МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра космического приборостроения и систем связи

УТВЕРЖДАЮ
Проректор по учебной работе
_____ О.Г. Локтионова
«22» 07 2019 г.

МЕТОДЫ И АЛГОРИТМЫ ПОМЕХОУСТОЙЧИВОГО КОДИРОВАНИЯ

Методические указания
по выполнению курсового проекта
для студентов, обучающихся по специальности
10.05.02 «Информационная безопасность
телекоммуникационных систем»
по дисциплине «Теория информации и кодирования»

Курск 2019

2

УДК 621.391

Составители: Д.С. Коптев, И.Г. Бабанин

Рецензент:

Доктор технических наук, старший научный сотрудник,
профессор кафедры космического приборостроения и систем связи
В.Г. Андронов

Методы и алгоритмы помехоустойчивого кодирования: методические указания по выполнению курсового проекта по дисциплине «Теория информации и кодирования» / Юго-Зап. гос. ун-т; сост.: Д.С. Коптев, И.Г. Бабанин. – Курск, 2019. – 17 с.: иллюстр. 2., табл. 3. – Библиогр.: с. 17.

Методические указания по выполнению курсового проекта содержат краткие теоретические сведения о информации, её свойствах, методах измерения её количества и качества, общих принципах кодирования информации в системах передачи, обработки и хранения, варианты заданий для выполнения курсового проекта и критерии его оценки.

Методические указания соответствуют учебному плану по специальности 10.05.02 «Информационная безопасность телекоммуникационных систем», а также рабочей программе дисциплины: «Теория информации и кодирования».

Предназначены для студентов, обучающихся по специальности 10.05.02 «Информационная безопасность телекоммуникационных систем», очной формы обучения.

Текст печатается в авторской редакции

Подписано в печать 22.04.19 Формат 60x84 1/16.
Усл. печ. л. 0,988. Уч.-изд. л. 0,894. Тираж 100 экз. Заказ. 546 Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

1 Цель работы

Ознакомиться с вероятностным подходом к определению количества информации. Научиться рассчитать среднее количество информации на одну букву, оценивать избыточность. Изучить основные эффективные коды и применить эти знания при выполнении задании по тексту

2 Теоретический материал

2.1 Предметный указатель

Как и любая научная дисциплина, сжатие информации использует свой своеобразный лексикон, во избежание дальнейших недоразумений приведем основные понятия:

Сжатие информации – это процесс сокращения количества битов, необходимых для хранения информации;

Сжатие без потерь – информация, восстановленная из сжатого состояния, в точности соответствует исходной (до начала сжатия);

Сжатие с потерями – информация, восстановленная после сжатия, только частично соответствует исходной (применяется при обработке изображений и звука);

Граф – совокупность множества узлов и множества дуг, направленных от одного узла к другому;

Дерево – граф, обладающий следующими свойствами:

- ни в один из узлов не входит более одной дуги (т.е. отсутствуют циклы);
- только в один узел не входит ни одной дуги, он называется корнем дерева;
- перемещаясь по дугам от корня, можно попасть в любой узел;

Лист дерева – узел, из которого не выходит ни одной дуги. В паре узлов дерева, соединенных между собой дугой, тот, из которого она выходит, называется родителем, другой же – ребенком. Два узла называются братьями, если имеют одного и того же родителя;

Двоичное дерево – дерево, у которого из всех узлов, кроме листьев, выходит по две дуги.

Дерево кодирования Хаффмана – двоичное дерево, у которого каждый узел имеет вес, и вес родителя равен суммарному весу го детей;

Входной алфавит – множество символов, входящих в сообщение.

2.2 Количество информации

Понятие "количества информации" часто встречается в технической литературе, однако на практике определить количество информации очень непросто.

С 40-х годов XX века предпринимаются попытки использовать понятие информации для объяснения и описания самых разнообразных явлений и процессов. В решении этой проблемы существуют два основных подхода, которые исторически возникли одновременно. В конце 40-х годов XX века один из основоположников кибернетики американский математик Клод Шеннон развил вероятностный подход к измерению количества информации, а работы по созданию ЭВМ привели к «объемному» подходу.

Рассмотрим более подробно вероятностный подход.

В качестве примера разберем опыт, связанный с бросанием правильной игральной кости, имеющей N граней. Результатом данного опыта считаем выпадение грани с одним из следующих знаков: 1, 2, . . . *N*.

