руководителя курсовой работы;
- дополнительные вопросы и замечания присутствующих на защите;
- ответы студента на замечания и на дополнительные вопросы.

Для доклада основных итогов работы студенту дается 7-10 минут. Основные положения работы при докладе должны быть представлены в виде компьютерной презентации.

Оценка за курсовую работу складывается из следующих показателей:

- степень соответствия пояснительной записки требованиям к содержанию и оформлению;
- уровень усвоения теоретических знаний, показанный при ответе на вопросы при защите;
- уровень практических навыков, контролируемый выполнением расчетной части работы;
- соблюдение установленных сроков выполнения работы.

Руководитель курсовой работы оценивает качество пояснительной записки, доклада, демонстрационного материала, а также ответов на заданные вопросы, учитывая мнения, высказанные в ходе группового обсуждения присутствовавших на защите.

Критерии оценки выполнения студентами курсовых работ следующие.

Оценка «отлично» ставится в том случае, если:

- работа выполнена в установленные сроки, в полном соответствии требованиям к содержанию и оформлению;
- при защите работы студент показывает глубокие знания вопросов темы, свободно оперирует данными, во время доклада результатов использует наглядные пособия (таблицы, схемы, графики и т.п.), доказательно отвечает на вопросы; количество правильных ответов на защите составляет от 80 до 100 процентов;
- расчетная часть курсовой работы выполнена верно.

Оценка «хорошо» ставится, если работа студента удовлетворяет основным требованиям к работе на оценку «отлично», но

в ней допущены несущественные ошибки или недочеты. Количество правильных ответов на защите от 66 до 79 процентов.

Оценка «удовлетворительно» ставится, если:

- без уважительной причины нарушался установленный график выполнения заданий;
- допущены существенные ошибки, работа отличается поверхностным анализом и недостаточно критическим разбором предмета работы, в ней просматривается непоследовательность изложения материала, имеются замечания по содержанию работы и методике анализа;
- оформление пояснительной записки не в полной мере отвечает установленным требованиям;
- при защите студент проявляет неуверенность, показывает слабое знание вопросов темы, не дает полного, аргументированного ответа на заданные вопросы; количество правильных ответов на защите от 50 до 65 процентов.

Оценка «неудовлетворительно» ставится, если:

- без уважительной причины нарушался установленный график выполнения заданий;
- оформление пояснительной записки не в полной мере отвечает установленным требованиям; допущены принципиальные ошибки в выполнении предусмотренных заданий, работа не содержит анализа и практического разбора предмета работы, не отвечает требованиям, изложенным в методических рекомендациях, высказываются сомнения руководителя о достоверности результатов и выводов;
- при защите работы студент затрудняется отвечать на поставленные вопросы и (или) допускает существенные ошибки; количество правильных ответов на защите менее 50 процентов.

Студенту, выполнившему работу в срок, но получившему при защите неудовлетворительную оценку, назначается повторная защита.

## 2. Структура пояснительной записки к курсовой работе

Пояснительная записка должна содержать следующие структурные элементы:

- **титульный лист;**
- **реферат;**
- **содержание;**
- определения;
- обозначения и сокращения;
- **введение;**
- **основная часть;**
- **заключение и выводы;**
- **список литературы;**
- приложения.

Обязательные структурные элементы выделены полужирным шрифтом, остальные включают в пояснительную записку при необходимости.

Наименования структурных элементов текста пояснительной записки, указанные выше, служат заголовками и не нумеруются. Исключение составляет основная часть пояснительной записки. Наименование «Основная часть» в заголовок не выносится; заголовки разделов основной части формулируются в соответствии с ее содержанием и им присваивается сквозная нумерация.

## 3. Содержание структурных элементов пояснительной записки

### 3.1. Титульный лист

Титульный лист является первым листом пояснительной записки и служит источником информации, необходимой для обработки и поиска документа.

Титульный лист следует оформлять по установленной форме и правилам оформления.

Необходимо иметь в виду, что макеты титульных листов периодически пересматриваются и переутверждаются локальными нормативными актами вуза. В связи с этим, следует проверять наличие актуальных изменений в макетах на кафедре и своевременно учитывать эти изменения при оформлении работы.

### 3.2. Реферат

Реферат должен содержать:

- сведения об объеме пояснительной записки, количестве иллюстраций, таблиц, приложений, количестве использованных источников;
- перечень ключевых слов;
- текст реферата.

Перечень ключевых слов должен включать от 5 до 15 слов или словосочетаний из текста пояснительной записки, которые в наибольшей мере характеризуют содержание и обеспечивают возможность информационного поиска. Ключевые слова приводятся в именительном падеже и печатаются строчными буквами в строку через запятые.

Текст реферата должен отражать:

- цель работы и объект исследования;
- метод или методологию проведения работы;
- сведения, раскрывающие содержание основной части работы;
- краткие выводы об особенностях работы, ее новизне, возможности и области применения полученных результатов.

Объем реферата не превышает 1 страницы.

### 3.3. Содержание

В содержании приводятся заголовки всех разделов, подразделов и более мелких рубрик (если они имеют наименование) с указанием номеров страниц, с которых они начинаются. Все приложения должны быть перечислены в содержании работы с указанием их номеров и заголовков. Содержание включают в общее количество страниц записки.

### 3.4. Определения, обозначения и сокращения

В курсовой работе должны применяться научно-технические термины, обозначения, сокращения слов, установленные соответствующими стандартами, а при их отсутствии – общепринятые в научно-технической литературе. Если в тексте используется специфическая терминология, обозначения, сокращения слов, то должны быть даны соответствующие разъяснения.

Определения, необходимые для уточнения или установления используемых терминов приводят в структурном элементе «Определения». Перечень определений начинают со слов: «В

настоящей работе применяют следующие термины с соответствующими определениями».

Перечень обозначений и сокращений, применяемых в работе, содержит структурный элемент «Обозначения и сокращения». Запись обозначений и сокращений проводят в порядке приведения их в тексте с необходимой расшифровкой и пояснениями.

Допускается определения, обозначения и сокращения приводить в одном структурном элементе «Определения, обозначения и сокращения».

Перечень должен располагаться столбцом. Слева в алфавитном порядке приводят сокращения, условные обозначения, символы, и термины, справа – их детальную расшифровку.

### 3.5. Введение

Функциональное назначение введения состоит в подготовке к восприятию основного текста, вовлечению в проблематику содержания курсовой работы. Оно представляет собой одну из наиболее ответственных частей пояснительной записки, поскольку содержит в сжатой форме все положения, обоснованию которых посвящена работа.

Во введении курсовой работы:

- отражается современное состояние научных исследований в области тематики, в рамках которой выполняется курсовая работа, что логично подводит к выявлению ее актуальности и практической значимости;
- определяется объект и предмет исследования, формулируются цели, определяются задачи и методы исследования;
- определяется план исследования, и кратко характеризуются основные разделы пояснительной записки.

Объем введения составляет не более 2-х страниц.

### 3.6. Основная часть

Основная часть состоит из 9 разделов (по числу заданий). В каждом разделе должны быть подробно приведены все вычисления, а также формулы, на основе которых они выполнены. Листинги кодов, применяемых при выполнении отдельных заданий, допускается перенести в раздел Приложения.

Объем основной части – от 20 до 30 страниц.

### 3.7. Заключение и выводы

Заключение – это финальная часть курсовой работы, которая завершает её. В нём кратко и ёмко обобщаются результаты исследования.

В структуре заключения могут содержаться:

- вводный текст о поставленной цели и задачах работы;
- краткое описание объекта и предмета исследования;
- выводы по каждому разделу, начиная с первого, последовательно изложенные;
- заключение о соответствии выполненной работы её плану, поставленной цели и задачам;
- подтверждённые доказательства актуальности и значимости выполненной работы;
- выявленные задачи и направления развития тематики курсовой работы.

Объём заключения в курсовой работе не должен превышать 1–3 страниц.

### 3.8. Список литературы

Список литературы – это одна из составляющих любой научной работы. Это систематизированный перечень использованных источников. Список литературы в курсовой работе оформляется согласно ГОСТ Р 7.0.100–2018 «Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления» и включает не менее 10 источников. При этом вся литература должна быть по теме курсовой, и кроме того, не рекомендуется использовать устаревшие издания.

## 4. Демонстрационный материал

Демонстрационный материал (презентация курсовой работы) должен содержать документы, обеспечивающие наглядное изложение сути курсовой работы, а именно:

- тему работы; фамилию, имя и отчество исполнителя; шифр группы; фамилию, имя, отчество и должность руководителя;
- содержание основной части;
- заключение и выводы.

Требования к содержанию и объему демонстрационного материала определяются студентом совместно с руководителем курсовой работы. При этом следует исходить из того, что представленный демонстрационный материал должен активно и полностью использоваться при докладе в процессе защиты работы.

# II. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

## 1. Случайные процессы. Понятие о цепях Маркова

***Случайные процессы.***

Обобщением случайных величин являются случайные процессы (случайные функции).

***Случайный (стохастический) процесс*** – это процесс (т. е. изменение во времени состояния некоторой системы), течение которого зависит от случая и для которого определена вероятность того или иного его течения. Чаще всего под случайным процессом понимают некоторую случайную величину $X(t)$, меняющуюся с течением времени $t$.

При описании случайных процессов используются понятия сечения и реализации случайного процесса.

Случайная величина $X(t_0)$, в которую обращается случайный процесс при $t = t_0$, называется ***сечением случайного процесса***, соответствующим данному значению аргумента $t$.

***Реализацией (траекторией) случайного процесса*** называется неслучайная функция $x(t)$, в которую превращается случайный процесс $X(t)$ в результате опыта (рис. 1.1). Например, записывая температуру воздуха в зависимости от времени в течение суток можно получить реализацию случайного процесса.

Любой случайный процесс можно рассматривать как совокупность всех его сечений или всех его реализаций.

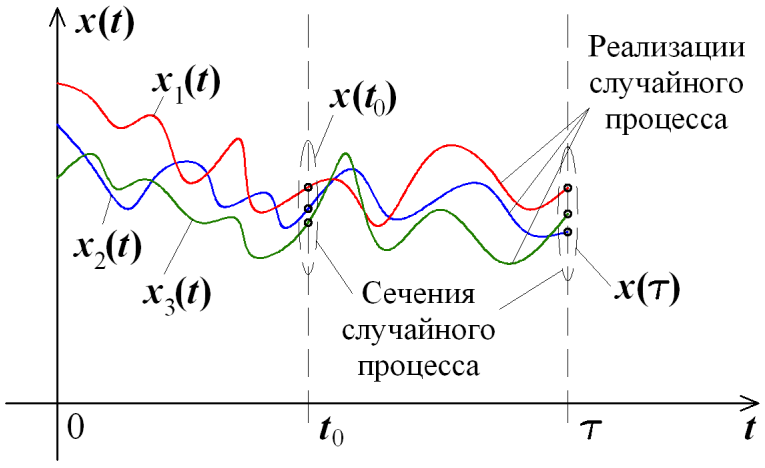


Рис. 1.1. Пример сечений и реализаций случайного процесса

Случайные процессы классифицируют по разным признакам, основными из которых являются классификации по времени $T$ и по состояниям $S$. В зависимости от множества $T$ значений времени $t$, в которые возможны переходы системы из состояния в состояние, а также множества $S$ самих состояний все случайные процессы можно разделить на четыре класса:

• процессы с дискретными состояниями и дискретным временем;

• процессы с дискретными состояниями и непрерывным временем;

• процессы с непрерывными состояниями и дискретным временем;

• процессы с непрерывными состояниями и непрерывным временем.

Простейшим видом случайных процессов являются марковские случайные процессы, широко используемые в теории информации для описания источников сообщений и каналов связи.

***Марковский процесс***[2] – это случайный процесс, эволюция которого после любого заданного значения временного параметра $t$ не зависит от эволюции, предшествовавшей $t$, при условии, что значение процесса в этот момент фиксировано. То есть в марковском процессе будущее состояние зависит только от настоящего состояния и не зависит от «предыстории» процесса.

***Понятие о цепях Маркова.***

Важным классом марковских процессов являются процессы с дискретными состояниями, также называемые цепями Маркова.

***Цепью Маркова*** называется случайный процесс, протекающий в системе с множеством состояний $S = \{s_1, s_2, \dots, s_m\}$ и обладающий следующим свойством: для каждого момента времени $t_0$ вероятность любого состояния системы в будущем ($t > t_0$) зависит только от ее состояния в настоящем ($t = t_0$) и не зависит от того когда и каким образом система пришла в это состояние.

Для анализа цепей Маркова используют ориентированные графы, называемые ***графами состояний*** (рис. 1.2). Вершинами

[2] Название дано в честь русского математика А.А. Маркова (1856 – 1922 гг.), разработавшего теорию данного класса случайных процессов.

такого графа являются состояния системы, а дугами – возможные переходы из одного состояния в другое.

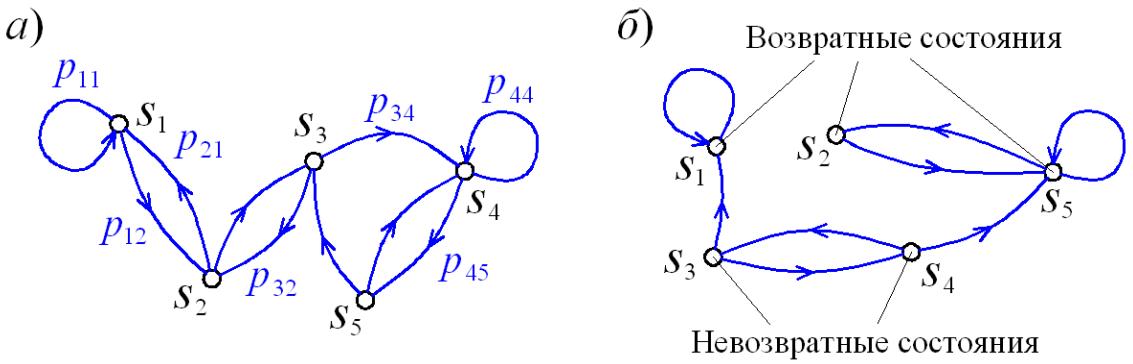


Рис. 1.2. Графы состояния марковских процессов: *а*) – эргодического; *б*) – неэргодического

Каждой дуге ($s_i$, $s_j$) графа состояний приписывается условная вероятность $p_{ij} = p(s_j \mid s_i)$ того, что система перейдет в состояние $s_j$ при условии, что ее текущим состоянием является $s_i$. Указанная вероятность $p_{ij}$ называется ***переходной вероятностью***.

Цепь Маркова может быть описана с помощью ***матрицы переходных вероятностей***, которая содержит все переходные вероятности системы и имеет следующий вид:

$$P_m = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ p_{m1} & p_{m2} & \cdots & p_{mm} \end{bmatrix}.$$

Сумма элементов всех элементов любой строки матрицы $P_m$ должна быть равна 1, поскольку текущее состояние системы должно обязательно перейти в какое-либо состояние:

$$\sum_{j=1}^{m} p_{ij} = \sum_{j=1}^{m} p(s_j \mid s_i) = 1 \qquad (i = 1, 2, \ldots, m).$$

Состояние $s_i$ марковкого процесса называется ***возвратным***, если вероятность возвращения в него через произвольный промежуток времени равна 1, и ***невозвратным***, если эта вероятность меньше 1.

