

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Институт кибернетики  
Кафедра автоматки и компьютерных систем  
Направление подготовки – 09.03.02 Информационные системы и технологии

**БАКАЛАВРСКАЯ РАБОТА**

<b>Тема работы</b>
<b>Программно-алгоритмическое обеспечение для защиты цифровых изображений формата bmp</b>

УДК 004.932.056.5

Студент

Группа	ФИО	Подпись	Дата
8И2А	Чан Тхюи Зунг		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. АиКС	Вичугова Анна Александровна	к.т.н		

**КОНСУЛЬТАНТЫ:**

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Хаперская Алена Васильевна			

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Мезенцева Ирина Леонидовна			

**ДОПУСТИТЬ К ЗАЩИТЕ:**

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
Зав.каф. АиКС	Фадеев Александр Сергеевич	к.т.н.		

Томск – 2016 г.

## ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Код результатов	Результат обучения (выпускник должен быть готов)
<i><b>Профессиональные и общепринятые компетенции</b></i>	
P1	Применять базовые и специальные естественнонаучные и математические знания для комплексной инженерной деятельности по созданию, внедрению и эксплуатации геоинформационных систем и технологий, а также информационных систем и технологий в бизнесе.
P2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.
P3	Ставить и решать задачи комплексного анализа, связанные с созданием геоинформационных систем и технологий, информационных систем в бизнесе, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
P4	Выполнять комплексные инженерные проекты по созданию информационных систем и технологий, а также средств их реализации (информационных, методических, математических, алгоритмических, технических и программных).
P5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания геоинформационных систем и технологий, а также информационных систем и технологий в бизнесе.
P6	Внедрять, эксплуатировать и обслуживать современные геоинформационные системы и технологии, информационные системы и технологии в бизнесе, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
<i><b>Универсальные (общекультурные) компетенции</b></i>	
P7	Владеть иностранным языком на уровне, позволяющем работать в интернациональной профессиональной среде с пониманием культурных, языковых и социально-экономических различий партнеров.
P8	Эффективно работать индивидуально, а также в качестве члена или руководителя группы с ответственностью за работу коллектива при решении инновационных инженерных задач в области средств автоматизации и систем управления техническими объектами; демонстрировать при этом готовность следовать профессиональной этике и нормам
P9	Эффективно работать индивидуально и в качестве члена команды, состоящей из специалистов различных направлений и квалификаций.
P10	Демонстрировать личную ответственность за результаты работы и готовность следовать профессиональной этике и нормам ведения комплексной инженерной деятельности.
P11	Демонстрировать знания правовых, социальных, экологических и культурных аспектов комплексной инженерной деятельности, а также готовность к достижению должного уровня физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности.



Институт кибернетики  
 Кафедра автоматизации и компьютерных систем  
 Направление подготовки 09.03.02 Информационные системы и технологии

УТВЕРЖДАЮ:  
 Зав. кафедрой АиКС ИК  
 \_\_\_\_\_ Фадеев

А.С.

**ЗАДАНИЕ  
 на выполнение выпускной квалификационной работы**

В форме:

Бакалаврской работы
---------------------

Студенту:

<b>Группа</b>	<b>ФИО</b>
8И2А	Чан Тхюи Зунг

Тема работы:

<b>Программно-алгоритмическое обеспечение для защиты цифровых изображений формата bmp</b>	
Утверждена приказом директора (дата, номер)	№702/с от 04.02.2016

Срок сдачи студентом выполненной работы:	05.06.2016
--	------------

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

<p><b>Исходные данные к работе</b>  <i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i></p>	1 Изображение BMP
<p><b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b>  <i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<ol style="list-style-type: none"> <li>1. Анализ структуры изображения BMP</li> <li>2. Анализ цифровых водяных знаков</li> <li>3. Анализ алгоритма Alattar, Differenceexpansion</li> <li>4. Анализ алгоритма CPTE, MCPTE</li> <li>5. Реализация приложения</li> <li>6. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</li> <li>7. Социальная ответственность</li> </ol>
<p><b>Перечень графического материала</b>  <i>(с точным указанием обязательных чертежей)</i></p>	Презентация в формате *.pptx на 15 слайдах.

<b>Консультанты по разделам выпускной квалификационной работы</b> <i>(с указанием разделов)</i>	
<b>Раздел</b>	<b>Консультант</b>
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Старший преподаватель каф. менеджмента Хаперская Алена Васильевна
Социальная ответственность	Ассистенткаф. ЭБЖ Мезенцева Ирина Леонидовна
Основная	Доцент каф. АиКС Вичугова Анна Александровна

<b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b>	10.01.2016
---	------------

**Задание выдал руководитель:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. АиКС	Вичугова Анна Александровна	к.т.н.		10.01.2016

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8И2А	Чан Тхюи Зунг		10.01.2016

**Министерство образования и науки Российской Федерации**  
 Федеральное государственное бюджетное образовательное учреждение  
 высшего профессионального образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
 ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**



Институт кибернетики  
 Кафедра автоматизации и компьютерных систем  
 Направление подготовки 09.03.02 Информационные системы и технологии  
 Уровень образования – бакалавриат  
 Период выполнения – осенний/весенний семестр 2015/2016 учебного года

Форма представления работы:

Бакалаврская работа

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН  
 выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:	05.06.2016
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
30.04.2016	Основная часть	75
15.05.2016	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
25.05.2016	Социальная ответственность	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. АиКС	Вичугова Анна Александровна	к.т.н.		

СОГЛАСОВАНО:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
Зав. каф. АИКС	Фадеев Александр Сергеевич	к.т.н.		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И  
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
<b>8И2А</b>	<b>Чан Тхюи Зунг</b>

<b>Институт</b>	<b>Кибернетики</b>	<b>Кафедра</b>	<b>Автоматики и компьютерных систем</b>
<b>Уровень образования</b>	<b>Бакалавриат</b>	<b>Направление/специальность</b>	<b>Информационные системы и технологии</b>

**Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:**

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Работа с информацией, представленной в российских и иностранных научных публикациях, аналитических материалах, статистических бюллетенях и изданиях, нормативно-правовых документах; анкетирование; опрос
2. Нормы и нормативы расходования ресурсов	
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	Проведение анализ: потенциальные потребители результатов исследования, анализ конкурентных технических решений, технология QuaD, SWOT –анализ.
2. Определение возможных альтернатив проведения научных исследований	Определение морфологической матрицы для программы
3. Планирование и формирование бюджета научных исследований	Определение возможных альтернатив проведения научных исследований структуры работ в рамках научного исследования, трудоемкости выполнения работ, разработка графика проведения научного исследования, бюджет научно-технического исследования,
4. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Определение интегрального показателя финансовой эффективности, интегрального показателя ресурсоэффективности, интегрального показателя эффективности и сравнительной эффективности научного исполнения.