Введем в рассмотрение численную величину — **энтропию** (обозначим ее *H*). **Энтропия - мера неопределенности некоторого опыта. В простейшем случае его исход зависит от выбора одного элемента из множества исходных.**

Согласно **шенноновской теории информации**, в случае равновероятного выпадения каждой из граней величины *N* и *H* связаны между собой **формулой Хартли**

$$H = \log_2 N .$$

При введении какой-либо новой величины важным является вопрос о единице измерения этой величины.

В соответствии с формулой Хартли становится очевидным, что энтропия H будет равна единице при N=2. Тогда, в качестве единицы измерения принимается количество информации, связанное с проведением опыта, состоящего в получении одного из двух равновероятных исходов (примером такого опыта может служить бросание монеты, при котором возможны два исхода- «орел», «решка»). Такая единица количества информации называется «бит».

В случае, когда вероятности P_i, результатов опыта (в примере, приведенном выше — бросания игральной кости) неодинаковы, имеет место формула Шеннона

$$H = -\sum_{i=1}^{N} P_i \cdot \log_2 P_i .$$

В случае равной вероятности событий $P_i = \frac{1}{N}$, формула Шеннона переходит в формулу Хартли (1).

Рассмотрим следующий пример. Необходимо определить количество информации, связанное с появлением каждого символа в сообщениях, записанные на русском языке. Считаем, что русский алфавит состоит из 33 букв и знака «пробел» для разделения слов.

По формуле Хартли

$$H = \log_2 34 \approx 5,09 \text{ бит.}$$

Однако в словах русского языка (равно как и в словах других языков) различные буквы встречаются неодинаково часто. В таблице 1 приведена вероятность частоты употребления различных знаков русского алфавита, полученная на основе анализа очень больших по объему текстов.

Таблица 1 - Частотность букв русского языка

i	Символ	*P(i)*	*i*	Символ	*P(i)*	*i*	Символ	*P(i)*
1	__	0,175	12	Л	0,035	23	Б	0,014
2	О	0,090	13	К	0,028	24	Г	0,012
3	Е	0,072	14	М	0,026	25	Ч	0,012
4	Ё	0,072	15	Д	0,025	26	Й	0,010
5	А	0,062	16	П	0,023	27	Х	0,009

6	И	0,062	17	У	0,021	28	Ж	0,007
7	Т	0,053	18	Я	0,018	29	Ю	0,006
8	Н	0,053	19	Ы	0,016	30	Ш	0,006
9	С	0,045	20	З	0,016	31	Ц	0,004
10	Р	0,040	21	Ь	0,014	32	Щ	0,003
11	В	0,038	22	Ъ	0,014	33	Э	0,003
						34	Ф	0,002

Найдем значение количества информации (энтропии) *H.* Для этого воспользуемся формулой Шеннона:

$$H = -\sum_{i=1}^{34} P_i \cdot \log_2 P_i \approx 4,72 \text{ бит.}$$

Полученное значение количества информации по Шеннону меньше вычисленного ранее. Это вытекает из основных свойств энтропии, как меры неопределенности сообщения. Количество информации, вычисляемое по формуле Хартли, является максимальным количеством информации, которое могло бы проходиться на один знак.

Аналогичные подсчеты количества информации можно провести и для других языков, например, использующих латинский алфавит — немецкий, французский и др. (26 различных букв и «пробел»).

По формуле Хартли получим

$$H = \log_2 27 \approx 4,76 \text{ бит.}$$

2.3 Избыточность сообщений

Чем больше энтропия, тем большее количество информации содержит в среднем каждый элемент сообщения.

Пусть энтропии двух источников сообщений $H_1 < H_2$ а количество информации, получаемое от них одинаковое, т.е. $I = n_1 H_1 = n_2 H_2$, где n_1 и n_2 - длина сообщения от первого и второго источников. Обозначим

$$\mu = \frac{n_2}{n_1} = \frac{H_1}{H_2}$$

При передаче одинакового количества информации сообщение тем длиннее, чем меньше его энтропия.

Величина μ, называемая коэффициентом сжатия, характеризует степень укорочения сообщения при переходе к кодированию состояний элементов, характеризующихся большей энтропией.