Для теории информации практический интерес представляют только такие марковские процессы, которые являются эргодическими (обладают свойством эргодичности).

Марковский процесс называется ***эргодическим***, если система, в которой он протекает, может из каждого своего состояния перейти в любое другое состояние (непосредственно или через определенное число промежуточных состояний) (рис. 1.2 *а*). В эргодическом марковском процессе не могут присутствовать невозвратные состояния. Марковские процессы, не обладающие таким свойством, называются ***неэргодическими*** (рис. 1.2 *б*).

Для эргодического марковского процесса с течением времени (спустя большое число переходов $k$) система приходит к предельному стационарному распределению $p^*(s_1)$, $p^*(s_2)$, … , $p^*(s_m)$, не зависящему от начального состояния:

$$\lim_{k\to\infty} \frac{k(s_j)}{k} = p^*(s_j);\ \ j = 1, 2, \ldots, m; \qquad (1.1)$$

где $k(s_j)$ – число переходов системы в состояние $s_j$ за общее число переходов $k$.

❒ ***Пример* 1.1.** ***Моделирование цепи Маркова.***

Требуется разработать компьютерную модель цепи Маркова с состояниями $S = \{s_1, s_2, s_3, s_4\}$ и матрицей переходных вероятностей:

| ***i, j*** | $s_1$ | $s_2$ | $s_3$ | $s_4$ |
|---|---|---|---|---|
| $s_1$ | 0,11 | 0,36 | 0,19 | 0,34 |
| $s_2$ | 0,27 | 0 | 0,45 | 0,28 |
| $s_3$ | 0,35 | 0,29 | 0,05 | 0,31 |
| $s_4$ | 0,23 | 0,48 | 0,29 | 0 |

В результате моделирования необходимо получить стационарное распределение вероятностей появления состояний $s_1$, $s_2$, $s_3$, $s_4$. Для этого требуется найти частоты появления состояний при заданном числе шагов моделирования.

Программу для моделирования создадим в форме консольного приложения на языке C#. Модель цепи Маркова реализуем в

форме класса **MarkovChain**, в который будут добавлены следующие открытые методы:

• **string GenerState()**, который случайным образом генерирует и возвращает состояние на основе матрицы переходных вероятностей и предыдущего состояния;

• **string GetFreqs()**, который возвращает строку с частотами появления символов.

Исходный код класса **MarkovChain** представлен в листинге 1.1.

В методе **GenerState()** класса **MarkovChain** производится проверка принадлежности случайного числа **x** определенному интервалу переходных вероятностей. Этот интервал выбирается как строка матрицы **p[m,m]**, которая имеет номер равный значению **k** (индекс предыдущего переданного символа).

Исходный код метода **Main()** консольного приложения приведен в листинге 1.2.

Результат работы консольного приложения при заданной длине исходного сообщения $N = 50000$ символов показан на рис. 1.3.

Стационарное распределение вероятностей появления состояний будет следующим:

$p^*(s_1) = 12101 / 50000 = 0{,}242;$

$p^*(s_2) = 13670 / 50000 = 0{,}273;$

$p^*(s_3) = 12376 / 50000 = 0{,}248;$

$p^*(s_4) = 11853 / 50000 = 0{,}237;$

$\sum_{i=1}^{4} p^*(s_i) = 0{,}242 + 0{,}273 + 0{,}248 + 0{,}237 = 1.$ ❐

Листинг 1.1. Исходный код класса **MarkovChain**

```
/// <summary>
/// Представляет цепи Маркова
/// </summary>
class MarkovChain
{
    const int m = 4; // Число состояний марковской цепи
    string[] s;      // Массив состояний
    double[,] p;     // Матрица переходных вероятностей
    int[] f;         // Массив частот появления состояний
    Random r;        // Случайное поле
    int k;           // Индекс текущего состояния

    /// <summary>
    /// Конструктор с параметрами по умолчанию
    /// </summary>
    public MarkovChain()
    {
        s = new string[m] { "s1", "s2", "s3", "s4" };
        p = new double[m, m] {{0.11, 0.36, 0.19, 0.34},
                              {0.27, 0.00, 0.45, 0.28},
                              {0.35, 0.29, 0.05, 0.31},
                              {0.23, 0.48, 0.29, 0.00}};
        f = new int[m];
        r = new Random();
        k = r.Next(0, m - 1);
    }

    /// <summary>
    /// Генерирует случайным образом состояние цепи и возвращает его
    /// </summary>
    /// <returns>Состояние</returns>
    public string GenerState()
    {
        string st = ""; // Текущее состояние цепи
        double x = r.NextDouble();
        double q = 0.0;  // Кумулятивная вероятность
        for (int i = 0; i < m; i++)
        {
            // Проверяем принадлежность x определенному
            // интервалу переходных вероятностей
            if ((x > q) && (x <= q + p[k, i]))
            {
                st = s[i]; k = i; f[i]++;
                break;
            }
            q += p[k, i];
        }
        return st;
    }

    /// <summary>
    /// Возвращает строку с частотами появления состояний
    /// </summary>
    /// <returns>Частоты появления состояний</returns>
    public string GetFreqs()
    {
        string buf = "\nЧастоты появления состояний:\n";
        for (int i = 0; i < m; i++)
        {
            buf += string.Format("s{0} - {1}\n", i + 1, f[i]);
        }
        return buf;
    }
}
```

Листинг 1.2. Исходный код метода **Main()** консольного приложения

```
10      class Program
11      {
12          static void Main(string[] args)
13          {
14              Console.Title = "Моделирование дискретной случайной величины";
15              MarkovChain r = new MarkovChain(); // Экземпляр класса Марков
16
17              Console.WriteLine("Введите число генерируемых значений:");
18              string buf = Console.ReadLine(); // Строка-буфер
19              int n = int.Parse(buf); // Число генерируемых значений
20              buf = "";
21
22              for (int i = 0; i < n; i++)
23              { buf += r.GenerState(); }
24
25              Console.WriteLine("\nПоследовательность значений:\n" + buf);
26              Console.WriteLine(r.GetFreqs());
27
28              Console.Read();
29          }
```

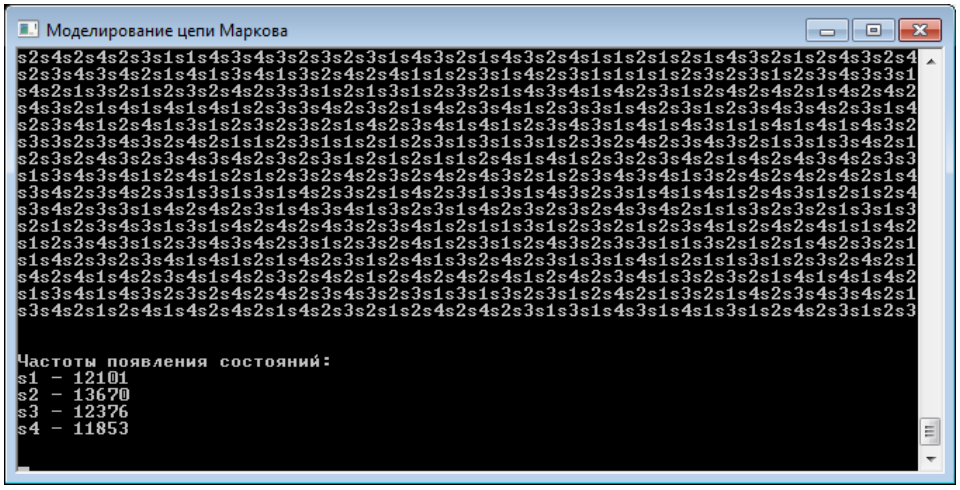


Рис. 1.3. Результат работы консольного приложения

## 2. Моделирование и расчет характеристик дискретных источников сообщений

### 2.1. Дискретные источники сообщений. Характеристики марковских источников сообщений

***Дискретные источники сообщений.***

Одним из основных объектов исследования в теории информации являются ***источники сообщений***, которые представляет собой исследуемый или наблюдаемый объект, формирующий сообщения о своем состоянии. В зависимости от вида формируемых сообщений, различают дискретные и непрерывные источники сообщений.

Источник сообщений, который может в каждый момент времени случайным образом принять одно из конечного множества возможных состояний, называют ***дискретным источником сообщений***. Каждому состоянию источника соответствует условное обозначение в виде передаваемого символа.

Множество всех символов $S = \{s_1, s_2, ..., s_m\}$, доступных источнику сообщений, называют ***алфавитом*** этого источника, а общее число символов $m$ – ***объемом алфавита***.

Простейшим видом дискретных источников сообщений являются ***источники без памяти*** (с нулевой памятью), в которых текущее состояние источника не зависит от его предшествующих состояний.

Дискретный источник сообщений без памяти в общем случае характеризуется ***ансамблем*** $S$, то есть полной совокупностью состояний с вероятностями их появления, составляющими в сумме единицу:

$$S = \begin{pmatrix} s_1 & s_2 & ... & s_m \\ p(s_1) & p(s_2) & ... & p(s_m) \end{pmatrix}, \qquad \sum_{i=1}^{m} p_i = 1.$$

Энтропия $H(S)$ дискретного источника сообщений без памяти, задаваемого ансамблем $S$, может быть определена следующим образом:

$$H(S) = -\sum_{i=1}^{m} p(s_i) \log_2 p(s_i), \text{ (бит/символ);} \qquad (2.1)$$

где $p(s_i)$ – вероятность появления символа $s_i$ из алфавита объемом $m$.

Моделирование дискретного источника сообщений без памяти является идентичным моделированию дискретной случай-

ной величины. В этом случае множество значений случайной величины является алфавитом источника.

Во многих случаях источник сообщений без памяти является грубой моделью реальных источников информации. Текущее состояние многих систем зависит от того, в каких состояниях эти системы находились ранее.

Дискретный источник сообщений называется ***источником с памятью n-го порядка***, если вероятность его перехода в определенное состояние зависит от того, через какие $n$ состояний этот источник последовательно прошел в предшествующие моменты времени.

В теории информации наиболее изученными дискретными источниками сообщений с памятью являются ***марковские источники***. Математическое описание марковских источников основывается на использовании такого класса случайных процессов, как марковские процессы с дискретными состояниями (цепи Маркова).

***Марковские источники сообщений и их характеристики.***

***Марковским источником*** называется дискретный источник сообщений с памятью 1-го порядка, работающий по схеме эргодической цепи Маркова. Графы состояний марковских источников, имеющих два, три и четыре состояния, показаны на рис. 2.1.

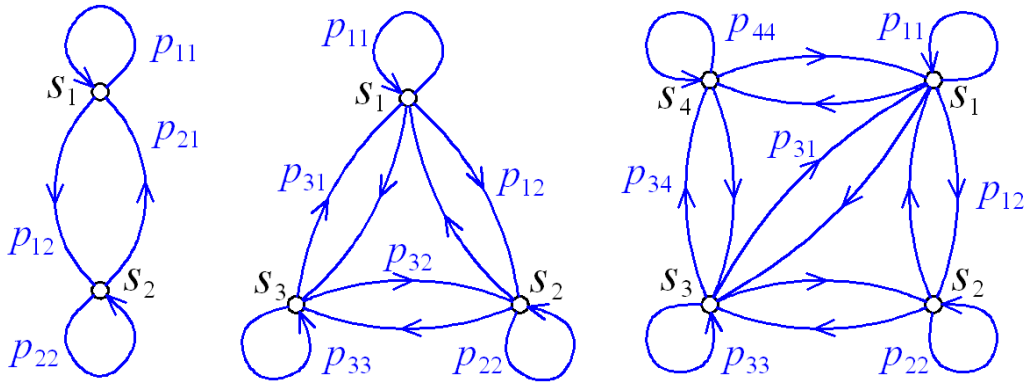


Рис. 2.1. Графы состояний простейших марковских источников

Марковский источник может быть описан с помощью ***матрицы переходных вероятностей***, которая имеет следующий вид:

$$P_m = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ p_{m1} & p_{m2} & \cdots & p_{mm} \end{bmatrix};$$

Сумма элементов всех элементов любой строки матрицы $P_m$ должна быть равна 1, поскольку текущее состояние источника должно обязательно перейти в какое-либо состояние:

$$\sum_{j=1}^{m} p_{ij} = \sum_{j=1}^{m} p(s_j \mid s_i) = 1 \qquad (i = 1, 2, \ldots, m).$$

С учетом статистической связи между парами символов энтропия марковского источника может быть найдена из формулы:

$$H_2(S) = -\sum_{i=1}^{m}\sum_{j=1}^{m} p(s_i)\, p(s_j \mid s_i) \log_2 p(s_j \mid s_i), \text{ (бит/символ);} \quad (2.2)$$

где $p(s_j \mid s_i)$ – условная вероятность появления символа $s_j$ при заданном символе $s_i$.

Для нахождения энтропии марковского источника требуется найти стационарное распределение вероятностей символов $p^*(s_i)$ этого источника.

Помимо энтропии, важными характеристиками источника сообщений, являются информационная эффективность, избыточность и производительность.

***Эффективность*** $E(S)$ источника сообщений $S$ определяется следующим образом:

$$E(S) = \frac{H(S)}{H_{\max}}; \quad (2.3)$$

где $H_{\max} = \log_2 m$ – максимально возможная для данного источника сообщений энтропия, которая достигается при равной вероятности появления символов.

***Избыточность*** $R(S)$ источника сообщений $S$ показывает относительную недогруженность информацией на символ его алфавита. Данная характеристика может быть определена следующим образом:

$$R(S) = 1 - \frac{H(S)}{H_{\max}} = 1 - E(S); \qquad (2.4)$$

Кроме общей информационной избыточности существуют частные избыточности, основными из которых являются:

• Избыточность, вызванная статистической связью между символами $s_1$, $s_2$, ..., $s_m$:

$$R_p = 1 - \frac{H(S)}{H_1(S)}; \qquad (2.5)$$

где $H_1(S) = -\sum_{i=1}^{m} p(s_i)\log_2 p(s_i)$ – энтропия источника сообщений при отсутствии статистической связи между символами.

• Избыточность, вызванная неэкстремальным распределением символов $s_1$, $s_2$, ..., $s_m$ (неравновероятным появлением символов в сообщениях):

$$R_\varphi = 1 - \frac{H_1(S)}{H_{\max}}. \qquad (2.6)$$

Общая информационная избыточность связана с частными избыточностями следующим образом:

$$R(S) = R_p + R_\varphi - R_p R_\varphi. \qquad (2.7)$$

❐ ***Пример* 2.1. *Определение информационных характеристик марковского источника сообщений.***

Требуется определить условную энтропию и избыточность дискретного источника сообщений с памятью, описанного в примере 1.1. Безусловные вероятности появления символов источника в сообщении также возьмем из примера 1.1.