**Перечень графического материала (с точным указанием обязательных чертежей):**

1. Карта сегментирования рынка услуг по разработке интернет-ресурсов
2. Морфологическая матрица

3. Разработка графика проведения научного исследования

Дата выдачи задания для раздела по линейному графику

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель	Хаперская Алена Васильевна			

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8И2А	Чан Тхюи Зунг		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
<b>8И2А</b>	<b>Чан Тхюи Зунг</b>

<b>Институт</b>	<b>Кибернетики</b>	<b>Кафедра</b>	<b>Автоматики и компьютерных систем</b>
<b>Уровень образования</b>	<b>Бакалавриат</b>	<b>Направление/специальность</b>	<b>Информационные системы и технологии</b>

<b>Исходные данные к разделу «Социальная ответственность»:</b>	
<p>1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения</p>	<p>Данная работа является исследованием способа сокрытия информации в изображениях формата BMP. Основным исполнителем работы является программист. Объектом для исследования работы является изображение BMP. Работа содержит 7 различных алгоритмов.</p>
<b>Перечень вопросов, подлежащих исследованию, проектированию и разработке:</b>	
<p><b>1. Производственная безопасность</b>                      1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения                      1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения</p>	<p>В данном пункте анализируются вредные и опасные факторы, которые могут возникать при разработке или эксплуатации проектируемого решения:</p> <ul style="list-style-type: none"> <li>- Отклонение показателей микроклимата в помещении.</li> <li>- Превышение уровня шума.</li> <li>- Естественное освещение помещений.</li> <li>- Электробезопасность.</li> <li>- Пожаровзрывобезопасность.</li> </ul>
<p><b>2. Экологическая безопасность</b></p>	<p>Непосредственно с выполнением данной работы, могут быть связаны негативно влияющие на экологию факторы, сопутствующие эксплуатации ПК.</p>
<p><b>3. Безопасность в чрезвычайных ситуациях</b></p>	<p>В данном подразделе рассматриваются вероятные</p>

	чрезвычайные ситуации, которые могут возникнуть при разработке или эксплуатации проектируемого решения.
<b>4. Правовые и организационные вопросы обеспечения безопасности</b>	<p>-Специальные (характерные для проектируемой рабочей зоны) правовые нормы трудового законодательства. В данном пункте приводятся особенности трудового законодательства применительно к конкретным условиям проекта.</p> <p>- Организационные мероприятия при компоновке рабочей зоны. В данном пункте приводятся эргономические требования к правильному расположению и компоновке рабочей зоны исследователя, проектируемой рабочей зоны в производственных условиях для создания комфортной рабочей среды.</p>

Дата выдачи задания для раздела по линейному графику	
--	--

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Мезенцева Ирина Леонидовна			

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8И2А	Чан Тхюи Зунг		

## РЕФЕРАТ

Выпускная квалификационная работа состоит из 80с., 2брис., 24 табл., 6 источников литературы, 5 прил.

Ключевые слова: BMP, CPTE, MCPTE, Alattar, сокрытия информации, цифровые водяные знаки.

Объектами исследования являются изображения BMP.

Цель работы – защита информации в изображении и защита изображения от несанкционированного изменения.

В процессе выполнения работы проведен анализ возможности сокрытия информации в изображениях формата BMP, изучены алгоритмы CPTE, MCPTE, Alattar, Difference expanition для сокрытия информации.

В результате исследования было разработано программно-алгоритмическое обеспечение для защиты цифровых изображений формата BMP и сокрытия информации в изображении BMP. Приложение разработано на языке программирования C# в среде «Microsoft Visual Studio 2013».

Область применения: результаты работы могут быть использованы для защиты графических файлов формата BMP от не санкционированного изменения.

## **ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

BMP – формат изображения

24-х – формат 24 бита изображения

8-х - формат 8 бит изображения

RGB, ( $x_R, x_G, x_B$ ) – Red, Green, Blue

Reversible watermarking - Реверсивные водяные знаки

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	16
1 СТРУКТУРА ИЗОБРАЖЕНИЯ BMP .....	17
2 СКРЫТИЕ ИНФОРМАЦИИ.....	21
2.1 Основные понятия .....	21
2.2 Цифровые водяные знаки.....	22
2.3 Алгоритмы сокрытия информации и генерации водяных знаков	22
2.3.1 Алгоритм сокрытия информации.....	23
2.3.2 Алгоритм восстановления.....	24
2.3.3 Алгоритм «DIFFERENCE EXPANSION».....	24
2.3.4 Алгоритм для не расширяемых элементов.....	25
2.3.5 Алгоритм шифрования MD5.....	26
2.4 Алгоритмы для защиты сообщения в изображении.....	32
2.4.1 Алгоритм CPTE.....	32
2.4.2 Алгоритм MCPTE .....	33
2.4.3 Функция ValAndNext.....	36
3 РЕЗУЛЬТАТ РАБОТЫ .....	42
3.1 Результат защиты изображения BMP .....	42
3.2 Результат защиты сообщения в изображениях.....	46
4 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ.....	52
4.1. Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения.....	52
4.1.1. Потенциальные потребители результатов исследования .....	52