При этом доля излишних элементов оценивается коэффициентом избыточности:

$$r = \frac{H_2 - H_1}{H_2} = 1 - \frac{H_1}{H_2} = 1 - \mu$$

Русский алфавит, включая пропуски между словами, содержит 32 элемента, следовательно, при одинаковых вероятностях появления всех 32 элементов алфавита, неопределенность, приходящаяся на один элемент, составляет $H_0 = \log 32 = 5\,бит$.

Анализ показывает, что с учетом неравномерного появления различных букв алфавита $H = 4,42\,бит$, а с учетом зависимости двухбуквенных сочетаний, $H' = 3,52\,бит$ т.е. $H' < H < H_0$ Обычно применяют три коэффициента избыточности:

1) частная избыточность, обусловленная взаимосвязью $r' = 1 - H'/H$;

2) частная избыточность, зависящая от распределения $r'' = 1 - H/H_0$;

3) полная избыточность $r_0 = 1 - H'/H_0$.

Эти три величины связаны зависимостью $r_0 = r' + r'' - r' \cdot r''$.

Вследствие зависимости между сочетаниями, содержащими две и больше букв, а также смысловой зависимости между словами, избыточность русского языка (как и других европейских языков) $50\%(r_0 = 1 - H'/H_0 = 1 - 3,25/5 = 0,30)$.

Избыточность играет положительную роль, т.к. благодаря ней сообщения защищены от помех. Это используют при помехоустойчивом кодировании.

2.4 Общие сведения о эффективных кодах

В настоящее время существует несколько алгоритмов сжатия без потерь, частично это открытые алгоритмы, частично

коммерческие алгоритмы. Коммерческие алгоритмы не публикуются и познакомится с ними невозможно, за исключением ознакомления с результатами работы программ на базе этих алгоритмов. Соответствующие программы (ZIP, ARJ, RAR, ACE и др.) достаточно известны и с ними можно познакомиться самостоятельно.

Алгоритмы обратимого сжатия данных можно разделить на две группы:

1) Алгоритмы частотного анализа - подсчет частоты различных символов в данных и преобразование кодов символов с соответствия с их частотой.

2) Алгоритмы корреляционного анализа - поиск корреляций (в простейшем случае точных повторов) между различными участками данных и замена коррелирующих данных на код(ы), позволяющая восстановить данные на основе предшествующих данных. В простейшем случае точных повторов, кодом является ссылка на начало предыдущего вхождения этой последовательности символов в данных и длина последовательности.

Можно отметить следующие алгоритмы обратимого сжатия данных из первой группы:

1) Метод Хаффмана - замена кода равной длины для символов на коды неравной длины в соответствии с частотой появления символов в данных, коды минимальной длины присваиваются наиболее часто встречающимся символам. Если частоты появления символов являются степенью двойки (2n), то этот метод достигает теоретической энтропийной границы эффективности сжатия для методов такого типа.

2) Метод Шеннона-Фано - сходен с методом Хаффмана, но использует другой алгоритм генерации кодов и не всегда дает оптимальные коды (оптимальный код – код дающий наибольшее сжатие данных из всех возможных для данного типа преобразования).

3) Арифметическое кодирование - усовершенствование метода Хаффмана, позволяющее получать оптимальные коды для данных, где частоты появления символов не являются степенью

двойки (2n). Этот метод достигает теоретической энтропийной границы эффективности сжатия этого типа для любого источника.

Для второй группы можно назвать следующие алгоритмы:

1) **Метод Running (или RLE)** - заменяет цепочки повторяющихся символов на код символа и число повторов. Это пример элементарного и очень понятного метода сжатия, но, к сожалению, он не обладает достаточной эффективностью.

2) **Методы Лемпеля-Зива** - это группа алгоритмов сжатия, объединенная общей идеей: поиск повторов фрагментов текста в данных и замена повторов ссылкой (кодом) на первое (или предыдущее) вхождение этого фрагмента в данные. Отличаются друг от друга методом поиска фрагментов и методом генерации ссылок (кодов).

Информационные характеристики

Все равномерные коды являются избыточными. Это значит, что число двоичных символов, используемых для кодирования сообщения, всегда больше количества информации в этом сообщении. Процедура, направленная на устранение избыточности в передаваемом сообщении, называется эффективным или статистическим кодированием источника. Эффективный код всегда является неравномерным. Символам алфавита с большой вероятностью появления будут соответствовать короткие кодовые слова, символам с малой вероятностью появления – длинные кодовые слова. Задача состоит в выборе таких правил кодирования, чтобы число двоичных символов кода, требуемых на один символ источника, было по возможности меньшим.