Условная энтропия дискретного источника сообщений с учетом парных сочетаний символов по формуле (2.2) будет:

$$H_2(S) = -[0{,}242\cdot(0{,}11\cdot\log_2 0{,}11 + 0{,}36\cdot\log_2 0{,}36 +$$
$$+0{,}19\cdot\log_2 0{,}19 + 0{,}34\cdot\log_2 0{,}34) + 0{,}273\cdot(0{,}27\cdot\log_2 0{,}27 +$$
$$+ 0{,}45\cdot\log_2 0{,}45 + 0{,}28\cdot\log_2 0{,}28) + 0{,}248\cdot(0{,}35\cdot\log_2 0{,}35 +$$
$$+ 0{,}29\cdot\log_2 0{,}29 + 0{,}05\cdot\log_2 0{,}05 + 0{,}31\cdot\log_2 0{,}31) +$$
$$+ 0{,}237\cdot(0{,}23\cdot\log_2 0{,}23 + 0{,}48\cdot\log_2 0{,}48 + 0{,}29\cdot\log_2 0{,}29)] =$$

$= 0{,}242\cdot(0{,}350 + 0{,}531 + 0{,}455 + 0{,}529) + 0{,}273\cdot(0{,}510 +$

$+ 0{,}518 + 0{,}514) + 0{,}248\cdot(0{,}530 + 0{,}518 + 0{,}216 + 0{,}523) +$

$+ 0{,}237\cdot(0{,}488 + 0{,}508 + 0{,}518) = 0{,}451 + 0{,}421 + 0{,}443 +$

$+ 0{,}359 = 1{,}674$ (бит/символ).

Безусловная энтропия источника может быть определена по формуле (2.1) на основе эмпирических вероятностей из примера 1.1:

$H_1(S) = -(0{,}242\cdot\log_2 0{,}242 + 0{,}273\cdot\log_2 0{,}273 + 0{,}248\cdot\log_2 0{,}248 + 0{,}237\cdot\log_2 0{,}237) = 1{,}998$ (бит/символ).

Набольшая энтропия источника будет:

$$H_{\max}(S) = \log_2 4 = 2 \text{ (бит/символ)}.$$

По формулам (2.3) и (2.4) найдем эффективность и избыточность источника сообщений:

$$E(S) = 1{,}674 / 2 = 0{,}837;$$

$$R(S) = 1 - 0{,}837 = 0{,}163.$$

Частные избыточности источника сообщений по формулам (2.5) и (2.6) будут:

$$R_p = 1 - 1{,}674 / 1{,}998 = 0{,}162;$$

$$R_\varphi = 1 - 1{,}998 / 2 = 0{,}001.$$

По формуле (2.7) получим следующее значение общей избыточности источника сообщений:

$$R(S) = 0{,}162 + 0{,}001 - 0{,}162\cdot 0{,}001 = 0{,}163.$$

Полученное значение совпадает с результатом, найденным по формуле (2.4). ❐

## 2.2. Статистические связанные источники сообщений и их характеристики

***Статистически связанные источники сообщений.***

Между состояниями двух дискретных источников $A$ и $B$ сообщений могут существовать статистические связи. В этом случае вероятности $p(b_j|a_i)$ перехода одного источника (зависимого)

в определенное состояние зависит от того, в каком состоянии находится другой источник.

Указанная статистическая зависимость может быть отображена в виде графа переходов состояний (рис. 2.2).

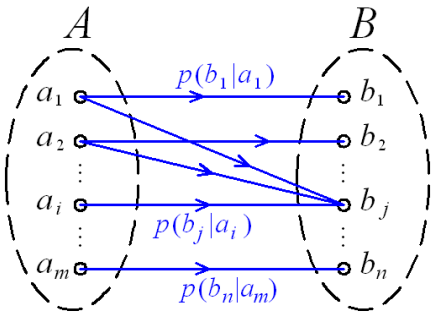


Рис. 2.2. Граф переходов состояний двух статистически взаимосвязанных источников *A* и *B*

Для задания статистической связи между состояниями источников используется матрица переходных вероятностей:

$$P_{mn} = \begin{bmatrix} p(b_1 \mid a_1) & p(b_2 \mid a_1) & \dots & p(b_n \mid a_1) \\ p(b_1 \mid a_2) & p(b_2 \mid a_2) & \dots & p(b_n \mid a_2) \\ \dots & \dots & \dots & \dots \\ p(b_1 \mid a_m) & p(b_2 \mid a_m) & \dots & p(b_n \mid a_m) \end{bmatrix}.$$

❒ ***Пример* 2.2. *Моделирование работы статистически связанных источников сообщений.***

Требуется разработать компьютерную модель дискретного источника сообщений *U*, символы которого статистически взаимосвязаны с символами в сообщениях источника *S* (пример 2.1).

Состояния источника *U* описываются следующей матрицей переходных вероятностей $\boldsymbol{p(s_j|u_i)}$:

| ***i, j*** | $\boldsymbol{u_1}$ | $\boldsymbol{u_2}$ | $\boldsymbol{u_3}$ | $\boldsymbol{u_4}$ | $\boldsymbol{u_5}$ |
|---|---|---|---|---|---|
| $\boldsymbol{s_1}$ | 0,24 | 0,06 | 0,38 | 0 | 0,32 |
| $\boldsymbol{s_2}$ | 0,09 | 0,25 | 0,15 | 0,24 | 0,27 |
| $\boldsymbol{s_3}$ | 0,18 | 0 | 0,33 | 0,19 | 0,3 |
| $\boldsymbol{s_4}$ | 0,1 | 0,25 | 0,29 | 0,36 | 0 |

Исходный код класса **SourceU** представлен в листинге 2.1.

Исходный код класса **Program** консольного приложения приведён в листинге 2.2.

Исходный код класса **SourceS** аналогичен коду класса **MarkovChain**, который представлен в листинге 1.1.

Результат моделирования для 50000 символов показан на рис. 2.3.

Определим эмпирические вероятности появления символов источника $U$:

$p^*(u_1) = 6876 / 50000 = 0{,}138;$

$p^*(u_2) = 7586 / 50000 = 0{,}152;$

$p^*(u_3) = 15592 / 50000 = 0{,}311;$

$p^*(u_4) = 6432 / 50000 = 0{,}129;$

$p^*(u_5) = 13514 / 50000 = 0{,}270;$

$$\sum_{i=1}^{5} p^*(u_i) = 0{,}138 + 0{,}152 + 0{,}311 + 0{,}129 + 0{,}270 = 1.$$

Отсюда безусловная энтропия источника $U$ по формуле (2.1) будет:

$$H(U) = -(0{,}138 \cdot \log_2 0{,}138 + 0{,}152 \cdot \log_2 0{,}152 + 0{,}311 \cdot \log_2 0{,}311 + 0{,}129 \cdot \log_2 0{,}129 + 0{,}27 \cdot \log_2 0{,}27) = 2{,}221 \text{ (бит/символ)}.$$

Определим наибольшую энтропию источника $U$:

$$H_{\max}(U) = \log_2 5 = 2{,}232 \text{ (бит/символ)}.$$

Найдем эффективность и избыточность источника сообщений $U$:

$$E(U) = H(U) / H_{\max}(U) = 2{,}221 / 2{,}232 = 0{,}995;$$

$$R(U) = 1 - E(U) = 1 - 0{,}995 = 0{,}005.$$

Листинг 2.1. Исходный код класса **SourceU**

```
/// <summary>
/// Представляет источники сообщений (источники U), статистически
/// связанные с другими источниками (источники S)
/// </summary>
class SourceU
{
    const int mu = 5; // Объем алфавита источника сообщений U
    const int ms = 4; // Объем алфавита источника сообщений S
    string[] u;       // Алфавит источника сообщений U
    double[,] p;      // Матрица переходных вероятностей
    int[] f;          // Массив частот появления символов
    Random rnd;

    /// <summary>
    /// Конструктор с параметрами по умолчанию
    /// </summary>
    public SourceU()
    {
        u = new string[mu] { "u1", "u2", "u3", "u4", "u5" };
        p = new double[ms, mu] {{0.24, 0.06, 0.38, 0.00, 0.32},
                                {0.09, 0.25, 0.15, 0.24, 0.27},
                                {0.18, 0.00, 0.33, 0.19, 0.30},
                                {0.10, 0.25, 0.29, 0.36, 0.00}};
        f = new int[mu];
        rnd = new Random();
    }

    /// <summary>
    /// Генерирует случайным образом символ и возвращает его
    /// </summary>
    /// <param name="si">Символ источника S</param>
    /// <returns>Символ источника U</returns>
    public string GenerSymbol(string si)
    {
        string uj = ""; // Текущий символ источника U
        int i = Convert.ToInt32(si.Remove(0, 1)) - 1; // Индекс символа источника S
        double q = 0.0;  // Нижняя граница интервала вероятностей
        double r = rnd.NextDouble();
        for (int j = 0; j < mu; j++)
        {
            if ((r > q) && (r <= q + p[i, j]))
            {
                uj = u[j]; f[j]++; break;
            }
            q += p[i, j];
        }
        return uj;
    }

    /// <summary>
    /// Возвращает строку с частотами появления символов
    /// </summary>
    /// <returns>Частоты появления символов</returns>
    public string GetFreqs()
    {
        string buf = "\nЧастоты появления символов:\n";
        for (int j = 0; j < mu; j++)
        {
            buf += string.Format("{0} - {1}\n", u[j], f[j]);
        }
        return buf;
    }
}
```

Листинг 2.2. Исходный код класса **Program** консольного приложения

```
 9      class Program
10      {
11          static void Main(string[] args)
12          {
13              Console.Title = "Моделирование работы статистически связанных источников";
14              Console.WriteLine("Введите число генерируемых символов:");
15              string buf = Console.ReadLine(); // Строка-буфер
16              int n = int.Parse(buf); // Число генерируемых символов
17              buf = "";
18
19              SourceS sourceS = new SourceS();
20              SourceU sourceU = new SourceU();
21              string mes = ""; // Сообщение, выводимое в консоли
22
23              for (int i = 0; i < n; i++)
24              {
25                  buf = sourceS.GenerSymbol();
26                  mes += string.Format(" {0}-->", buf);
27                  buf = sourceU.GenerSymbol(buf);
28                  mes += string.Format("{0}; ", buf);
29              }
30
31              Console.WriteLine(mes);
32              Console.WriteLine(sourceU.GetFreqs());
33              Console.Read();
34          }
35      }
```

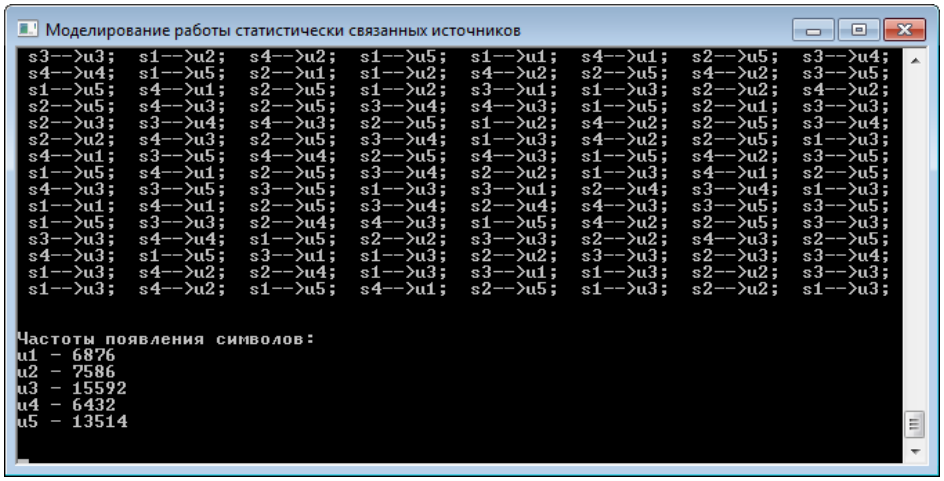


Рис. 2.3. Результат работы консольного приложения

Таким образом, источник $U$ имеет следующие информационные характеристики: $H$ = 2,221 бит/символ, $H_{max}$ = 2,232 бит/символ, $E$ = 0,995, $R$ = 0,005. ❐

***Условная энтропия, энтропия объединения и взаимная информация дискретных источников сообщений.***

***Условная энтропия*** $H(B|A)$ источника сообщений $B$ относительно статистически связанного с ним источника $A$ характеризует степень неопределенности состояния источника $B$, при известном состоянии источника $A$.

Понятие условной энтропии в теории информации используется при определении взаимозависимости между символами кодируемого алфавита, для определения потерь при передаче информации по каналам связи, при вычислении энтропии объединения.

Условная энтропия дискретного источника сообщений $B$ относительно источника $A$ может быть определена следующим образом:

$$H(B \mid A) = -\sum_{i=1}^{m}\sum_{j=1}^{n} p(a_i)\, p(b_j \mid a_i) \log_2 p(b_j \mid a_i) =$$

$$= -\sum_{i=1}^{m}\sum_{j=1}^{n} p(a_i, b_j) \log_2 p(b_j \mid a_i); \text{ (бит/символ).} \quad (2.8)$$

Под ***объединением*** двух источников сообщений $A$ и $B$ с возможными состояниями $a_1, a_2, \ldots, a_m$; $b_1, b_2, \ldots, b_n$ понимается сложный источник $(A,B)$, состояния которого представляют все возможные комбинации состояний $a_i$, $b_j$ источников $A$ и $B$.

$$(A,B) = \begin{pmatrix} (a_1,b_1) & (a_2,b_1) & \ldots & (a_i,b_j) & \ldots & (a_m,b_n) \\ p(a_1,b_1) & p(a_2,b_1) & \ldots & p(a_i,b_j) & \ldots & p(a_m,b_n) \end{pmatrix}.$$

где $p(a_i, b_j)$ – вероятность того, что система $(A,B)$ находится в состоянии $(a_i, b_j)$.

***Энтропия объединения*** $H(A,B)$ источников сообщений $A$ и $B$ характеризует неопределенность того, что источник $(A,B)$ находится в состоянии $(a_i,b_j)$.

Для источников сообщений $A$ и $B$ ***энтропия объединения*** (взаимная энтропия) представляет собой сумму вида:

$$H(A,B) = -\sum_{i=1}^{m}\sum_{j=1}^{n} p(a_i,b_j) \log_2 p(a_i,b_j); \text{ (бит/два символа);} \quad (2.9)$$

где $p(a_i,b_j)$ – вероятность совместного появления в сообщениях символов $a_i$ и $b_j$ из источников $A$ и $B$.

Энтропия объединения и условная энтропия дискретных источников $A$ и $B$ при наличии статистической связи между их состояниями связаны между собой соотношениями:

$$H(A,B) = H(A) + H(B|A) = H(B) + H(A|B); \qquad (2.10)$$

$$H(B|A) = H(A,B) - H(A);$$

$$H(A|B) = H(A,B) - H(B).$$

***Взаимной информацией***, содержащейся в источниках $A$ и $B$, называется величина $I(A,B)$, которая характеризует среднее количество информации, получаемое от одного символа источника $A$ при одном переданном символе источника $B$.