4.1.2. Анализ конкурентных технических решений.....	52
4.1.3 Технология QuaD .....	53
4.1.4 SWOT-анализ.....	54
4.2. Определение возможных альтернатив проведения научных исследований .....	55
4.3 Планирование научно-исследовательских работ .....	55
4.3.1 Структура работ в рамках научного исследования.....	55
4.3.2. Определение трудоемкости выполнения работ.....	56
4.3.3 Разработка графика проведения научного исследования.....	57
4.3.4 Бюджет научно-технического исследования (НТИ) .....	62
4.3.4.1 Расчет материальных затрат НТИ.....	62
4.3.4.2 Расчет затрат на специальное оборудование для научных (экспериментальных) работ .....	62
4.3.4.3 Основная заработная плата исполнителей темы .....	63
4.3.4.4 Дополнительная заработная плата исполнителей темы .....	65
4.3.4.5 Отчисления во внебюджетные фонды (страховые отчисления) .....	65
4.3.4.6 Накладные расходы .....	66
4.3.4.7 Формирование бюджета затрат научно-исследовательского проекта.....	67
4.4 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования .....	67
<b>5 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ.....</b>	<b>70</b>
Введение.....	70
5.1 Производственная безопасность .....	71

5.1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения .....	72
5.1.1.1 Отклонение показателей микроклимата в помещении.....	72
5.1.1.2 Превышение уровня шума.....	73
5.1.1.3 Естественное освещение помещений .....	73
5.1.2 Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения .....	74
5.1.2.1 Электробезопасность.....	74
5.1.2.2 Пожаровзрывобезопасность .....	76
5.2 Экологическая безопасность .....	76
5.3 Безопасность в чрезвычайных ситуациях .....	77
5.3.1 Инструкция по помещению безопасности.....	77
5.3.2 Обязанности проживающих в помещениях .....	78
5.3.3 Необходимые действия при возникновении пожара в помещении .....	78
5.4. Правовые и организационные вопросы обеспечения безопасности .....	79
5.4.1 Правовые нормы трудового законодательства.....	79
5.4.2 Организационные мероприятия обеспечения безопасности.....	80
ЗАКЛЮЧЕНИЕ .....	81
СПИСОК ПУБЛИКАЦИЙ .....	82
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	83
Приложение А.(Обязательное) .....	84
Приложение Б.(Обязательное).....	86
Приложение В.(Обязательное) .....	88

ПриложениеГ.(Рекомендуемое).....	91
ПриложениеД. (Рекомендуемое) .....	93

## **ВВЕДЕНИЕ**

В настоящее время информация является одним из ценнейших предметов. Получение доступа к ней с появлением глобальных компьютерных сетей стало невероятно простым. В то же время, легкость и скорость такого доступа значительно повысили угрозу нарушения безопасности данных при отсутствии мер относительно их защиты, а именно угрозу несанкционированного доступа к информации. Поэтому проблема разработки, совершенствования и применения методов защиты информации в процессе её хранения и передачи на сегодняшний день является одной из наиболее актуальных. В современных системах защиты информации огромную роль играют не только методы стеганографии, но и методы цифровые водяные знаки.

В рамках данной работы мы исследуем способы для защиты информации в изображениях формата BMP. По результатам анализа будет разработано программно-алгоритмическое обеспечение в виде приложения.

## 1 СТРУКТУРА ИЗОБРАЖЕНИЯ BMP

BMP (от англ. Bitmap Picture) — формат хранения растровых изображений, разработанный компанией Microsoft. Файлы формата BMP могут иметь расширения .bmp [1].

Любой bmp-файл состоит из четырех частей:

- Заголовок файла (File header)
- Информационный заголовок (Image header)
- Таблица цветов (Color Table)
- Данные изображения (BitMap Array)

Если в файле хранится изображение с глубиной цвета 24 бита (16 млн. цветов), то таблица цветов может отсутствовать, однако в 256-цветном случае она есть[2].

Заголовок файла начинается с сигнатуры «BM», а затем идет длина файла, выраженная в байтах. Следующие 4 байта зарезервированы для дальнейших расширений формата, а заканчивается этот заголовок смещением от начала файла до записанных в нем данных изображения. Структура части «Заголовок файла» представлена на таблице 1 (14 байт).

Таблица 1. Структура файл «Заголовок файла»

Наименование	Количество байт	Описание
Type	2	Сигнатура файла BMP «B» и «M»
Size	4	Размер файла
Reserved 1	2	Если не используется, то значение равно 0
Reserved 2	2	Если не используется, то значение равно 0
OffBits	4	Местонахождение данных растрового массива

Информационный заголовок начинается с собственной длины (она может изменяться, но для 256-цветного файла составляет 40 байт) и содержит размеры изображения, разрешение, характеристики представления цвета и другие параметры.

«Количество плоскостей» могло применяться в файлах, имеющих небольшую глубину цвета. При числе цветов 256 и больше оно всегда равно 1,

поэтому сейчас это поле уже можно считать устаревшим, но для совместимости оно сохраняется[5].

Размер изображения — количество байт памяти, требующихся для хранения этого изображения, не считая данных палитры.

Горизонтальное и вертикальное разрешения измеряются в точках растра на метр. Они особенно важны для сохранения масштаба отсканированных картинок. Изображения, созданные с помощью графических редакторов, как правило, имеют в этих полях нули.

Число цветов позволяет сократить размер таблицы палитры, если в изображении реально присутствует меньше цветов, чем это допускает выбранная глубина цвета.

Число основных цветов — идет с начала палитры, и его желательно выводить без искажений. Данное поле бывает важно тогда, когда максимальное число цветов дисплея было меньше, чем в палитре Bmp-файла.

Структура части «Информационный заголовок» представлена на таблице 2 (40 байт).

Таблица 2. Структура файл «Заголовок изображения»

Наименование	Количество байт	Описание
Size	4	Длина этого заголовка
Width	4	Ширина изображения
Height	4	Высота изображения
Planes	2	Число цветовых плоскостей
BitCount	2	Бит(писел)
Compression	4	Метод сжатия. Если значение равно 0, то нельзя сжать
SizeImage	4	Размер изображения
XPelsPerMeter	4	Горизонтальное разрешение
YPelPerMeter	4	Вертикальное разрешение
ClrUsed	4	Число цветов изображения
ClrImportant	4	Число основных цветов

За информационным заголовком следует таблица цветов, представляющая собой массив из 256 (по числу цветов) 4-байтовых полей. Каждое поле соответствует своему цвету в палитре, а три байта из четырех —

компонентам синей, зеленой и красной составляющих для этого цвета. Последний, самый старший байт каждого поля зарезервирован и равен 0.