Основная теорема кодирования, сформулированная Клодом Шенноном в 1948 г., устанавливает связь между энтропией источника $H(A)$ и средним числом двоичных символов кодового слова $\tilde{n}$:

• Для любого однозначно декодируемого кода всегда выполняется неравенство

$$\tilde{n} >= H(A)$$

• Существует однозначно декодируемый код, для которого выполняется неравенство

$$\tilde{n} < H(A) + 1$$

Это означает, что невозможно закодировать источник таким образом, что средняя длина кодового слова будет меньше энтропии. Кроме этого, обязательно существует код, для которого средняя длина кодового слова немного больше энтропии источника.

2.5 Кодирование по методу Хаффмана

Часто дает более экономный код, чем метод Щеннона-Фано, нашел широкое применение на практике, например, в факсимильных устройствах. Построение кода Хаффмана основывается на преобразовании, которое называется сжатием алфавита.

Алгоритм кодирования может быть представлен алгоритмом:

1. Расположить символы исходного алфавита А в порядке убывания вероятности.

2. Два наименее вероятных символа алфавита А будем считать одним символом нового сжатого алфавита А1.

3. Располагаем символы алфавита А1 в порядке убывания вероятности. И снова подвергаем его сжатию, как в пункте 2.

4. Повторяем процедуру, пока не придем к алфавиту, содержащему всего два символа.

5. Припишем символам последнего алфавита кодовые обозначения 0 (например - верхнему) и 1 (в нашем примере – нижнему). Это старшие символы будущих кодовых слов.

6. В предпоследнем алфавите кодовые обозначения получаются следующим образом:

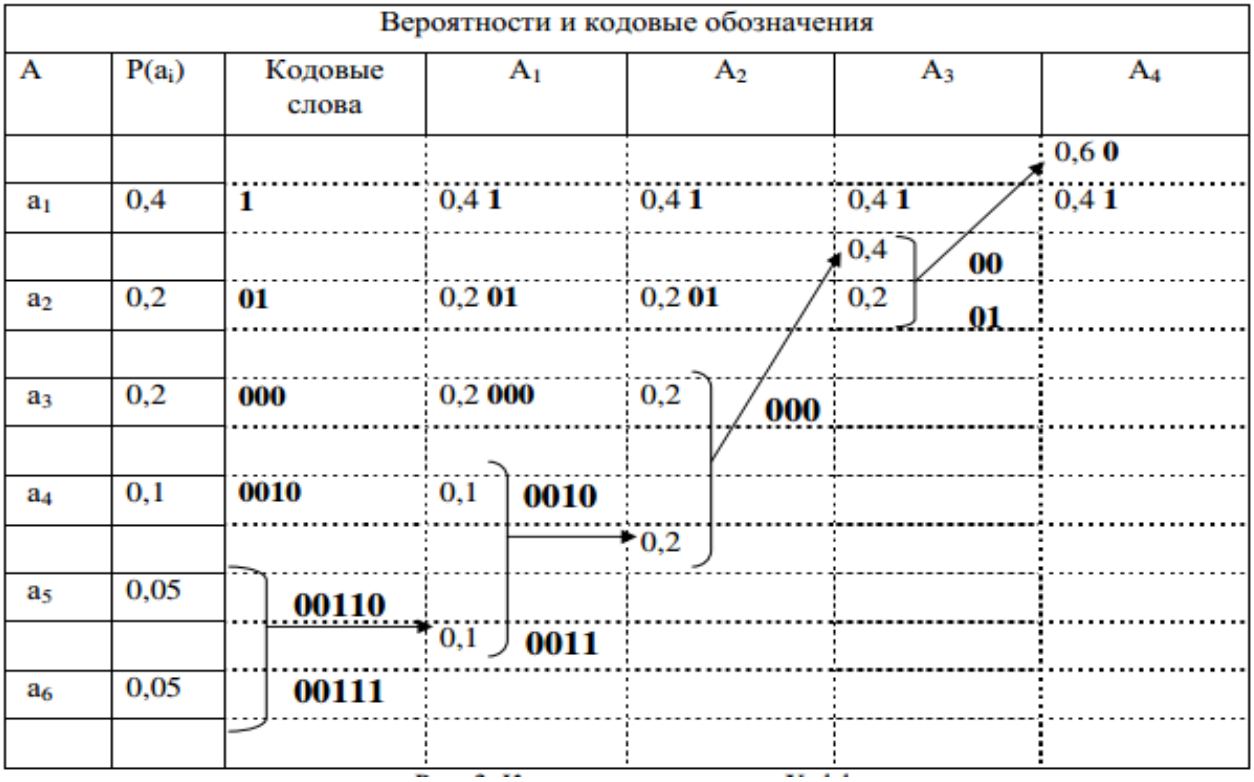
- Символ, который сохранился в последнем алфавите, имеет то же кодовое обозначение.