Взаимную информацию можно определить как уменьшение энтропии источника $B$ в результате получения сведений о состоянии источника $A$ и наоборот:

$$I(A,B) = H(B) - H(B|A) = H(A) - H(A|B). \qquad (2.11)$$

Через вероятности состояний источников $A$, $B$ и $(A,B)$ взаимная информация может быть найдена следующим образом:

$$I(A,B) = \sum_{i=1}^{m}\sum_{j=1}^{n} p(a_i,b_j)\log_2 \frac{p(a_i,b_j)}{p(a_i)p(b_j)}; \text{ (бит/символ).} \qquad (2.12)$$

❐ ***Пример* 2.3. *Определение информационных характеристик взаимосвязанных источников сообщений.***

Требуется определить условную энтропию $H(U|S)$ источника сообщений $U$ относительно источника $S$ (пример 2.1), также энтропию объединения $H(S,U)$ указанных двух источников, а также их взаимную информацию $I(S,U)$.

Состояния источника $U$ описываются следующей матрицей переходных вероятностей $\boldsymbol{p(s_j|u_i)}$:

| $i, j$ | $u_1$ | $u_2$ | $u_3$ | $u_4$ | $u_5$ |
|---|---|---|---|---|---|
| $s_1$ | 0,24 | 0,06 | 0,38 | 0 | 0,32 |
| $s_2$ | 0,09 | 0,25 | 0,15 | 0,24 | 0,27 |
| $s_3$ | 0,18 | 0 | 0,33 | 0,19 | 0,3 |
| $s_4$ | 0,1 | 0,25 | 0,29 | 0,36 | 0 |

На основе матрицы переходных вероятностей и стационарного распределения вероятностей источника $S$ (пример 2.1) получим условную энтропию источника $U$ относительно источника $S$ по формуле (2.8):

$$H(U|S) = -[0{,}242\cdot(0{,}24\cdot\log_2 0{,}24 + 0{,}06\cdot\log_2 0{,}06 +$$
$$+ 0{,}38\cdot\log_2 0{,}38 + 0{,}32\cdot\log_2 0{,}32) + 0{,}273\cdot(0{,}09\cdot\log_2 0{,}09 +$$
$$+ 0{,}25\cdot\log_2 0{,}25 + 0{,}15\cdot\log_2 0{,}15 + 0{,}24\cdot\log_2 0{,}24 +$$
$$+ 0{,}27\cdot\log_2 0{,}27) + 0{,}248\cdot(0{,}18\cdot\log_2 0{,}18 + 0{,}33\cdot\log_2 0{,}33 +$$
$$+ 0{,}19\cdot\log_2 0{,}19 + 0{,}3\cdot\log_2 0{,}3) + 0{,}237\cdot(0{,}1\cdot\log_2 0{,}1 +$$
$$+ 0{,}25\cdot\log_2 0{,}25 + 0{,}29\cdot\log_2 0{,}29 + 0{,}36\cdot\log_2 0{,}36)] =$$
$$= 0{,}434 + 0{,}608 + 0{,}483 + 0{,}446 = 1{,}971 \text{ (бит/символ)}.$$

Энтропия объединения по формуле (2.10) может быть найдена через энтропию $H(S)$ источника $S$ (пример 2.2) и условную энтропию $H(U|S)$:

$$H(S,U) = H(S) + H(U|S) = 1{,}674 + 1{,}971 = 3{,}645 \text{ (бит/два символа)}.$$

По формуле (2.11) найдем взаимную информацию источников $S$ и $U$:

$$I(S,U) = H(U) - H(U|S) = 2{,}221 - 1{,}638 = 0{,}583 \text{ (бит/символ)}.$$

Таким образом, взаимосвязанные источники $S$ и $U$ имеют следующие информационные характеристики: $H(U|S)$ = 1,638 бит/символ, $H(S,U)$ = 3,312 бит/два символа, $I(S,U)$ = 0,583 бит/символ. ❐

## 2.3. Коды, их характеристики и виды. Кодирование источников сообщений

***Понятие кода. Характеристики и виды кодов.***

Символы, выдаваемые источником сообщений, обычно подаются на вход кодера, который осуществляет процесс кодирования информации.

Под ***кодированием информации*** понимают процесс преобразования информации из формы, удобной для непосредственного ее использования, в форму, удобную для передачи, хранения

или автоматической обработки. Обратный процесс называется ***декодированием***.

Кодер имеет собственный алфавит символов $C = \{c_1, c_2, ..., c_K\}$, называемый ***вторичным алфавитом*** (за первичный алфавит принимается алфавит источника сообщений). Кроме того, кодер имеет определенный алгоритм кодирования (код).

***Код*** – правило сопоставления каждому конкретному сообщению (символу первичного алфавита) строго определённой комбинации символов вторичного алфавита. Отдельная комбинация таких символов называется ***кодовым словом***. Иногда используют термины «кодовая комбинация», «кодовая последовательность» или «кодограмма».

Основными целями кодирования информации являются:

• преобразование информации в форму, пригодную для её автоматической обработки с помощью технических средств;

• эффективное использование канала связи; уменьшение стоимости передачи и хранения; уменьшение избыточности;

• защита от помех в канале связи; повышение помехоустойчивости и достоверности передачи сообщений;

• криптографическая защита информации; сокрытие смысла передаваемой информации от непосвящённых лиц (шифры).

Основные характеристики кодов:

• ***Основание кода***. Объем алфавита $K$ кодера, то есть число различных символов кода называют ***основанием кода***. Например, если $K = 2, 3, 4, \ldots$, то код называют бинарным (двоичным), триарным (троичным), тетрарным и т. д.

• ***Длина кодового слова*** (разрядность) $L$. Число символов $L$ в кодовом слове называется ***длиной кодового слова***.

Коды, все слова которых имеют одинаковую длину, называются ***равномерными*** (***блоковыми***). Соответственно коды, в которых данное условие не выполняется, называют ***неравномерными***. Классическим примером равномерного кода является пятизначный двоичный код Бодо, используемый в телеграфной связи. К неравномерным кодам относится код Морзе.

• ***Общее число кодовых слов*** $N$ (мощность кода). Если все кодовые слов используются для кодирования сообщений, то такой код называется полным.

Для равномерного кода общее число кодовых слов определяется следующим образом:

$$N = K^L.$$

• ***Взвешенность кода*** – соответствие символов кода весовым коэффициентам системы счисления.

Если все кодовые комбинации кода соответствуют числам выбранной системы счисления, то такие коды ***называются взвешенными*** (арифметическими). Взвешенные коды широко используются для передачи и обработки числовой информации. В противном случае коды называют ***невзвешенными***. Примером взвешенного кода является натуральный двоичный код (НДК), в котором кодовые комбинации соответствуют последовательности натуральных чисел. Примером невзвешенного кода является двоичный код Грэя.

По назначению коды можно разделить на следующие группы:

• ***Коды представления данных*** обеспечивают запись данных в требуемом для автоматической обработки формате.

• ***Эффективные (оптимальные) коды*** позволяют сжать передаваемые сообщения.

• ***Помехоустойчивые коды*** обеспечивают обнаружение и исправление ошибок в полученных сообщениях.

• ***Шифры*** служат для защиты передаваемой информации от прочтения противником.

***Кодирование источников сообщений.***

Первым кодером в системе передачи сообщений является ***кодер источника сообщений***, который служит для представления сообщений от источника в наиболее компактной форме. Это нужно, чтобы максимально эффективно использовать ресурсы канала связи либо запоминающего устройства. Кодеры источника устраняют из сообщений избыточность.

***Декодер источника сообщений*** преобразует закодированное сообщение в исходную форму, пригодную для восприятия получателем.

❒ ***Пример* 2.4. *Моделирование кодера и декодера источника сообщений.***

Требуется разработать модель кодера и декодера для источника сообщений, описанного в примере 2.1.

Примем, что кодер осуществляет простое кодирование символов источников сообщений с помощью следующего двоичного кода:

$$s_1 \rightarrow \mathbf{00}; \quad s_2 \rightarrow \mathbf{01}; \quad s_3 \rightarrow \mathbf{10}; \quad s_4 \rightarrow \mathbf{11}.$$

Соответственно, декодер осуществляет обратное преобразование.

Декодер источника сообщений преобразует кодовые слова в символы источника, а также определяет число кодовых слов, которые из-за ошибок не могут быть декодированы.

В проект консольного приложения из примера 2.1 добавим классы **SourceEncoder** и **SourceDecoder**, представляющие соответственно кодер и декодер источника сообщений.

В классе **SourceEncoder** должен присутствовать метод **string SimpleEncoding(string _s)**, который для заданного символа **_s** возвращает соответствующее ему кодовое слово. В класс **SourceDecoder** добавим метод **string SimpleDecoding(string _c),** который для полученного кодового слова **_c** возвращает требуемый символ источника.

Дополнительно в классе **SourceDecoder** реализуем метод **string GetErrorNumber()**, который возвращает число ошибок при декодировании.

Исходный код классов **SourceEncoder** и **SourceDecoder** приведен в листингах 2.3 и 2.4.

В методе **Main()** консольного приложения выполним форматный вывод в окно консоли строк, возвращаемых методами классов **MarkovChain**, **SourceEncoder** и **SourceDecoder**.

Исходный код метода **Main()** представлен в листинге 2.5.

Результат работы консольного приложения для введенного числа символов показан на рис. 2.4. ❒

Листинг 2.3. Исходный код класса **SourceEncoder**

```
/// <summary>
/// Представляет кодер источника сообщений
/// </summary>
class SourceEncoder
{
    /// <summary>
    /// Выполняет простое кодирование символов источника
    /// </summary>
    /// <param name="s">Символ источника</param>
    /// <returns>Кодовое слово</returns>
    public string SimpleEncoding(string _s)
    {
        string c = ""; // Кодовое слово
        switch (_s)
        {
            case "s1": c = "00"; break;
            case "s2": c = "01"; break;
            case "s3": c = "10"; break;
            case "s4": c = "11"; break;
        }
        return c;
    }
}
```

Листинг 2.4. Исходный код класса **SourceDecoder** (часть 1)

```
/// <summary>
/// Представляет декодер источника сообщений
/// </summary>
class SourceDecoder
{
    int errNum; // Число ошибок декодирования

    /// <summary>
    /// Конструктор по умолчанию
    /// </summary>
    public SourceDecoder()
    { this.errNum = 0; }

    /// <summary>
    /// Выполняет простое декодирование кодовых слов
    /// </summary>
    /// <param name="_c">Кодовое слово</param>
    /// <returns>Символ источника</returns>
    public string SimpleDecoding(string _c)
    {
        string s = ""; // Символ
        switch (_c)
        {
            case "00": s = "s1"; break;
            case "01": s = "s2"; break;
            case "10": s = "s3"; break;
            case "11": s = "s4"; break;
            default:   s = "><"; errNum++; break; // Знак ошибки
        }
        return s;
    }

```

Листинг 2.4. Исходный код класса **SourceDecoder** (часть 2)

```
41          /// <summary>
42          /// Возвращает строку с данными об ошибках декодирования
43          /// </summary>
44          /// <returns>Данные об ошибках</returns>
45          public string GetErrorNumber()
46          {
47              return string.Format("Число ошибок декодирования: " + errNum);
48          }
49      }
```

Листинг 2.5. Исходный код метода **Main()** консольного приложения

```
 9      class Program
10      {
11          static void Main(string[] args)
12          {
13              Console.Title = "Моделирование марковского источника";
14              // Экземпляр класса марковский источник сообщений
15              MarkovChain r = new MarkovChain();
16              // Экземпляр класса кодер источника сообщений
17              SourceEncoder sEnc = new SourceEncoder();
18              // Экземпляр класса декодер источника сообщений
19              SourceDecoder sDec = new SourceDecoder();
20
21              Console.WriteLine("Введите число генерируемых символов:");
22              string buf = Console.ReadLine();
23              int n = int.Parse(buf); // Число генерируемых символов
24
25              string src = ""; // Сообщение на выходе источника (ИС)
26              string enc = ""; // Сообщение на выходе кодера источника (КИС)
27              string dec = ""; // Сообщение на выходе декодера источника (ДИС)
28              buf = "|{0,5}|{1,7}|{2,7}|{3,7}|"; // Строка форматного вывода
29
30              Console.WriteLine(buf, "№", "ИС", "КИС", "ДИС");
31              for (int i = 0; i < n; i++)
32              {
33                  src = r.GenerState();
34                  enc = sEnc.SimpleEncoding(src);
35                  dec = sDec.SimpleDecoding(enc);
36                  Console.WriteLine(buf, i + 1, src, enc, dec);
37              }
38
39              Console.WriteLine(r.GetFreqs());
40              Console.WriteLine(sDec.GetErrorNumber());
41              Console.Read();
42
43          }
44      }
```

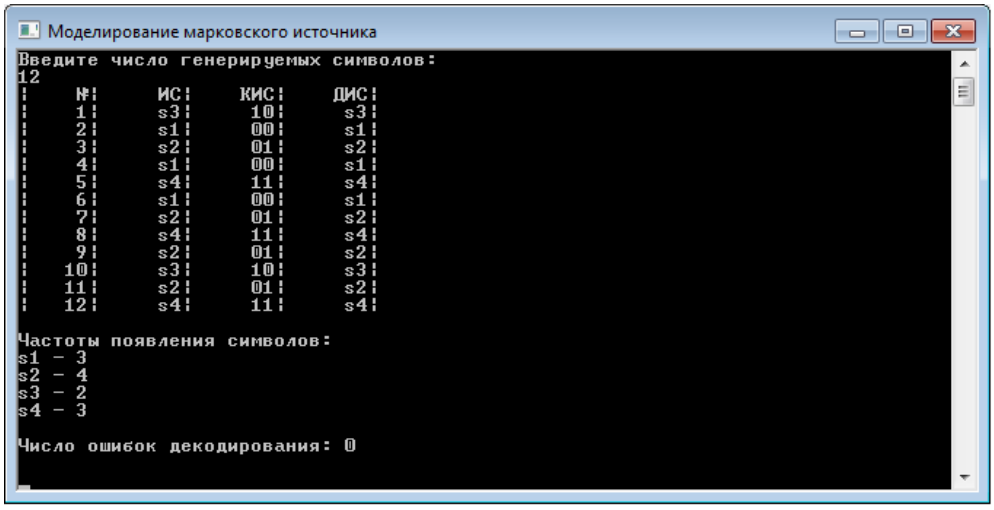


Рис. 2.4. Результат работы консольного приложения

## 3. Моделирование и расчет характеристик дискретных каналов связи

### 3.1. Дискретные каналы связи. Информационные потери при передаче сообщений через канал с помехами

***Дискретные каналы связи.***

***Каналом связи*** (КС) называют совокупность средств, предназначенных для передачи сообщений через определенную физическую среду в пространстве и во времени (рис. 3.1). В частном случае в качестве физической среды может выступать носитель информации.

Каналы связи могут быть описаны в терминах множества входных сигналов $X$ и выходных сигналов $Y$.