Таблица 3. Таблица цветов

Наименование	Количество байт	Описание
ColorTable	1024	256 элементов по 4 байта

После таблицы цветов находятся данные изображения, которые представляют собой список пикселей, из которых состоит изображение. Для изображения 8-х бит, каждый пиксель описывается по 1 байту. Для изображения 24-х бит, каждый пиксель описывается по 3 байта.

Таблица 4. Данные изображения

Наименование	Количество байт	Описание
ColorTable	1024	256 элементов по 4 байта

Для отображения данных о структуре изображения ВМРная языке программирования С# был создан класс «Sizeimage». Класс содержит указатель. Например, необходимо получить отображение размера файла. Значит, указателю нужно находиться в бите номер 2. Биты 2,3,4,5 отображают данные о изображении (потому что количество байт размера: 4). Программный код представлен далее:

```

fstream.Position = 2;
byte2 = read.ReadByte();
byte3 = read.ReadByte();
byte4 = read.ReadByte();
        byte5 = read.ReadByte();

```

Кроме этого, необходимо перемножить двоичное число на восьмеричное число.

```

sizeofFile = byte3 * Math.Pow(256, 3) + byte2 * Math.Pow(256, 2) + byte1 * Math.Pow(256, 1) +
byte0;

```

Итак, получен `sizeofFile` - результат размера изображения. Аналогично, можно прочитать данные о ширине и высоте изображения, также размер длины этого заголовка:

```
fstream.Position = 14;
    byte0 = read.ReadByte();
    byte1 = read.ReadByte();
    byte2 = read.ReadByte();
    byte3 = read.ReadByte();
    KichthuocInfoheader = byte3 * Math.Pow(256, 3) + byte2 *
Math.Pow(256, 2) + byte1 * Math.Pow(256, 1) + byte0;
    fread.Position = 18;
    byte0 = read.ReadByte();
    byte1 = read.ReadByte();
    byte2 = read.ReadByte();
    byte3 = read.ReadByte();
    width = byte3 * Math.Pow(256, 3) + byte2 * Math.Pow(256, 2) +
byte1 * Math.Pow(256, 1) + byte0;
    byte0 = read.ReadByte();
    byte1 = read.ReadByte();
    byte2 = read.ReadByte();
    byte3 = read.ReadByte();
    height = byte3 * Math.Pow(256, 3) + byte2 * Math.Pow(256, 2) +
byte1 * Math.Pow(256, 1) + byte0;
```

## 2СКРЫТИЕ ИНФОРМАЦИИ

### 2.1 Основные понятия

Скрытие информации является методом, который скрывает содержимое секретного сообщения в объекте (например: изображение, видео). Цель метода – это защита информации или защита автора информации. Скрытие информации разделится на два метода:

- Стеганография (Steganography): безопасность данных между отправителем и пользователей. Метод разработан на основе техники сокрытия информации, с целью сокрытия наибольший объем информации в скрываемом объекте.
- Цифровые водяные знаки: технология, созданная для защиты авторских прав мультимедийных файлов.

Объектом сокрытия может либо изображение, либо видео. Изображение не изменяется с течением времени, а видео данные меняются. Поэтому алгоритмы сокрытия данных для этих объектов различны и зависят от типа изображения (BMP, JPEG), зрительной системы человека. Технически сокрытие не изменит размера изображения.

Различают следующие критерии оценки степени технического сокрытия изображения:

- безопасность;
- вместимость сокрытия изображения: определяется между секретными информациями и размерами изображениями. Цель алгоритма: сокрытия наибольший объем информации в скрываемом объекте.
- устойчивость;
- целостность: шифрование используется для предотвращения изменения информации при передаче или хранении;

- качество изображения после сокрытия информации оценивается визуально, а кроме того, математически.

## **2.2 Цифровые водяные знаки**

Цифровые водяные знаки являются эффективным способом защиты авторских прав в рамках задачи обеспечения информационной безопасности. Цель способа: защита объекта, который содержит важное скрытое сообщение.

Цифровые водяные знаки разделяют по степени хрупкости.

Знак называют хрупким, если при малейшей модификации его уже нельзя обнаружить. Такие знаки обычно используются для проверки целостности.

Знак называют полу-хрупким, если он выдерживает незначительные модификации сигнала, но вредоносные преобразования не выдерживает. Такие знаки обычно используют для обнаружения атаки на сигнал.

Таким образом, хрупкие водяные знаки (*fragile watermarking*) встроены в информационное содержание данных. Когда содержание изменилось, водяной знак будет меняться соответственно.

Знак называют надёжным (*robust watermarking*), если он противостоит всем известным видам атак. Такие знаки используют в системах защиты от копирования и идентификации, в частности, для выявления информации об авторских правах. Эта технология используется в водяных знаках данных, встроенных мультимедийных создателем содержания, например, владелец лейбла или встроенных в этикетке, например, серийный номер.

Кроме этого, цифровые водяные знаки разделены на 2 вида: видимые (*visible watermarking*) и невидимые (*invisible watermarking*)

## **2.3 Алгоритмы сокрытия информации и генерации водяных знаков**

М.У. Wu и J.Н. Lee развили алгоритм сокрытия информации в бинарной изображении. На основе алгоритма Wu-Lee, YuYanChen, HsiangKuangPan, YeCheeTseng создали алгоритм СРТ. В алгоритме использована ключевая и

массовая матрицы. Алгоритм позволяет скрыть информацию в изображении. Основным недостатком данного метода остается невысокое количество информации. Количество скрытия информации в матрице  $F$  с размером:  $m \times n$ :  $r$ (бит) где  $r = \lfloor \log_2(mn+1) \rfloor$ . Поэтому, в рамках настоящей работы, мы будем исследовать алгоритм СРТЕ для улучшения количество информации, а также качества изображения. Кроме этого, мы ещё исследуем способ для защиты не только информации в изображении, но и защита этой изображения.