- Символам, которые слились в последнем алфавите, приписывают справа 0 (в нашем примере – верхнему символу) и 1 (нижнему символу).

7. Повторяем процедуру, последовательно возвращаясь к исходному алфавиту.

Средняя длина кодового слова получается равной ñ =2,3 бит.

В данном случае среднее число двоичных символов кода, приходящихся на один символ источника, получилось таким же, как в коде Шеннона-Фано. В общем случае математиками доказано, что код Хаффмана является самым экономным в том смысле, что никакой другой метод кодирования алфавита не позволяет получить среднее число двоичных символов на один символ алфавита меньше, чем в случае кода Хаффмана.



2.6 Кодирования по Хэммингу

Кодирование по Хэммингу весьма несложный процесс. Достоинство кода в том, что реализация алгоритма требует небольших ресурсов и может быть выполнена аппаратно.

Исходными данными для кодирования является произвольная двоичная последовательность, например.

Исходная битовая последовательность

№ бита	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	…	n
Значение бита	0	1	1	1	0	1	0	0	1	1	1	0	1	0	0	1	1	1	0	1	0	0	1

Прежде всего, двоичная последовательность разделяется на куски размером в m бит. Размеры кусков не произвольны, их длина определяются формулой

$$m = 2^r - r - 1,$$

где r – любое целое число большее 2. Куски исходной двоичной последовательности будем называть «блоки исходного кода» и обозначать ai. Рассмотрим для определенности r = 4, тогда m = 11.

Далее исходные коды расширяют до n бит каждый, дополняя r контрольными битами. Полученные n-битные коды образуются так:

- Позиции с номерами 2^r (i = 1, 2, …r) резервируются под контрольные биты;
- в остальные биты копируется исходный код в порядке следования его битов.

Расширенные блоки будем называть «блок кода» и обозначать b_i.

Затем вычисляют контрольные разряды. Для вычисления контрольных разрядов нужна вспомогательная матрица M размером (2r – 1) строк и r столбцов. Матрица заполняется по строкам, в каждую строку записывают двоичное представление чисел от 1 до $2^r - 1$, младшие биты пишут первыми.. Далее вычисляются контрольные разряды c_i, для этого из матрицы M выбираются и суммируются по колонкам все строки номера которых совпадают с ненулевыми битами блока кода b_i. Полученная строка из r битов записывается в контрольные разряды блока кода b_i в порядке следования битов. Вычисление контрольных разрядов c_i можно представить матричным умножением

$$c_i = \left(b_i\right)^T * M$$

здесь $(b_i)^T$ — строка (вместо столбца) расширенного кода, где контрольные биты равны 0.

Полученные блоки кода можно вновь преобразовать в битовую последовательность и передавать по каналу связи. Код Хемминга способен детектировать и исправлять 1 (одну) ошибку на блок.

2.7 Описание декодирования и исправления ошибки по Хэммингу

Переданная по информационному каналу в приемник битовая последовательность делится на куски по $n = 2^r - 1$ бит — получаются блоки кода. С каждым таким блоком выполняется операция

$$c_i = (b_i)^T * M,$$

здесь $(b_i)^T$ — строка (вместо столбца) расширенного кода. Причем здесь контрольные разряды участвуют в вычислении суммы.

Если получено, что все биты c_i равны нулю, то значит ошибок нет и коррекция не нужна. Если хотя бы один бит c_i не равен нулю, то имела место ошибка. Значение c_i преобразуют из битового представления в десятичное число i и бит блока кода с номером i — ошибочный бит (передан с ошибкой). Для исправления значение бита инвертируют: заменяют ноль на единицу, а единицу на ноль. В результате получаем правильное значение блока кода.

2.8 Равномерный двоичный код для русского языка.