Канал связи называется ***дискретным***, если множества $X$ и $Y$ являются конечными, и называется ***непрерывным***, если $X$ и $Y$ – непрерывные множества. Если одно из множеств $X$ и $Y$ является непрерывным, а другое – дискретным, то такой канал называется ***непрерывно-дискретным*** (дискретно-непрерывным).

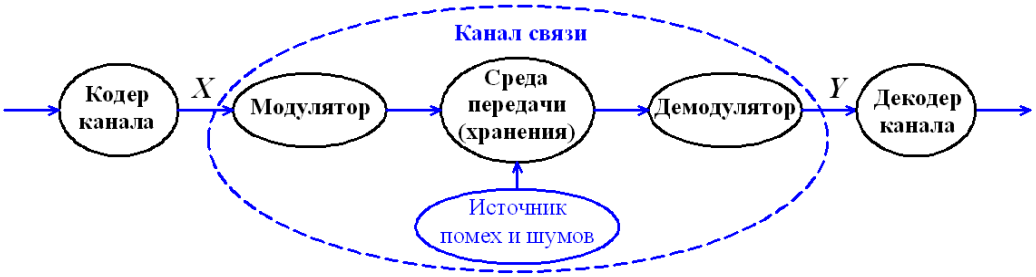


Рис. 3.1. Схема канала связи

В случае дискретного канала множество $X = \{x_i\}$ называют ***входным алфавитом***, а множество $Y = \{y_j\}$ – ***выходным алфавитом***.

Наличие помех и шумов в дискретном канале связи приводит к тому, что определенный входной символ $x_i$ с вероятностями, зависящими от самого канала, может переходить в различные выходные символы $y_j$.

Дискретный канал связи называется ***каналом без памяти***, если вероятности перехода входных символов в выходные символы не зависят от ранее переданных символов.

Отобразить влияние помех и шумов на передачу сообщений через дискретный канал без памяти можно с помощью графа перехода состояний (рис. 3.2).

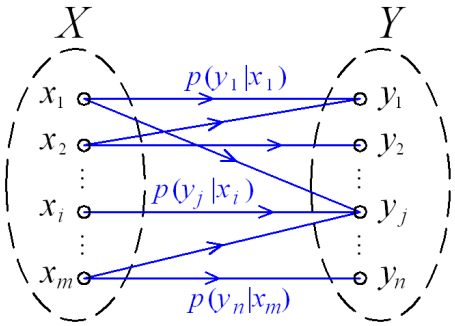


Рис. 3.2. Граф перехода состояний дискретного канала связи без памяти

Для описания дискретных каналов связи без памяти используются матрицы переходных вероятностей, которые имеют следующий вид:

$$P_{mn} = \begin{bmatrix} p(y_1 \mid x_1) & p(y_1 \mid x_2) & \dots & p(y_1 \mid x_n) \\ p(y_2 \mid x_1) & p(y_2 \mid x_2) & \dots & p(y_2 \mid x_n) \\ \dots & \dots & \dots & \dots \\ p(y_n \mid x_1) & p(y_n \mid x_2) & \dots & p(y_n \mid x_n) \end{bmatrix};$$

где $p\ (y_j \mid x_i)$ – вероятность перехода символа $x_i$ входного алфавита в символ $y_j$ выходного алфавита ($i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$).

Среди простых моделей дискретных каналов связи без памяти наибольшее практическое значение имеют модели двоичных каналов (рис. 3.3).

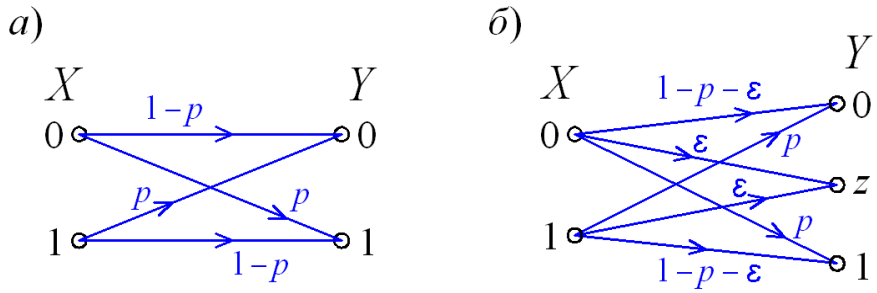


Рис. 3.3. Графы перехода состояний двоичных каналов связи: *а*) – двоичный симметричный канал; *б*) – двоичный канал со стиранием

Матрица переходных вероятностей двоичного симметричного канала связи записывается следующим образом:

$$P = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix};$$

где $p$ – вероятность ошибки при передаче одного символа.

Матрица переходных вероятностей двоичного канала связи со стиранием имеет следующий вид:

$$P = \begin{bmatrix} 1-p-\varepsilon & \varepsilon & p \\ p & \varepsilon & 1-p-\varepsilon \end{bmatrix};$$

где $\varepsilon$ – вероятность стирания символа в канале связи.

❒ ***Пример* 3.1. *Моделирование двоичного канала связи со стиранием*.**

Требуется разработать компьютерную модель двоичного канала связи со стиранием, для которого вероятность ошибки при передаче одного символа составляет $p$ = 0,02, а вероятность стирания символа – $\varepsilon$ = 0,02. Полученную модель будем использовать в составе системы, полученной в примере 2.4.

Для реализации модели двоичного канала связи создадим в консольном приложении класс **BinaryChannel**. В данный класс добавим метод передачи последовательности двоичных символов **string TransmitBits(string cx)**, который в качестве параметра принимает строку **cx** (входная двоичная последовательность) и возвращает строку, учитывающую влияние помех в канале связи.

Исходный код класса **BinaryChannel** приведен в листингах 3.1 и 3.2.

Листинг 3.1. Исходный код класса **BinaryChannel** (часть 1)

```
/// <summary>
/// Представляет двоичный канал связи со стиранием (ДКС)
/// </summary>
class BinaryChannel
{
    const int m = 2; // Объём входного алфавита ДКС
    const int n = 3; // Объём выходного алфавита ДКС
    string[] x; // Входной алфавит ДКС
    string[] y; // Выходной алфавит ДКС
    double p; // Вероятность ошибки при передаче символа через ДКС
    double e; // Вероятность стирания символа при передаче через ДКС
    double[,] pM; // Матрица переходных вероятностей ДКС
    int errNum; // Число ошибок при передаче через ДКС
    Random rnd;

    /// <summary>
    /// Конструктор с параметрами по умолчанию
    /// </summary>
    public BinaryChannel()
    {
        x = new string[m] { "0", "1" };
        y = new string[n] { "0", " ", "1" };
        p = 0.02;
        e = 0.01;
        pM = new double[m, n]{{1 - p - e, e, p        },
                              {p,         e, 1 - p - e}};
        errNum = 0;
        rnd = new Random();
    }
```

Листинг 3.1. Исходный код класса **BinaryChannel** (часть 2)

```
55          /// <summary>
56          /// Возвращает переданную двоичную последовательность
57          /// </summary>
58          /// <param name="cx">Входная последовательность</param>
59          /// <returns>Выходная последовательность</returns>
60          public string TransmitBits(string cx)
61          {
62              string cy = ""; // Выходная последовательность
63              double r = rnd.NextDouble();
64              double q = 0.0; // Кумулятивная вероятность
65              int i = 0; // Индекс символа входного алфавита
66
67              for (int k = 0; k < cx.Length; k++)
68              {
69                  i = Convert.ToInt32(cx.Substring(k, 1));
70                  for (int j = 0; j < n; j++)
71                  {
72                      if ((r > q) && (r <= q + pM[i, j]))
73                      {
74                          cy += y[j];
75                          if (x[i] != y[j]) errNum++; // Подсчет ошибок
76                          break;
77                      }
78                      q += pM[i, j];
79                  }
80              }
81              return cy;
82          }
83
84          /// <summary>
85          /// Возвращает строку с данными об ошибках при передаче
86          /// </summary>
87          /// <returns>Данные об ошибках при передаче</returns>
88          public string GetErrorNumber()
89          {
90              return string.Format("Число ошибок при передаче: " + errNum);
91          }
92      }
```

В код метода **Main()** из примера 2.4 следует добавить создание экземпляра класса **BinaryChannel**, а также вызов его метода **TramsmitBits()**. Кроме того, в строке форматного вывода должно присутствовать поле подстановки для сообщения на выходе канала связи.

Результат работы консольного приложения приведен на рис. 3.4, из которого видно, что в сообщениях 13 и 17 произошли ошибки, а в сообщении 20 был потерян символ. ❒

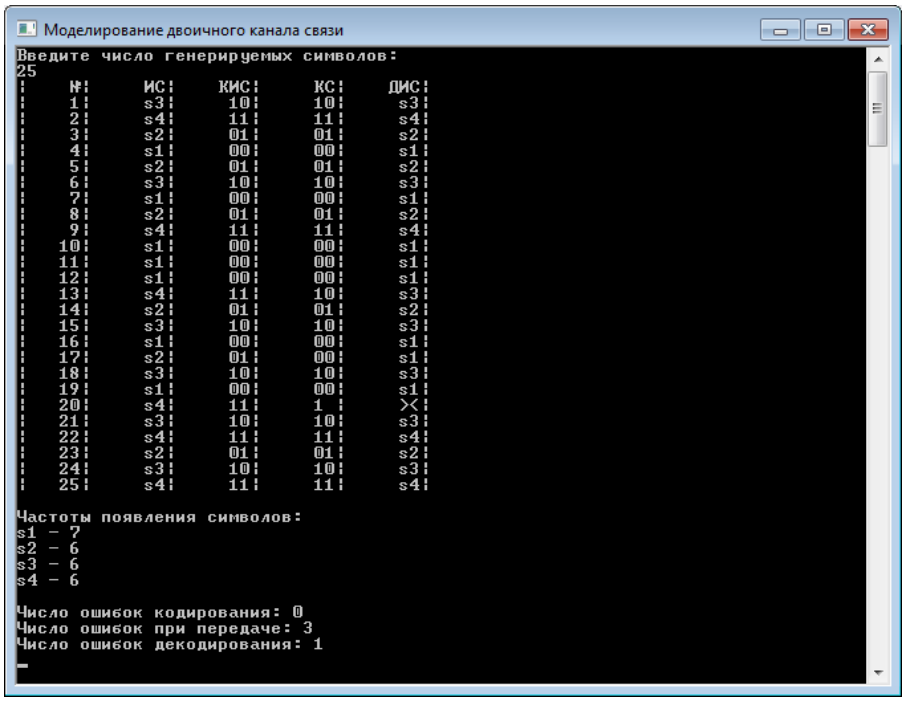


Рис. 3.4. Результат работы консольного приложения

***Информационные потери при передаче сообщений через канал связи с помехами.***

В канале связи без помех количество информации, получаемой с выхода канала связи, совпадает с количеством информации, подаваемой на его вход. То есть в этом случае передача осуществляется без потерь.

При наличии помех в канале связи нарушается соответствие между средним количеством информации, получаемой с выхода канала в единицу времени, и количеством информации на его входе за тот же период времени.

Количество переданной через канал связи информации определяется взаимной информацией:

$$I(X,Y) = H(X) - H(X|Y). \qquad (3.1)$$

Схема потоков информации через канал связи показана на рис. 3.5, где энтропия $H(X|Y)$ характеризует неопределенность состояния источника $X$ при известном состоянии источника $Y$. Данная величина определяет информационные потери при передаче одного символа через канал связи.

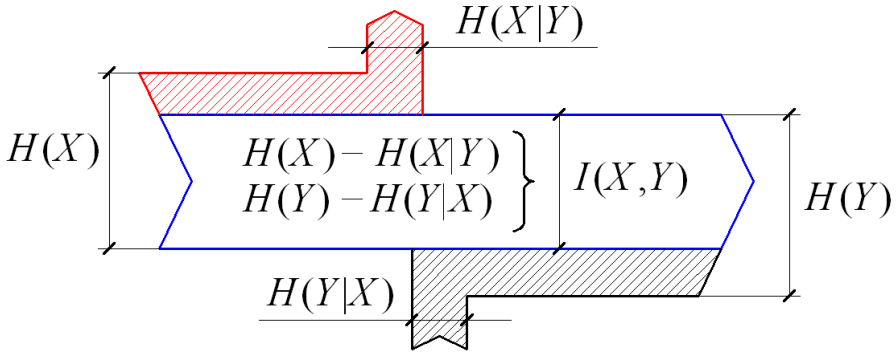


Рис. 3.5. Схема информационного потока через канал связи с помехами

Если канал полностью зашумлен, то $H(X|Y) = H(X)$ и никакой передачи информации не происходит (вся передаваемая информация теряется), то есть $I(X,Y) = 0$.

Информационные потери при передаче $L$ символов по данному каналу связи можно найти из формулы:

$$\Delta I = L \cdot H(X|Y); \qquad (3.2)$$

где $H(X|Y)$ – условная энтропия входного алфавита $X$ канала связи относительно выходного алфавита $Y$.

Отсюда среднее количество принятой информации может быть определено следующим образом:

$$I = L \cdot H(Y) - \Delta I = L[H(Y) - H(X|Y)]. \qquad (3.3)$$

❒ ***Пример* 3.2. *Расчет информационных потерь при передаче информации по каналу связи с помехами.***

Задан дискретный канал связи без памяти, в котором присутствуют помехи. Требуется найти информационные потери при передаче 500 символов через канал, а также количество принятой информации.

Входным и выходным алфавитами канала связи являются множества: $X = \{x_1, x_2, x_3, x_4\}$, $Y = \{y_1, y_2, y_3, y_4\}$. Вероятности появления символов входного алфавита $X$ равны:

$$p(x_1) = 0{,}37; \quad p(x_2) = 0{,}3; \quad p(x_3) = 0{,}23; \quad p(x_4) = 0{,}1.$$

Матрица переходных вероятностей $p(y_j|x_i)$ канала связи имеет следующий вид:

$$P(Y \mid X) = \begin{bmatrix} 0{,}97 & 0{,}02 & 0{,}01 & 0 \\ 0{,}02 & 0{,}98 & 0 & 0 \\ 0 & 0{,}02 & 0{,}96 & 0{,}02 \\ 0 & 0 & 0{,}01 & 0{,}99 \end{bmatrix}.$$

Информационные потери можно найти по формуле (3.2), в которой условная энтропия $H(X|Y)$ может быть определена следующим образом:

$$H(X|Y) = H(X,Y) - H(Y);$$

где $H(X,Y)$ – энтропия объединения входного $X$ и выходного $Y$ алфавитов; $H(Y)$ – энтропия выходного алфавита $Y$.

Определим энтропию $H(Y)$ выходного алфавита $Y$:

$$H(Y) = -\sum_{j=1}^{n} p(y_j) \log_2 p(y_j);$$

где $p(y_j) = \sum_{i=1}^{m} p(x_i) p(y_j \mid x_i)$; причем $\sum_{j=1}^{n} p(y_j) = 1$.