Основным алгоритмом выбран алгоритм «Реверсивные водяные знаки», суть которого в разделении исходное изображение на пары пикселей. Для каждого пикселя  $(x, y)$ , мы вычислим  $h = x - y$ , сдвигаем  $h$  бит влево. Вправо остается пустой бит. Потом выполняется вставка бита цифрового водяного знака. После вставки бит  $b$ ,  $h$  преобразуется в  $h' = 2h + b$ ,  $(x, y)$  преобразуется в  $(x', y')$ :

$$\begin{cases} x' = \left\lfloor \frac{x + y}{2} \right\rfloor + \left\lfloor \frac{h' + 1}{2} \right\rfloor \\ y' = \left\lfloor \frac{x + y}{2} \right\rfloor - \left\lfloor \frac{h'}{2} \right\rfloor \end{cases}$$

Значение  $(x', y')$  принадлежит промежутку  $[0, 255]$ .

Идея разработанного алгоритма «ALATTAR» состоит в преобразовании пары пикселей в вектор  $U$  с несколькими пикселями:  $U = (u_0, u_1, \dots, u_{n-1})$  ( $n > 2$ ).

### 2.3.1 Алгоритм скрытия информации

Для скрытия бит  $B = (b_1, b_2, \dots, b_{n-1})$  в  $U = (u_0, u_1, \dots, u_{n-1})$  необходимо выполнить следующие шаги.

Вычисление:  $V = (v_0, v_1, \dots, v_n)$ .

$$v_0 = \left\lfloor \frac{\sum_{i=0}^{n-1} a_i u_i}{\sum_{i=0}^{n-1} a_i} \right\rfloor, v_j = 2(u_j - u_0) + b_j, j=1, \dots, n-1 \quad (a_i > 0, )$$

Вычисление:  $U' = (u'_0, u'_1, \dots, u'_{n-1})$ :

$$u'_0 = v_0 - \left[ \frac{\sum_{i=0}^{n-1} a_i v'_i}{\sum_{i=0}^{n-1} a_i} \right], u'_j = v'_j + u'_0, j=1, 2, \dots, n-1$$

### 2.3.2 Алгоритм восстановления

Для восстановления вектора  $U = (u_0, u_1, \dots, u_{n-1})$  из вектора  $U' = (u'_0, u'_1, \dots, u'_{n-1})$  необходимо выполнить следующие шаги.

Изменение  $U'$  на  $V' = (v'_0, v'_1, \dots, v'_{n-1})$ :

$$v_0 = \left[ \frac{\sum_{i=0}^{n-1} a_i u'_i}{\sum_{i=0}^{n-1} a_i} \right], v_j = \left[ \frac{u'_j - u'_0}{2} \right], j=1, 2, \dots, n-1$$

Потом мы получим  $V$  по формуле:

$$b_j = (u'_j - u'_0) \bmod 2, j=1, 2, \dots, n-1$$

Восстановление вектор  $U$  из  $V'$  по формуле:

$$u_0 = v'_0 - \left[ \frac{\sum_{i=0}^{n-1} a_i v'_i}{\sum_{i=0}^{n-1} a_i} \right], u_j = v'_j + u_0, j=1, 2, \dots, n-1$$

### 2.3.3 Алгоритм «DIFFERENCE EXPANSION»

Было разработано оригинальный алгоритм DIFFERENCE EXPANSION.

Для скрытия бит  $B = (b_1, b_2, \dots, b_{n-1})$  в  $U = (u_0, u_1, \dots, u_{n-1})$  необходимо выполнить следующие шаги:

1. сортировка элементов вектора  $U$  по возрастанию, получается вектор  $V$ ;
2. определение базового элемента: предположим  $a = V_m$ , ( $m = \lfloor \frac{n}{2} \rfloor$ ), необходимо определить значение  $k$ , при которых значения  $u_k = a$  и  $u_k -$  первое значение;
3. скрытие бит  $b_j$  в элементе  $u_j$  по формуле:

$$h_j = |u_j - a| \quad (j = 0, \dots, n-1 \text{ и } j \neq k)$$

$$u'_j = \begin{cases} a + (2h_j + b_j) & \text{если } u_j \geq a \text{ и } j \neq k \\ a - (2h_j + b_j) & \text{если } u_j < a \text{ и } j \neq k \\ a & \text{если } j = k \end{cases}$$

Для восстановления вектор  $U = (u_0, u_1, \dots, u_{n-1})$  из вектора  $U'$  ( $U'$  уже содержал информацию) необходимо выполнить следующие шаги:

1. сортировка элементов вектора  $U'$  по возрастанию, получается вектор  $V'$ ;
2. определение базового элемента: Необходимо найти максимальное значение  $j$ , при которых  $1 \leq j \leq \lfloor \frac{n}{2} \rfloor$  и  $v'_{j-1} \leq v'_j - 2$ . Предположим  $a = v'_0$ . Необходимо найти  $k$ , при которых значения  $u'_k = a$ ;
3. восстановление  $u_j$  из  $u'_j$  по формуле:

$$h'_j = |u'_j - a|, b_j = h'_j \bmod 2 \quad (j \neq k)$$

$$u'_j = \begin{cases} a + \lfloor \frac{h'_j}{2} \rfloor & \text{если } u'_j \geq a \text{ и } j \neq k \\ a - \lfloor \frac{h'_j}{2} \rfloor & \text{если } u'_j < a \text{ и } j \neq k \\ a & \text{если } j = k \end{cases}$$

Если элементы не принадлежат промежутку  $[0, 255]$ , то есть другой алгоритм для решения задачи.