№	буква	Кодовая комбинация		№	буква	Кодовая комбинация
1	**А**	**00001**		**17**	**Р**	**10001**
2	**Б**	**00010**		**18**	**С**	**10010**
3	**В**	**00011**		**19**	**Т**	**10011**

4	**Г**	**00100**		**20**	**У**	**10100**
5	**Д**	**00101**		**21**	**Ф**	**10101**
6	**Е,Ё**	**00110**		**22**	**Х**	**10110**
7	**Ж**	**00111**		**23**	**Ц**	**10111**
8	**З**	**01000**		**24**	**Ч**	**11000**
9	**И**	**01001**		**25**	**Ш**	**11001**
10	**Й**	**01010**		**26**	**Щ**	**11010**
11	**К**	**01011**		**27**	**Ъ**	**11011**
12	**Л**	**01100**		**28**	**Ы**	**11100**
13	**М**	**01101**		**29**	**Ь**	**11101**
14	**Н**	**01110**		**30**	**Э**	**11110**
15	**О**	**01111**		**31**	**Ю**	**11111**
16	**П**	**10000**		**32**	**Я**	**00000**

В заключение, контрольные разряды удаляются из блока и получается блок исходного кода. Эту операцию (проверка-коррекция) проводят с каждым блоком кода.

3 Задание на курсовое проектирование

Вариант студента выбирается по порядковому номеру студента в журнале преподавателя.

Провести статистическую обработку текста:

1) Найти статистические вероятности (относительные частоты) букв, используемых в тексте (другие знаки не учитывать).

2) Рассчитать среднее количество информации на одну букву. Оценить избыточность.

3) Провести кодирование кодом Хаффмана и рассчитать эффективность кода.

4) Выбрать из текста слово, состоящее минимум из 6 букв, и закодировать равномерным двоичным кодом (смотреть в теории выше).

5) Закодировать полученное двоичное слово кодом Хэмминга, исправляющим однократные ошибки.

6) Показать, как определяются однократные ошибки в разряде, совпадающем с номером варианта студента.

В качестве исходного текста предлагается использовать 2 страницы художественного произведения.

Варианты заданий на курсовое проектирования находятся у преподавателя.

4 Критерии оценивания курсового проекта

Формальные критерии (нормоконтроль) (от 0 до 30 баллов)

- оформление титульного листа, оглавления, заглавий и текста;
- оформление библиографии;
- использование зарубежной литературы;
- оформление приложений, применение иллюстративного материала;
- оформление ссылок, сносок и выносок;
- грамматика, пунктуация и шрифтовое оформление работы;
- соблюдение графика подготовки и сроков сдачи законченной работы.

Содержательные критерии (от 0 до 50 баллов)

- соответствие работы теме задания;
- выбор цели и постановка задач
- наличие вариантов решения и сравнительного анализа;
- структура работы, сбалансированность разделов;
- соответствие решений современному состоянию уровня техники;
- наличие элементов научной новизны, практическая ценность;
- детальность проработки решения;
- правильность деления объёма материала по разделам;
- степень самостоятельности работы;
- стиль изложения.

Защита (от 0 до 20 баллов)

- раскрытие содержания работы;
- структура и качество доклада, качество презентации;
- умение чётко и ясно излагать сущность задач и методов решения;
- умение оценивать качество предложенных решений, их сильные и слабые стороны;
- оперирование профессиональной терминологией;
- ответы на вопросы по теме работы.

Дополнительные баллы (от 0 до 20) могут быть получены за:

- апробацию материалов работы на научных конференциях;
- использование современных научных методов исследования и интернет-технологий;
- получение квалифицированной рецензии на работу;
- публикацию по теме работы в периодических научных изданиях.

Список использованных источников

1. Информатика [Текст]: учебник / В. В. Трофимов [и др.] ; под ред. В. В. Трофимова ; С.-Петерб. гос. ун-т экономики и финансов. - М. : Юрайт, 2011. - 911 с.

2. Острейковский, В. А. Информатика: учеб. пособие для студентов учреждений сред. проф. образования / В. А. Острейковский. - М. : Высшая школа, 2003. - 319 с.

3. Романова, Ю.Д. Информатика и информационные технологии: учебное пособие /Ю.Д. Романова, И.Г. Лесничая, В.И. Шестаков, И.В. Мисинг, П.А. Музычкин; под ред. Ю.Д. Романовой. – 3-е изд., перераб. И доп. – М.: Эксмо, 2008. – С. 309-404. – (Высшее экономическое образование).