По матрице переходных вероятностей получим:

$p(y_1) = 0{,}37 \cdot 0{,}97 + 0{,}3 \cdot 0{,}02 = 0{,}3649;$

$p(y_2) = 0{,}37 \cdot 0{,}02 + 0{,}3 \cdot 0{,}98 + 0{,}23 \cdot 0{,}02 = 0{,}306;$

$p(y_3) = 0{,}37 \cdot 0{,}01 + 0{,}23 \cdot 0{,}96 + 0{,}1 \cdot 0{,}01 = 0{,}2255;$

$p(y_4) = 0{,}23 \cdot 0{,}02 + 0{,}1 \cdot 0{,}99 = 0{,}1036;$

$$0{,}3649 + 0{,}306 + 0{,}2255 + 0{,}1036 = 1.$$

Отсюда энтропия выходного алфавита будет:

$H(Y) = -(0{,}3649 \log_2 0{,}3649 + 0{,}306 \log_2 0{,}306 +$

$+ 0{,}2255 \log_2 0{,}2255 + 0{,}1036 \log_2 0{,}1036 = 0{,}5306 + 0{,}5227 +$

$+ 0{,}4842 + 0{,}3377 = 1{,}874$ (бит/символ).

Энтропию объединения $H(X,Y)$ можно определить по формуле:

$$H(X,Y) = -\sum_{i=1}^{m} \sum_{j=1}^{n} p(x_i, y_j) \log_2 p(x_i, y_j).$$

Для нахождения энтропии объединения $H(X,Y)$ получим матрицу совместных вероятностей $P(X,Y)$, определив для этого все совместные вероятности:

$$p(x_i,y_j) = p(x_i)\cdot p(y_j|x_i);$$

$p(x_1,y_1) = 0{,}37\cdot 0{,}97 = 0{,}3589$; $p(x_1,y_2) = 0{,}37\cdot 0{,}02 = 0{,}0074$;

$p(x_1,y_3) = 0{,}37\cdot 0{,}01 = 0{,}0037$; $p(x_1,y_4) = 0{,}37\cdot 0 = 0$;

$p(x_2,y_1) = 0{,}3\cdot 0{,}02 = 0{,}006$; $p(x_2,y_2) = 0{,}3\cdot 0{,}98 = 0{,}249$;

$p(x_2,y_3) = 0{,}3\cdot 0 = 0$; $p(x_2,y_4) = 0{,}3\cdot 0 = 0$;

$p(x_3,y_1) = 0{,}23\cdot 0 = 0$; $p(x_3,y_2) = 0{,}23\cdot 0{,}02 = 0{,}0046$;

$p(x_3,y_3) = 0{,}23\cdot 0{,}96 = 0{,}2208$; $p(x_3,y_4) = 0{,}23\cdot 0{,}02 = 0{,}0046$;

$p(x_4,y_1) = 0{,}1\cdot 0 = 0$; $p(x_4,y_2) = 0{,}1\cdot 0 = 0$;

$p(x_4,y_3) = 0{,}1\cdot 0{,}01 = 0{,}001$; $p(x_4,y_4) = 0{,}1\cdot 0{,}99 = 0{,}099$.

Матрица совместных вероятностей будет иметь следующий вид:

$$P(X,Y) = \begin{bmatrix} 0{,}3589 & 0{,}0074 & 0{,}0037 & 0 \\ 0{,}006 & 0{,}249 & 0 & 0 \\ 0 & 0{,}0046 & 0{,}2208 & 0{,}0046 \\ 0 & 0 & 0{,}001 & 0{,}099 \end{bmatrix}.$$

Отсюда энтропия объединения будет:

$H(X,Y) = -(0{,}3589\cdot\log_2 0{,}3589 + 0{,}0074\cdot\log_2 0{,}0074 +$

$+ 0{,}0037\cdot\log_2 0{,}0037 + 0{,}006\cdot\log_2 0{,}006 + 0{,}249\cdot\log_2 0{,}249 +$

$+ 0{,}0046\cdot\log_2 0{,}0046 + 0{,}2208\cdot\log_2 0{,}2208 + 0{,}0046\cdot\log_2 0{,}0046 +$

$+ 0{,}001\cdot\log_2 0{,}001 + 0{,}099\cdot\log_2 0{,}099) = 2{,}069$ (бит/два симв.).

Найдем условную энтропию входного алфавита относительно выходного:

$H(X|Y) = 2{,}069 - 1{,}874 = 0{,}196$ (бит/символ).

Таким образом, информационные потери при передаче заданного числа символов через канал связи будут:

$\Delta I = 500\cdot 0{,}196 = 97{,}76$ (бит).

Отсюда количество принятой информации по формуле (3.3) будет:

$I = 500 \cdot 1{,}87 - 97{,}76 = 839{,}11$ (бит). ❐

## 3.2. Пропускная способность дискретного канала связи. Кодеры и декодеры канала связи

***Пропускная способность дискретного канала связи.***

Для описания возможностей канала по передаче информации используют понятие скорости передачи (технической и информационной).

Под ***технической скоростью передачи*** $V_T$ (скорость манипуляции) понимают число элементарных сигналов (символов), передаваемых по каналу в единицу времени. Техническая скорость зависит от свойств линии связи и быстродействия аппаратуры канала и определяется следующим образом:

$$V_T = 1 / \tau_{ср}, \text{ (бод)} \qquad (3.4)$$

где $\tau_{ср}$ – среднее время передачи одного символа через канал связи.

Под ***информационной скоростью*** $V$ (скоростью передачи информации) понимают среднее количество информации, которое передается по каналу связи в единицу времени.

При известной технической скорости $V_T$ скорость передачи информации по каналу связи определяется соотношением:

$$V = V_T \cdot I(X,Y), \text{ (бит/с)} \qquad (3.5)$$

где $I(X,Y)$ – средняя взаимная информация, переносимая одним символом.

***Пропускной способностью*** $C$ дискретного канала связи называется максимальное значение скорости передачи информации по этому каналу:

$$C = \max_{\bar{p}(x)} \{V\} = \max_{\bar{p}(x)} \{V_T \cdot I(X,Y)\}, \text{ (бит/с)} \qquad (3.6)$$

Пропускная способность канала связи при отсутствии помех будет:

$$C = V_T \cdot \log_2 m. \qquad (3.7)$$

❒ ***Пример* 3.3. *Расчет пропускной способности дискретного канала связи с помехами.***

Требуется определить пропускную способность дискретного канала связи, описанного в примере 3.2. Примем, что каждый символ сообщения передается через данный канал в среднем за $2 \cdot 10^{-4}$ с.

По формуле (3.3) техническая скорость передачи будет:

$$V_T = 1 / 0{,}0002 = 5000. \text{ (бод)}$$

Для нахождения пропускной способности канала связи по формуле (3.6) необходимо определить взаимную информацию $I(X,Y)$ входного $X$ и выходного $Y$ алфавитов:

$$I(X,Y) = H(X) - H(X|Y);$$

$$I(X,Y) = H(Y) - H(Y|X).$$

В примере 3.2. было получено, что $H(Y) = 1{,}874$ бит/символ и $H(X|Y) = 0{,}196$ бит/символ.

Энтропия $H(X)$ входного алфавита $X$ может быть найдена следующим образом:

$$H(X) = -\sum_{i=1}^{m} p(x_i) \log_2 p(x_i);$$

$H(X) = -(0{,}37 \cdot \log_2 0{,}37 + 0{,}3 \cdot \log_2 0{,}3 + 0{,}23 \cdot \log_2 0{,}23 +$

$+ 0{,}1 \cdot \log_2 0{,}1) = 0{,}531 + 0{,}521 + 0{,}488 + 0{,}332 =$

$= 1{,}872$ (бит/символ).

Определим условную энтропию $H(Y|X)$ выходного алфавита $Y$ относительно входного $X$ по формуле:

$$H(Y \mid X) = -\sum_{i=1}^{m} \sum_{j=1}^{n} p(x_i) p(y_j \mid x_i) \log_2 p(y_j \mid x_i);$$

$H(Y|X) = -[0{,}37 \cdot (0{,}97 \cdot \log_2 0{,}97 + 0{,}02 \cdot \log_2 0{,}02 +$

$+ 0{,}01 \cdot \log_2 0{,}01) + 0{,}3 \cdot (0{,}02 \cdot \log_2 0{,}02 + 0{,}98 \cdot \log_2 0{,}98) +$

$+ 0{,}23 \cdot (0{,}02 \cdot \log_2 0{,}02 + 0{,}96 \cdot \log_2 0{,}96) + 0{,}01 \cdot (0{,}02 \cdot \log_2 0{,}02 +$

$+ 0{,}96 \cdot \log_2 0{,}96) + 0{,}1 \cdot (0{,}01 \cdot \log_2 0{,}01 + 0{,}99 \cdot \log_2 0{,}99) =$

$= 0{,}37(0{,}043 + 0{,}113 + 0{,}066) + 0{,}3\ (0{,}113 + 0{,}029) +$

+ 0,23 (0,113 + 0,057) + 0,01 (0,066 + 0,014) = 0,082 +

+ 0,042 + 0,065 + 0,008 = 0,198 (бит/символ).

Отсюда получим взаимную информацию:

$I(X,Y)$ = 1,872 – 0,196 = 1,676 (бит/символ);

$I(X,Y)$ = 1,874 – 0,198 = 1,676 (бит/символ).

Таким образом, пропускная способность канала связи будет:
$C = 5000 \cdot 1{,}676 = 8381$ (бит/с). ❒

***Кодер и декодер канала связи. Код с битом четности.***

При наличии помех в канале связи обязательными частями системы передачи сообщений являются кодер и декодер канала связи.

***Кодер канала связи*** служит для представления сообщений в форме, обеспечивающей их защиту от помех при передаче по каналу либо возможных искажений при хранении информации.

Для защиты информации от помех кодер канала преобразует сообщения в ***помехоустойчивый код***, в котором к символам, несущим полезную информацию (***информационные символы***), добавлены дополнительные символы (***проверочные символы***). Значения проверочных символов помехоустойчивого кода определяются путем проведения математических операций над информационными символами.

***Декодер канала связи*** обнаруживает и исправляет ошибки в сообщениях на выходе канала, используя свойства помехоустойчивого кода. Обнаружение ошибки происходит, если декодер канала сигнализирует об отличии принятого кодового слова от переданного. Ошибка может быть исправлена, если декодер канала указывает позицию и правильное значение искаженного символа кодового слова.

Простейшим примером помехоустойчивого кода является двоичный код с битом чётности (с проверкой на чётность).

***Код с битом чётности*** – код, в котором к информационным символам, добавляется символ, принимающий значение 0 или 1 так, чтобы общее число единиц в кодовом слове было чётным. Например, к кодовому слову 10101101 следует добавить 1,

чтобы получить 110101101. Если общее число единиц в полученном кодовом слове нечётное, то в слове присутствует ошибка.

Код с битом чётности позволяет обнаружить наличие ошибки, но не позволяет исправить, поскольку неизвестно в каком знаке она произошла. Кроме того, может произойти серия ошибок, которая переведет одну разрешенную комбинацию в другую. Например, вместо 110101101 будет получено 110011101. В этом случае код с битом чётности ошибку не обнаружит.

❒ ***Пример* 3.4. *Моделирование системы передачи сообщений.***

Требуется разработать модель системы передачи сообщений, включающей такие элементы, как источник, кодер и декодер источника, канал, кодер и декодер канала.

Для решения этой задачи в приложение, полученное в примере 3.1, добавим классы **ChannelEncoder** и **ChannelDecoder**, соответствующие кодеру и декодеру канала связи. Примем, что кодер и декодер канала связи используют в своей работе код с битом чётности.

В качестве канала связи будет использовать двоичный симметричный канал. Для этого в модели канала, полученной в примере 3.1, зададим вероятность стирания символа $\varepsilon = 0$.

Структура моделируемой системы с указанием имен классов и передаваемых строковых переменных представлена на рис. 3.6.

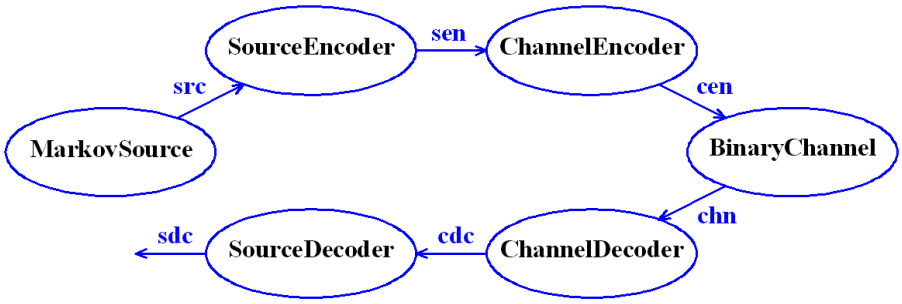


Рис. 3.6. Элементы моделируемой системы передачи сообщений

Для реализации помехоустойчивого кодирования в класс **ChannelEncoder** добавим метод **string ParityBitEncoding(string**

**cx),** который принимает значение входной двоичной последовательности **cx**, а возвращает кодовое слово с битом четности.

Для декодирования в классе **ChannelDecoder** будет присутствовать метод **string ParityBitEncoding(string cx)**, в котором строковый параметр **cx** соответствует кодовому слову с битом четности. Данный метод должен возвращать двоичную последовательность без бита четности.

Исходный код классов **ChannelEncoder** и **ChannelDecoder** приведен в листингах 3.2 и 3.3.

Листинг 3.2. Исходный код класса **ChannelEncoder**

```
9   /// <summary>
10  /// Представлет кодер канала связи (ККС)
11  /// </summary>
12  class ChannelEncoder
13  {
14      /// <summary>
15      /// Выполняет кодирование с битом чётности
16      /// </summary>
17      /// <param name="cx">Входная последовательность</param>
18      /// <returns>Кодовое слово с битом чётности</returns>
19      public string ParityBitEncoding(string cx)
20      {
21          int n = cx.Length; // Длина входной последовательности
22          int u = 0; // Число единиц во входной последовательности
23          for (int i = 0; i < n; i++)
24          {
25              // Проверить, является ли символ единицей
26              if (cx.Substring(i, 1) == "1") u++;
27          }
28          // Проверить, является ли число единиц четным,
29          if (u % 2 == 0) cx += "0"; // добавить бит чётности
30          else cx += "1";
31
32          return cx;
33      }
34  }
```

Листинг 3.3. Исходный код класса **ChannelDecoder**

```
/// <summary>
/// Представляет декодер канала связи
/// </summary>
class ChannelDecoder
{
    int errNum; // Число ошибок при декодировании

    /// <summary>
    /// Конструктор по умолчанию
    /// </summary>
    public ChannelDecoder()
    { this.errNum = 0; }

    /// <summary>
    /// Декодирует кодовые слова с битом четности
    /// </summary>
    /// <param name="cx">Входная последовательность</param>
    /// <returns>Выходная последовательность</returns>
    public string ParityBitDecoding(string cx)
    {
        int n = cx.Length; // Длина входной последовательности
        int u = 0; // Число единиц во входной последовательности
        for (int i = 0; i < n; i++)
        {
            if (cx.Substring(i, 1) == "1") u++;
        }
        // Проверить, является ли число единиц четным
        if (u % 2 == 0)
        {
            cx = cx.Remove(n - 1, 1); // Удалить бит чётности
        }
        else
        {
            errNum++;
            cx = "111"; // Ошибочная последовательность
        }

        return cx;
    }

    /// <summary>
    /// Возвращает строку с данными об ошибках декодирования
    /// </summary>
    /// <returns>Данные об ошибках</returns>
    public string GetErrorNumber()
    {
        return string.Format("Число обнаруженных ошибок: " + errNum);
    }
}
```

Исходный код класса **Program** консольного приложения представлен в листинге 3.4.