### 2.3.4 Алгоритм для нерасширяемых элементов

В рамках данного алгоритма выполняются следующие шаги:

1. сортировка элементов вектора  $U = (u_0, u_1, \dots, u_{n-1})$  по возрастанию, получается вектор  $V$ ;
2. определение базового элемента: предположим  $a = V_m$ , ( $m = \lfloor \frac{n}{2} \rfloor$ ), необходимо определить значение  $k$ , при которых значения  $u_k = a$  и  $u_k$  – первое значение;
3. сокрытие бит  $b_j$  в элементе  $u_j$  по формуле:

$$h_j = |u_j - a| \quad (j = 0, \dots, n-1 \text{ и } j \neq k)$$

$$u'_j = \begin{cases} a + \left( 2 \lfloor \frac{h_j}{2} \rfloor + b_j \right) & \text{если } u_j \geq a + 2 \text{ и } j \neq k \\ a - \left( 2 \lfloor \frac{h_j}{2} \rfloor + b_j \right) & \text{если } u_j < a - 2 \text{ и } j \neq k \\ u_j & \text{если } a - 1 \leq u_j \leq a + 1 \end{cases}$$

Для восстановления вектор  $U = (u_0, u_1, \dots, u_{n-1})$  из вектора  $U' = (u'_0, u'_1, \dots, u'_{n-1})$ , необходимо сохранить всех битов  $h_j$  ( $LSB(h_j)$ ):

1. сортировка элементов вектора  $U' = (u'_0, u'_1, \dots, u'_{n-1})$  по возрастанию, получается вектор  $V'$ ;
2. определение базового элемента:  $a = V'(\lfloor \frac{n}{2} \rfloor)$ ;
3. вычисление  $b_j = |u'_j - a| \bmod 2$ , предположим  $h'_j = |u'_j - a|$

$$u_j = \begin{cases} a + \left(2 \lfloor \frac{h'_j}{2} \rfloor + LSB(h_j)\right) & \text{если } u'_j \geq a + 2 \\ a - \left(2 \lfloor \frac{h'_j}{2} \rfloor + LSB(h_j)\right) & \text{если } u'_j \leq a - 2 \\ u'_j & \text{если } a - 1 \leq u'_j \leq a + 1 \end{cases}$$

### 2.3.5 Алгоритм шифрования MD5

Необходимо определить количество битов, которым надо сокрыть в изображении. В этой задаче мы используем шифрование MD5 для получения цифрового знака изображения в формате BMP.

MD5 (англ. MessageDigest 5) — 128-битный алгоритм хеширования, разработанный профессором Рональдом Л. Ривестом из Массачусетского технологического института (Massachusetts Institute of Technology, MIT) в 1991 году. Предназначен для создания «отпечатков» или дайджестов сообщения произвольной длины и последующей проверки их подлинности. Широко применялся для проверки целостности информации и хранения паролей в закрытом виде[6]. На рисунке 1 показана генерация дайджеста сообщения алгоритмом MD5.

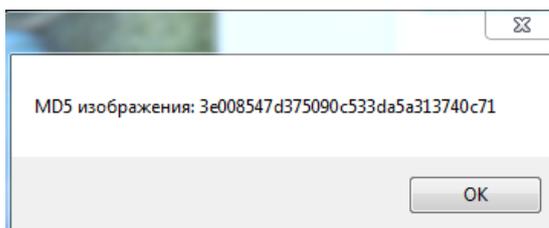


Рис.1. Цифровой знак

Процесс выполняется следующим образом:

1. использование MD5 и получение цифрового знака;
2. скрывание цифрового знака в изображении, в результате получается новое изображение, которое содержит цифровые водяные знаки;
3. выполнение обратного процесса: из выходного изображения получаем исходное изображение и цифровой знак;
4. сравнение значений цифровых знаков между первым и вторым изображениями. Если они не равны, значит, изображение изменилось.

Далее представлен код реализации алгоритма Alattar на языке программирования C#. Чтобы вычислить векторы  $v$  и  $u'$  по формуле был создан класс «NhungTin\_Alattar». Необходимо определить значения  $u_0$ . Сначала задаем значения  $u_0 = 0$ -й бит и  $u_1, u_2, \dots, u_n = 1$ -й, 2-й, ...  $n$ -й бит.

```

{
int n = pixel.Count();
int[] tempPixel = new int[n];
int[] resultPixel = new int[n];
int sumU = 0, sumV = 0, temp = 0;
for(int i = 0; i < n; i++) sumU += pixel[i];
//-----это v
tempPixel[0] = (int) Math.Floor((sumU*1.0)/n); //tempPixel это v
for(int i = 1; i < n; i++)
{
tempPixel[i] = 2*(pixel[i] - pixel[0]) + bit[i-1];
}
for (int i = 1; i < n; i++) sumV += tempPixel[i];
temp = (int) Math.Floor((sumV*1.0)/n);
resultPixel[0] = tempPixel[0] - temp;
//-----это u'

for (int i = 1; i < n; i++)
{
resultPixel[i] = tempPixel[i] + resultPixel[0]; //resultPixel это u'
}

```

```

return resultPixel;
}

```

Функция возвращает значения  $u'$ , которые содержат значения бит  $b$ . Далее получаем массив элементов, значения которого должны лежать в промежутке  $[0,255]$ .

```

{
int[] result = NhungTin_Alattar(pixel, bit);
for (int i = 0; i < pixel.Count(); i++)
if (result[i] < 0 || result[i] > 255) return false;
return true;
}

```

Если значения элементов не принадлежат промежутку  $[0,255]$ , то необходимо использовать разработанную функцию «NhungTinChenBitThap». Необходимо сортировать tempPixel (как  $U$ ) и задать значения базового элемента  $a = \text{tempPixel}[\text{tempPixel.Count()}/2]$ :

```

int[] tempPixel = new int[pixel.Count()];
int[] resultPixel = new int[pixel.Count()];
for (int i = 0; i < pixel.Count(); i++) tempPixel[i] = pixel[i];
//-----
Array.Sort(tempPixel);
int a = tempPixel[tempPixel.Count() / 2];
//int k = TimChiSo_Thuan(pixel, a);
//-----
int index = 0;
for (int i = 0; i < pixel.Count(); i++)
if (Math.Abs(pixel[i] - a) <= 1) resultPixel[i] = pixel[i];
else
{
if (pixel[i] >= a + 2)
{
resultPixel[i] = a + (2 * (Math.Abs(pixel[i] - a) / 2) + bit[index]);
}
else resultPixel[i] = a - (2 * (Math.Abs(pixel[i] - a) / 2) + bit[index]);
index++;
}
}

```

```
return resultPixel;
```

Функция возвращает значения `u'` (`resultPixel`), которые содержат значения бит `b`. Далее создаем функцию MD5.

```
public static string CalculateMD5Hash(int[] input)
{
    System.Security.Cryptography.MD5 md5 = System.Security.Cryptography.MD5.Create();
    byte[] inputByte = new byte[input.Count()];
    for (int i = 0; i < input.Count(); i++) inputByte[i] = (byte)input[i];
    byte[] hash = md5.ComputeHash(inputByte);
    StringBuilder sb = new StringBuilder();
    for (int i = 0; i < hash.Length; i++)
    {
        sb.Append(hash[i].ToString("x2"));
    }
    return sb.ToString();
}
```

Используем функцию «OpenFileDialog» для загрузки изображения. Задаем постоянный размер 480\*480 в Form (C#).

```
OpenFileDialog folder = new OpenFileDialog();
    folder.Filter = "(.bmp)*.bmp";
    folder.ShowDialog();
    if (folder.FileName.Length != 0)
    {
        DirFile = folder.FileName;
        NameFile = folder.SafeFileName;
        pic_ImageMain.ImageLocation = folder.FileName;
        ImageInfo imageInfo = new ImageInfo(pic_ImageMain.ImageLocation, NameFile, false);
        pic_ImageMain.Height = 480;
        pic_ImageMain.Width = 480;
    }
    lb_Info.Text = "неизвестно";
    lb_PSNR.Text = "PSNR: не известно ";
}
```

Далее необходимо использовать функцию MD5 для шифрования изображения.

```
for (int i = PartSize- 1; i >= 0; i--)
    {
for (int j = PartSize-1; j >= 0; j--)
    {
for (int u = 0; u < chieungang; u++)
for (int v = 0; v < chieudoc; v++)
        tmp_dataImage[v * chieungang + u] = image.pixelMatrix[j*chieudoc +
v][i*chieungang + u];
        myhash = Functions.MD5ToBit(Functions.CalculateMD5Hash(tmp_dataImage));
    }
    }
```

Далее начнем генерацию цифровых водяных знаков в изображении. Для этого необходимо создать функцию, которая содержит «myhash» и «pixel» для сокрытия «myhash» в «pixel». Самое трудное – это необходимо частично скрыть данные изображения, например: 2\*2, 3\*3, 4\*4, 5\*5. Если элементы все расширяемые, то легкий случай. Но элементы не только расширяемые, но и не расширяемые, то надо использовать функции «NhungTin\_Alattar» и «NhungTinChenBitThap» [8]. Например, для случая 5\*5:

```
int count = 0;
int[] tmpData = newint[size];
int[] resData = newint[size];
int[] bitHide = newint[size-1];
int part = hash.Length/(size-1);
int num = pixel.Length / size;
int[] map = newint[num];
for (int i = 0; i < num; i++)
    {
if(count < part)
    {
for(int j = 0; j < size; j++) tmpData[j] = pixel[i*size + j];
for (int j = 0; j < size - 1; j++) bitHide[j] = hash[(size - 1) * count + j];
        check = ReversibleWatermarking.KiemtraKhaMo(tmpData, bitHide);
```

```

if (check)
    {
        resData = ReversibleWatermarking.NhungTin_Alattar(tmpData, bitHide);
        map[i] = 1;
        count++;
for (int j = 0; j < size; j++) result[i * size + j] = resData[j];
        }
    }
elsebreak;
    }
for (int i = 0; i < num; i++)
    {
for (int j = 0; j < size; j++) tmpData[j] = pixel[i * size + j];
if (ReversibleWatermarking.SoBitChenBitThap(tmpData) ==
5&&ReversibleWatermarking.KiemtraChenBitThap(tmpData))
    {
        pos[count1] = (byte)i;
        count1++;
if (count1 > 1) break;
    }
    }

```

Аналогично, мы можем создать функции для размера 2\*2, 3\*3,4\*4...8\*8.

Далее показан код для отображения качества изображения:

```

for (inti = 0; i<image.Height; i++)
for (int j = 0; j < image.Width; j++)
    sum += (image.pixelMatrix[i][j] - basepixel[i * image.Width + j]) *
(image.pixelMatrix[i][j] - basepixel[i * image.Width + j]);
double MSE = sum / (image.Width * image.Height);
double PSNR = 20 * Math.Log10(255 / (Math.Sqrt(MSE)));

```

Для восстановления данных тоже нужно использовать две функции.

Если элементы не расширяемые, то мы получим бит b (bit) по формуле:

```

for (int i = 1; i < n; i++)
    {
        bit[i - 1] = (pixel[i] - pixel[0]) % 2;
        tempPixel[i] = (int)Math.Floor(((pixel[i] - pixel[0])*1.0) / 2);
    }

```

```
}
```

Если элементы расширяемые, то мы получим бит  $b$  по формуле:

```
Array.Sort(tempPixel);
int a = tempPixel[tempPixel.Count() / 2];
int index = 0;
for (int i = 0; i < pixel.Count(); i++)
if (Math.Abs(pixel[i] - a) <= 1) resultPixel[i] = pixel[i];
else
    {
        bit[index] = Math.Abs(pixel[i] - a) % 2;
    }
```

Наконец, следует сравнить значения. Если они не равны, то изображение изменилось.

## 2.4 Алгоритмы для защиты сообщения в изображении

### 2.4.1 Алгоритм СРТЕ

Алгоритм СРТЕ развивается из алгоритма СРТ. В алгоритме СРТ, количество скрытия информации в матрице  $F$  с размером:  $m*n$ :  $r$  (бит) где  $r = \lceil \log_2(mn+1) \rceil$ .

В алгоритме СРТЕ, количество скрытия информации в матрице  $F$  с размером:  $m*n$ :  $r^2+1$  (бит) где  $r = \lceil \log_2(mn) \rceil$

Но если  $mn = 2^r - 1 \Rightarrow r^2 + 1 = r$ , значит, алгоритмы СРТЕ и СРТ одинаковые. Алгоритм СРТЕ эффективен, чем СРТ при  $mn \neq 2^r - 1$ .

Предоставим бинарное изображение размером:  $m*n$ . В алгоритме используются 4 вида исходных данных:

- бинарная матрица  $F$ ;
- бинарная ключевая матрица  $K$ ;
- матрица весов  $W$  (значение элементов матрицы  $W$  принадлежат множеству  $\{1, 2, \dots, 2r\}$ , где  $r = \lceil \log_2(m*n) \rceil$ );
- последовательность битов  $b$  ( $r+1$  бит) которые необходимо скрыть в матрице  $F$

Содержание алгоритма:

1. Вычислим матрицу:  $T = F \text{ (xor) } K$  и  $r = \lceil \log_2(mn) \rceil$ .
2.  $S = \sum_{i=1}^m \sum_{j=1}^n T * M \text{ mod } 2^{r+1}$ . Получим:  $0 < S < 2^{r+1} - 1$ .
3. Скрытие информации  $b = b_1 b_2 \dots b_n$ . Биты равны 0 или 1,  $0 < b < 2^{r+1} - 1$ .