Листинг 3.4. Исходный код класса **Program** консольного приложения

```
class Program
{
    static void Main(string[] args)
    {
        Console.Title = "Моделирование системы передачи сообщений";
        // Экземпляр класса марковский источник сообщений (ИС)
        MarkovSource mSrc = new MarkovSource();
        // Экземпляр класса кодер источника сообщений (КИС)
        SourceEncoder sEnc = new SourceEncoder();
        // Экземпляр класса кодер канала связи (ККС)
        ChannelEncoder cEnc = new ChannelEncoder();
        // Экземпляр класса двоичный канал связи (КС)
        BinaryChannel bChn = new BinaryChannel();
        // Экземпляр класса декодер канала связи (ДКС)
        ChannelDecoder cDec = new ChannelDecoder();
        // Экземпляр класса декодер источника сообщений (ДИС)
        SourceDecoder sDec = new SourceDecoder();

        Console.WriteLine("Введите число генерируемых символов:");
        string buf = Console.ReadLine();
        int n = int.Parse(buf); // Число генерируемых символов

        string src = ""; // Сообщение на выходе ИС
        string sen = ""; // Сообщение на выходе КИС
        string cen = ""; // Сообщение на выходе ККС
        string chn = ""; // Сообщение на выходе КС
        string cdc = ""; // Сообщение на выходе ДКС
        string sdc = ""; // Сообщение на выходе ДИС
        // Строка форматного вывода
        buf = "|{0,5}|{1,7}|{2,7}|{3,7}|{4,7}|{5,7}|{6,7}|";

        Console.WriteLine("\nСообщения на выходе элементов системы:");
        Console.WriteLine(buf, "№", "ИС", "КИС", "ККС", "КС", "ДКС", "ДИС");
        for (int i = 0; i < n; i++)
        {
            src = mSrc.GenerSimbol();
            sen = sEnc.SimpleEncoding(src);
            cen = cEnc.ParityBitEncoding(sen);
            chn = bChn.TransmitBits(cen);
            cdc = cDec.ParityBitDecoding(chn);
            sdc = sDec.SimpleDecoding(cdc);
            Console.WriteLine(buf, i + 1, src, sen, cen, chn, cdc, sdc);
        }

        Console.WriteLine(mSrc.GetFreqs());
        Console.WriteLine(bChn.GetErrorNumber());
        Console.WriteLine(cDec.GetErrorNumber());
        Console.WriteLine(sDec.GetErrorNumber());
        Console.Read();
    }
}
```

Исходный код класса **MarkovSource** аналогичен коду класса **MarkovChain**, который представлен в листинге 1.1 (метод **GenerSymbol** аналогичен методу **GenerState**).

Результат работы консольного приложения для введенного числа символов показан на рис. 3.6.

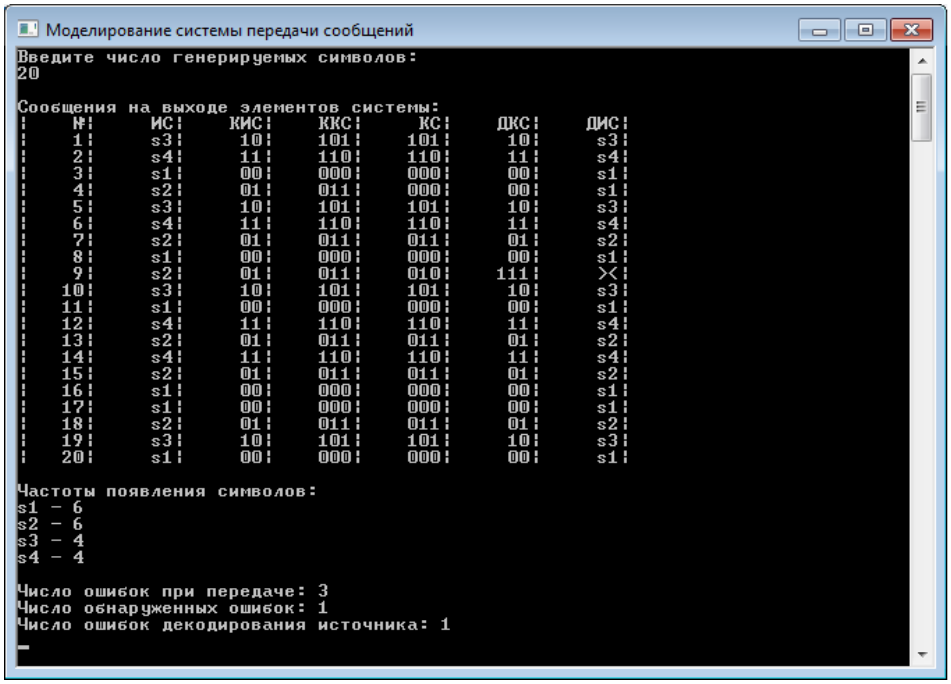


Рис. 3.6. Результат работы консольного приложения

Из результата видно, что декодер канала связи обнаружил одиночную ошибку в сообщении 9, но пропустил двойную ошибку в сообщении 4. ❐

## III. ФОРМУЛИРОВКА ЗАДАНИЯ И ВАРИАНТЫ

***Задание на курсовую работу.***

Данная курсовая работа предполагает выполнение следующих заданий:

1. Разработать компьютерную модель марковского источника сообщения с алфавитом $S = \{s_1, s_2, \dots, s_8\}$. Матрица переходных вероятностей, описывающая данный источник, приведена в табл. 1. С помощью модели произвести вычислительный эксперимент с целью нахождения стационарных вероятностей. Для этого в приложении требуется задать достаточно большую длину сообщения (например, несколько десятков тысяч символов).

2. Определить энтропию марковского источника сообщений $S$ на основе данных из табл. 2 (переходные вероятности) и результатов моделирования (стационарные вероятности состояний). Также определить эффективность $E$, частные избыточности, вызванные наличием связи между символами $R_p$ и неэкстремальным распределением символов $R_\varphi$. На основе частных избыточностей найти общую информационную избыточность $R$ источника сообщений.

3. Разработать компьютерную модель источника сообщений $U$, зависимого от марковского источника $S$. Статистическая связь между состояниями источников задана матрицей переходных вероятностей (таблица 2). Осуществить моделирование работы источника при достаточно большой длине сообщения (например, 50000 символов). На основе полученных частот появления символов определить их эмпирические вероятности. Найти энтропию источника $U$, его эффективность и избыточность.

4. Определить условную энтропию $H(U|S)$ источника $U$, состояния которого статистически связаны с состояниями марковского источника $S$. Кроме того, требуется найти энтропию объединения $H(U, S)$ двух источников $U$ и $S$, а также их взаимную информацию $H(U, S)$.

5. Разработать модели кодера и декодера источника сообщений. В качестве кода использовать равномерный двоичный код с трехразрядными кодовыми словами.

6. Разработать модель двоичного канала связи со стиранием (с параметрами: $p = 0{,}02$, $\varepsilon = 0.01$). Добавить полученную модель

к системе из задания 5. Для построенной модели системы передачи сообщений определить число ошибок в полученном сообщении при разных значениях вероятностей $p$ и $\varepsilon$.

7. На основе заданного распределения вероятностей символов входного алфавита $X$ (таблица 3) и матрицы переходных вероятностей $p(y|x)$ канала связи (таблица 4) найти информационные потери при передаче 500 символов, а также количество принятой информации.

8. Используя описание дискретного канала связи без памяти (таблицы 3 и 4) определить его пропускную способность $C$. Среднее время передачи одного символа принять $\tau = 1 \cdot 10^{-4}$ с.

9. Разработать модели кодера и декодера двоичного канала связи. В качестве помехоустойчивого кода принять код с битом четности. Добавить полученные модели элементов к системе из задания 5. Промоделировать полную систему передачи сообщений и рассмотреть влияние ошибок в канале связи на работу декодеров.

***Индивидуальные варианты заданий.***

Таблица 1.

Матрицы переходных вероятностей $\boldsymbol{p(s_j \mid s_i)}$ цепи Маркова

| **№ вар.** | **Матрица переходных вероятностей $p(s_j\|s_i)$** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $s_i \backslash s_j$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ | $s_7$ | $s_8$ |
| **1, 9, 17** | $s_1$ | 0,14 | 0,2 | 0,02 | 0,27 | 0,09 | 0,02 | 0,1 | 0,16 |
| | $s_2$ | 0,2 | 0,05 | 0,13 | 0,08 | 0,2 | 0,13 | 0,2 | 0,01 |
| | $s_3$ | 0,06 | 0,2 | 0,16 | 0,22 | 0,12 | 0,09 | 0,05 | 0,1 |
| | $s_4$ | 0,07 | 0,18 | 0,2 | 0,04 | 0,2 | 0,09 | 0,1 | 0,12 |
| | $s_5$ | 0,09 | 0,3 | 0,08 | 0,2 | 0,01 | 0,1 | 0,07 | 0,15 |
| | $s_6$ | 0,04 | 0,1 | 0,1 | 0,05 | 0,28 | 0,07 | 0,23 | 0,13 |
| | $s_7$ | 0,15 | 0,06 | 0,07 | 0,2 | 0,2 | 0,08 | 0,2 | 0,04 |
| | $s_8$ | 0,2 | 0,03 | 0,16 | 0,1 | 0,09 | 0,1 | 0,07 | 0,25 |
| **2, 10, 18** | $s_1$ | 0,07 | 0,1 | 0,2 | 0,08 | 0,13 | 0,08 | 0,23 | 0,11 |
| | $s_2$ | 0,13 | 0,05 | 0,06 | 0,2 | 0,19 | 0,1 | 0,03 | 0,24 |
| | $s_3$ | 0,1 | 0,04 | 0,1 | 0,15 | 0,2 | 0,09 | 0,07 | 0,25 |
| | $s_4$ | 0,07 | 0,18 | 0,2 | 0,04 | 0,2 | 0,08 | 0,1 | 0,13 |
| | $s_5$ | 0,2 | 0,06 | 0,1 | 0,08 | 0,09 | 0,1 | 0,28 | 0,09 |
| | $s_6$ | 0,06 | 0,2 | 0,04 | 0,17 | 0,09 | 0,3 | 0,03 | 0,11 |

| № вар. | Матрица переходных вероятностей $p(s_j\|s_i)$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $s_i \backslash s_j$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ | $s_7$ | $s_8$ |
| | $s_7$ | 0,05 | 0,08 | 0,1 | 0,27 | 0,2 | 0,08 | 0,2 | 0,02 |
| | $s_8$ | 0,09 | 0,25 | 0,14 | 0,1 | 0,06 | 0,21 | 0,1 | 0,05 |
| **3, 11, 19** | $s_1$ | 0,2 | 0,2 | 0,04 | 0,15 | 0,2 | 0,02 | 0,14 | 0,05 |
| | $s_2$ | 0,1 | 0,05 | 0,08 | 0,14 | 0,19 | 0,23 | 0,2 | 0,01 |
| | $s_3$ | 0,04 | 0,1 | 0,27 | 0,24 | 0,15 | 0,1 | 0,01 | 0,09 |
| | $s_4$ | 0,09 | 0,13 | 0,2 | 0,08 | 0,2 | 0,05 | 0,15 | 0,1 |
| | $s_5$ | 0,11 | 0,22 | 0,2 | 0,07 | 0,07 | 0,2 | 0,03 | 0,1 |
| | $s_6$ | 0,05 | 0,06 | 0,09 | 0,09 | 0,2 | 0,1 | 0,2 | 0,21 |
| | $s_7$ | 0,1 | 0,05 | 0,2 | 0,08 | 0,14 | 0,17 | 0,2 | 0,06 |
| | $s_8$ | 0,06 | 0,04 | 0,3 | 0,1 | 0,03 | 0,2 | 0,07 | 0,2 |
| **4, 12, 20** | $s_1$ | 0,08 | 0,1 | 0,07 | 0,02 | 0,3 | 0,1 | 0,13 | 0,2 |
| | $s_2$ | 0,29 | 0,1 | 0,01 | 0,08 | 0,21 | 0,17 | 0,04 | 0,1 |
| | $s_3$ | 0,06 | 0,05 | 0,1 | 0,26 | 0,1 | 0,08 | 0,07 | 0,28 |
| | $s_4$ | 0,14 | 0,09 | 0,3 | 0,02 | 0,2 | 0,09 | 0,1 | 0,06 |
| | $s_5$ | 0,2 | 0,2 | 0,07 | 0,07 | 0,01 | 0,1 | 0,2 | 0,15 |
| | $s_6$ | 0,15 | 0,2 | 0,01 | 0,06 | 0,11 | 0,2 | 0,09 | 0,18 |
| | $s_7$ | 0,02 | 0,1 | 0,28 | 0,15 | 0,09 | 0,2 | 0,1 | 0,06 |
| | $s_8$ | 0,1 | 0,07 | 0,18 | 0,09 | 0,2 | 0,21 | 0,05 | 0,1 |
| **5, 13, 21** | $s_1$ | 0,25 | 0,07 | 0,1 | 0,09 | 0,1 | 0,16 | 0,03 | 0,2 |
| | $s_2$ | 0,04 | 0,2 | 0,08 | 0,2 | 0,2 | 0,07 | 0,06 | 0,15 |
| | $s_3$ | 0,13 | 0,22 | 0,07 | 0,29 | 0,05 | 0,1 | 0,1 | 0,04 |
| | $s_4$ | 0,15 | 0,07 | 0,1 | 0,01 | 0,2 | 0,08 | 0,3 | 0,09 |
| | $s_5$ | 0,12 | 0,1 | 0,09 | 0,2 | 0,04 | 0,2 | 0,18 | 0,07 |
| | $s_6$ | 0,1 | 0,05 | 0,09 | 0,12 | 0,22 | 0,16 | 0,2 | 0,06 |
| | $s_7$ | 0,01 | 0,2 | 0,13 | 0,2 | 0,08 | 0,13 | 0,05 | 0,2 |
| | $s_8$ | 0,16 | 0,1 | 0,02 | 0,09 | 0,27 | 0,02 | 0,2 | 0,14 |
| **6, 14, 22** | $s_1$ | 0,05 | 0,1 | 0,21 | 0,06 | 0,1 | 0,14 | 0,25 | 0,09 |
| | $s_2$ | 0,02 | 0,2 | 0,08 | 0,2 | 0,27 | 0,1 | 0,08 | 0,05 |
| | $s_3$ | 0,11 | 0,03 | 0,3 | 0,09 | 0,17 | 0,04 | 0,2 | 0,06 |
| | $s_4$ | 0,09 | 0,28 | 0,1 | 0,09 | 0,08 | 0,1 | 0,06 | 0,2 |
| | $s_5$ | 0,13 | 0,1 | 0,08 | 0,2 | 0,04 | 0,2 | 0,18 | 0,07 |
| | $s_6$ | 0,25 | 0,07 | 0,09 | 0,2 | 0,15 | 0,1 | 0,04 | 0,1 |
| | $s_7$ | 0,24 | 0,03 | 0,1 | 0,19 | 0,2 | 0,06 | 0,05 | 0,13 |
| | $s_8$ | 0,11 | 0,23 | 0,08 | 0,13 | 0,08 | 0,2 | 0,1 | 0,07 |
| **7, 15, 23** | $s_1$ | 0,2 | 0,07 | 0,2 | 0,03 | 0,1 | 0,3 | 0,04 | 0,06 |
| | $s_2$ | 0,06 | 0,2 | 0,17 | 0,14 | 0,08 | 0,2 | 0,05 | 0,1 |
| | $s_3$ | 0,21 | 0,2 | 0,1 | 0,2 | 0,09 | 0,09 | 0,06 | 0,05 |
| | $s_4$ | 0,1 | 0,03 | 0,2 | 0,07 | 0,07 | 0,28 | 0,15 | 0,1 |
| | $s_5$ | 0,11 | 0,14 | 0,05 | 0,2 | 0,08 | 0,2 | 0,13 | 0,09 |
| | $s_6$ | 0,09 | 0,01 | 0,1 | 0,15 | 0,24 | 0,27 | 0,1 | 0,04 |

| № вар. | Матрица переходных вероятностей $p(s_j\|s_i)$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | $s_i \backslash s_j$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ | $s_7$ | $s_8$ |
| | $s_7$ | 0,01 | 0,2 | 0,23 | 0,19 | 0,14 | 0,08 | 0,05 | 0,1 |
| | $s_8$ | 0,05 | 0,14 | 0,02 | 0,2 | 0,15 | 0,04 | 0,2 | 0,2 |
| **8,** | $s_1$ | 0,1 | 0,05 | 0,21 | 0,2 | 0,09 | 0,18 | 0,07 | 0,1 |
| **16,** | $s_2$ | 0,06 | 0,1 | 0,2 | 0,09 | 0,15 | 0,28 | 0,1 | 0,02 |
| **24** | $s_3$ | 0,18 | 0,09 | 0,2 | 0,11 | 0,06 | 0,01 | 0,2 | 0,15 |
| | $s_4$ | 0,15 | 0,2 | 0,1 | 0,01 | 0,07 | 0,07 | 0,2 | 0,2 |
| | $s_5$ | 0,06 | 0,1 | 0,09 | 0,2 | 0,02 | 0,3 | 0,09 | 0,14 |
| | $s_6$ | 0,28 | 0,07 | 0,08 | 0,1 | 0,26 | 0,1 | 0,05 | 0,06 |
| | $s_7$ | 0,1 | 0,04 | 0,17 | 0,21 | 0,08 | 0,01 | 0,1 | 0,29 |
| | $s_8$ | 0,2 | 0,13 | 0,1 | 0,3 | 0,02 | 0,07 | 0,1 | 0,08 |

Таблица 2.

Матрицы переходных вероятностей $\boldsymbol{p(u_j \mid s_i)}$ дискретного источника сообщений *U* относительно источника *S*

| № вар. | Матрица переходных вероятностей $p(u_j \mid s_i)$ | | | | | | |
|---|---|---|---|---|---|---|---|
| | $s_i \backslash u_j$ | $u_1$ | $u_2$ | $u_3$ | $u_4$ | $u_5$ | $u_6$ |
| **1,** | $s_1$ | 0,23 | 0,19 | 0 | 0,31 | 0,14 | 0,13 |
| **7,** | $s_2$ | 0,16 | 0,25 | 0,13 | 0,09 | 0,22 | 0,15 |
| **13,** | $s_3$ | 0,07 | 0,22 | 0,18 | 0,16 | 0,32 | 0,05 |
| **19** | $s_4$ | 0,21 | 0,34 | 0 | 0,1 | 0,08 | 0,27 |
| | $s_5$ | 0,13 | 0,09 | 0,28 | 0,26 | 0 | 0,24 |
| | $s_6$ | 0,25 | 0,14 | 0,23 | 0,03 | 0,27 | 0,08 |
| | $s_7$ | 0,05 | 0 | 0,09 | 0,35 | 0,26 | 0,25 |
| | $s_8$ | 0,33 | 0,27 | 0,11 | 0,06 | 0,12 | 0,11 |
| **2,** | $s_1$ | 0,31 | 0,12 | 0,05 | 0,27 | 0,11 | 0,14 |
| **8,** | $s_2$ | 0,09 | 0,24 | 0,21 | 0 | 0,29 | 0,17 |
| **14,** | $s_3$ | 0,23 | 0,31 | 0,04 | 0,17 | 0,2 | 0,05 |
| **20** | $s_4$ | 0,24 | 0 | 0,36 | 0,08 | 0,03 | 0,29 |
| | $s_5$ | 0,04 | 0,17 | 0,33 | 0 | 0,16 | 0,3 |
| | $s_6$ | 0,16 | 0,02 | 0,15 | 0,34 | 0,1 | 0,23 |
| | $s_7$ | 0,29 | 0,13 | 0,09 | 0,16 | 0,28 | 0,05 |
| | $s_8$ | 0,15 | 0,29 | 0,18 | 0 | 0,17 | 0,21 |
| **3,** | $s_1$ | 0,18 | 0,22 | 0 | 0,33 | 0,21 | 0,06 |
| **9,** | $s_2$ | 0,14 | 0,25 | 0,09 | 0,17 | 0,13 | 0,22 |
| **15,** | $s_3$ | 0,32 | 0,1 | 0,16 | 0,26 | 0,16 | 0 |
| **21** | $s_4$ | 0,19 | 0,08 | 0,31 | 0,11 | 0,24 | 0,07 |
| | $s_5$ | 0,35 | 0,13 | 0,25 | 0 | 0,16 | 0,11 |

| № вар. | Матрица переходных вероятностей $p(u_j \mid s_i)$ | | | | | | |
|---|---|---|---|---|---|---|---|
| | $s_i \backslash u_j$ | $u_1$ | $u_2$ | $u_3$ | $u_4$ | $u_5$ | $u_6$ |
| | $s_6$ | 0,27 | 0,36 | 0,1 | 0,12 | 0,07 | 0,08 |
| | $s_7$ | 0,13 | 0,12 | 0,31 | 0,05 | 0,15 | 0,24 |
| | $s_8$ | 0,08 | 0,15 | 0,29 | 0,11 | 0 | 0,37 |
| **4, 10, 16, 22** | $s_1$ | 0,05 | 0,14 | 0,28 | 0,16 | 0,27 | 0,1 |
| | $s_2$ | 0,3 | 0,26 | 0 | 0,12 | 0,14 | 0,18 |
| | $s_3$ | 0,23 | 0 | 0,19 | 0,07 | 0,35 | 0,16 |
| | $s_4$ | 0,12 | 0,25 | 0,06 | 0,21 | 0,13 | 0,23 |
| | $s_5$ | 0,08 | 0,31 | 0,26 | 0,1 | 0,16 | 0,09 |
| | $s_6$ | 0,32 | 0,04 | 0,3 | 0,12 | 0 | 0,22 |
| | $s_7$ | 0,17 | 0,14 | 0,16 | 0,23 | 0,24 | 0,06 |
| | $s_8$ | 0,2 | 0 | 0,24 | 0,05 | 0,36 | 0,15 |
| **5, 11, 17, 23** | $s_1$ | 0,13 | 0 | 0,3 | 0,09 | 0,25 | 0,23 |
| | $s_2$ | 0,04 | 0,26 | 0,21 | 0,33 | 0,11 | 0,05 |
| | $s_3$ | 0,35 | 0,18 | 0 | 0,24 | 0,06 | 0,17 |
| | $s_4$ | 0,22 | 0,05 | 0,19 | 0,15 | 0,32 | 0,07 |
| | $s_5$ | 0,16 | 0,23 | 0,15 | 0,28 | 0,13 | 0,05 |
| | $s_6$ | 0,2 | 0 | 0,36 | 0,22 | 0,08 | 0,14 |
| | $s_7$ | 0,06 | 0,17 | 0,25 | 0,18 | 0,24 | 0,1 |
| | $s_8$ | 0,32 | 0,08 | 0,26 | 0,19 | 0,15 | 0 |
| **6, 12, 18, 24** | $s_1$ | 0,24 | 0,09 | 0,15 | 0,27 | 0,06 | 0,19 |
| | $s_2$ | 0 | 0,35 | 0,08 | 0,16 | 0,31 | 0,1 |
| | $s_3$ | 0,23 | 0,14 | 0 | 0,34 | 0,18 | 0,11 |
| | $s_4$ | 0,07 | 0,26 | 0,28 | 0,21 | 0,15 | 0,03 |
| | $s_5$ | 0,16 | 0,18 | 0,22 | 0,17 | 0,2 | 0,07 |
| | $s_6$ | 0,3 | 0,06 | 0,34 | 0 | 0,09 | 0,21 |
| | $s_7$ | 0,19 | 0,24 | 0,16 | 0,04 | 0,25 | 0,12 |
| | $s_8$ | 0,14 | 0,19 | 0 | 0,25 | 0,07 | 0,35 |

Таблица 3.

Вероятности появления символов входного алфавита

| № вар. | Входной алфавит X | | | | № вар. | Входной алфавит X | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
| **1** | 0,23 | 0,27 | 0,39 | 0,11 | **13** | 0,24 | 0,16 | 0,28 | 0,32 |
| **2** | 0,21 | 0,32 | 0,09 | 0,38 | **14** | 0,15 | 0,41 | 0,35 | 0,09 |
| **3** | 0,18 | 0,42 | 0,25 | 0,15 | **15** | 0,25 | 0,35 | 0,09 | 0,31 |
| **4** | 0,08 | 0,26 | 0,32 | 0,34 | **16** | 0,40 | 0,17 | 0,20 | 0,23 |
| **5** | 0,37 | 0,13 | 0,09 | 0,41 | **17** | 0,40 | 0,10 | 0,17 | 0,33 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **6** | 0,16 | 0,31 | 0,24 | 0,29 | **18** | 0,36 | 0,08 | 0,14 | 0,42 |
| **7** | 0,10 | 0,40 | 0,31 | 0,19 | **19** | 0,19 | 0,31 | 0,08 | 0,42 |
| **8** | 0,24 | 0,27 | 0,16 | 0,33 | **20** | 0,34 | 0,18 | 0,26 | 0,22 |
| **9** | 0,15 | 0,25 | 0,42 | 0,18 | **21** | 0,31 | 0,19 | 0,41 | 0,09 |
| **10** | 0,17 | 0,42 | 0,33 | 0,08 | **22** | 0,37 | 0,15 | 0,23 | 0,25 |
| **11** | 0,40 | 0,30 | 0,08 | 0,22 | **23** | 0,22 | 0,08 | 0,29 | 0,41 |
| **12** | 0,26 | 0,31 | 0,24 | 0,19 | **24** | 0,12 | 0,30 | 0,18 | 0,40 |

Таблица 4.

Матрицы переходных вероятностей $p(y_j|x_i)$ канала связи

| **№ вар.** | **X** | **Выходной алфавит Y** | | | | **№ вар.** | **X** | **Выходной алфавит Y** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $y_1$ | $y_2$ | $y_3$ | $y_4$ | | | $y_1$ | $y_2$ | $y_3$ | $y_4$ |
| **1, 9, 17** | $x_1$ | 0,95 | 0,03 | 0,02 | 0 | **5, 13, 20** | $x_1$ | 0,98 | 0,02 | 0 | 0 |
| | $x_2$ | 0,02 | 0,97 | 0,01 | 0 | | $x_2$ | 0,03 | 0,96 | 0,01 | 0 |
| | $x_3$ | 0 | 0 | 0,99 | 0,01 | | $x_3$ | 0 | 0,02 | 0,94 | 0,04 |
| | $x_4$ | 0 | 0,02 | 0,04 | 0,94 | | $x_4$ | 0 | 0,01 | 0,02 | 0,97 |
| **2, 10, 18** | $x_1$ | 0,97 | 0,02 | 0,01 | 0 | **6, 14, 22** | $x_1$ | 0,94 | 0,03 | 0,02 | 0,01 |
| | $x_2$ | 0,02 | 0,98 | 0 | 0 | | $x_2$ | 0,01 | 0,99 | 0 | 0 |
| | $x_3$ | 0 | 0,02 | 0,95 | 0,03 | | $x_3$ | 0 | 0,03 | 0,96 | 0,01 |
| | $x_4$ | 0 | 0 | 0,01 | 0,99 | | $x_4$ | 0 | 0,01 | 0,04 | 0,95 |
| **3, 11, 19** | $x_1$ | 0,96 | 0,03 | 0,01 | 0 | **7, 15, 23** | $x_1$ | 0,99 | 0,01 | 0 | 0 |
| | $x_2$ | 0,03 | 0,95 | 0,02 | 0 | | $x_2$ | 0,02 | 0,94 | 0,03 | 0,01 |
| | $x_3$ | 0 | 0,01 | 0,97 | 0,02 | | $x_3$ | 0 | 0,02 | 0,98 | 0 |
| | $x_4$ | 0 | 0 | 0,02 | 0,98 | | $x_4$ | 0 | 0,01 | 0,03 | 0,96 |
| **4, 12, 20** | $x_1$ | 0,98 | 0 | 0,01 | 0,01 | **8, 16, 24** | $x_1$ | 0,95 | 0,02 | 0 | 0,03 |
| | $x_2$ | 0,02 | 0,95 | 0 | 0,03 | | $x_2$ | 0,01 | 0,98 | 0,01 | 0 |
| | $x_3$ | 0 | 0,03 | 0,96 | 0,01 | | $x_3$ | 0 | 0,03 | 0,96 | 0,01 |
| | $x_4$ | 0 | 0,01 | 0 | 0,99 | | $x_4$ | 0,01 | 0 | 0,02 | 0,97 |

# ИСПОЛЬЗУЕМАЯ ЛИТЕРАТУРА

1. Теория информации: учебник для вузов / В.Т. Еременко, В.А. Минаев, А.П. Фисун, И.С. Константинов, А.В. Коськин, В.А. Зернов, ЮА. Белевская, С.В. Дворянкин; под общей научной редакцией В.Т. Еременко, В.А. Минаева, А.П. Фисуна, В.А.Зернова, А.В. Коськина. – Орел: ОрелГТУ, ОГУ, 2010. – 443 с.

2. Турчин Д.Е. Теория информации. Лабораторный практикум: электронное учебное пособие для студентов направления подготовки 09.03.02. «Информационные системы и технологии» / Д. Е. Турчин; КузГТУ. – Электрон. дан. – Кемерово, 2016. – 1 электрон. опт. диск (2,6 Мб).

3. Шавенько Н.К. Основы теории информации. Учебное пособие. – М,: Изд-во МИИГАиК, 2019. – 135 с.