Поэтому необходимо инвертировать значение битов матрицы  $F$  (1->0 или 0->1), количество инверсий не более 2 раз, тогда получим:

$$S = \text{SUM}(T \text{ xor } W) \text{ mod } 2^{r+1} = b.$$

Для этого необходимо вычислить  $\alpha = b - \text{mod } 2^{r+1}$ . Если  $S$  равно  $b \text{ mod } 2^{r+1}$  (или  $\alpha = 0 \text{ mod } 2^{r+1}$ ) тогда инвертировать биты в матрице  $F$  не нужно. Если  $\alpha \neq 0$ , продолжаем далее.

4. Допустим  $S_\alpha$  – множество ячеек, хранящих скрытые биты. После перемещения некоторых битов в  $F$ , значение  $S$  увеличивает на  $\alpha$ .

Находим матрицу:

- Если  $1 \leq \alpha \leq 2^r$ , то

$$S_\alpha = \{F_{ij} \mid (T_{ij} = 0, W_{ij} = \alpha)\}$$

- Если  $2^r < \alpha \leq 2^{r+1} - 1$ , то

$$S_\alpha = \{(F_{ij}) \mid \{T_{ij} = 1, W_{ij} = 2^{r+1} - \alpha\}\}$$

- Если  $\alpha = 2^r$ , то

$$S_\alpha = \{(F_{ij}) \mid \{W_{ij} = \alpha\}\}$$

В процессе нахождения, будет 3 случая:

- Если  $S = b$ , то не надо перемещать местами биты в матрице.

- Если  $\alpha \neq 0$  и  $S_\alpha \neq \emptyset$ , то инвертируем бит на любое место и алгоритм

кончается.

- Если  $\alpha \neq 0$  и  $S_\alpha = \emptyset$ , необходимо инвертировать.

#### 2.4.2 Алгоритм МСРТЕ

Возникает задача улучшения качества изображения. Двоичное изображение содержит только элементы 0 и 1. Если изображение содержит только элементы 1 или 0, то инвертируем 0 на 1 или 0 на 1.

Идея алгоритма: необходимо разделить изображение на некоторых частях  $F_{m \times n}$ , тогда: если изображения содержит только элементы 1 или 0, то нельзя использовать скрыть информацию. Если новая матрица сама похожа с исходной матрицей. Например, матрица  $F$  превратилась в  $F'$  или  $F''$ , как показано на рисунке 2.

$$F = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad F' = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad F'' = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Рис.2. Матрица  $F$  превратилась в  $F'$  или  $F''$

Качество матрицы  $F'$  выше, чем  $F''$ , потому что в  $F''$  белый пиксел, значение которого равно 1, находится в окружении черных, равным 0.

Содержание алгоритма: предоставим бинарное изображение размером:  $m \times n$ . В алгоритме используются 4 вида исходных данных:

- бинарная матрица  $F$ ;
- бинарная ключевая матрица  $K$ ;
- матрица весов  $W \{W_{ij}, 1 \leq i \leq m, 1 \leq j \leq n\} = \{1, 2, \dots, 2^r\}$
- $r = \lceil \log_2(N) \rceil$ , где  $N = m \times n$
- последовательность битов  $b = b_1 b_2 \dots b_r$  (r бит), которые необходимо скрыть в матрице  $F$ .

Выходные данные: инвертируем значения битов матрицы  $F$  (1  $\rightarrow$  0 или 0  $\rightarrow$  1), количество инверсий не более 2 раз. Значит, мы получим новую матрицу  $F'$ .

$$S' = \text{SUM}[(F' \text{ xor } K) * W] \text{ mod } 2^{r+1}$$

$$\text{Тогда: } b_{r+1} = S' \text{ mod } 2 \text{ или } b_1 b_2 \dots b_r = \lfloor S'/2 \rfloor$$

Значит, мы уже используем последний бит для флага. Последний бит равно 0 и это значит, матрица содержит скрытые данные.

Добавим условия для матрицы весов  $W$ : в подматрица  $2 \times 2$  всегда существует не менее одогонечетного числа.

Для скрытия информации, необходимо выполнить несколько последовательных шагов:

1. Если  $F$  содержит только 0 или 1, то алгоритм закончивается. Если нет, вычислим матрицу:  $T = F \text{ (xor) } K$ .
2. Вычислим  $S = \sum_{i=1}^m \sum_{j=1}^n T * M \text{ mod } 2^{r+1}$ .
3. Сдвигаем 1бит влево, получим  $b' = b_1 b_2 \dots b_r 0$ . Значит,  $0 < b' < 2^{r+1} - 1$ .
  1.  $b'$  тоже последовательность  $r+1$  битов. Далее используем алгоритм СРТЕ для сокрытия  $b'$  в матрице  $F$ . Вычислим  $\alpha = b' - S \text{ mod } 2^{r+1}$ 
    - Если  $S = b' \text{ mod } 2^{r+1}$  тогда следует инвертировать биты в матрице  $F$  не нужно;
    - Если  $\alpha \neq 0$ , продолжаем.
4. Находим матрицу:
  - Если  $1 \leq \alpha \leq 2^r$ , то
 
$$S_\alpha = \{F_{ij} \mid (T_{ij} = 0, W_{ij} = \alpha \& d^2(F) \leq 2)\}$$
    - Если  $2^r < \alpha \leq 2^{r+1} - 1$ , то
 
$$S_\alpha = \{(F_{ij}) \mid \{T_{ij} = 1, W_{ij} = 2^{r+1} - \alpha \& d^2(F) \leq 2\}\}$$
      - Если  $\alpha = 2^r$ , то
 
$$S_\alpha = \{(F_{ij}) \mid \{W_{ij} = \alpha d^2(F) \leq 2\}\}$$

Необходимо добавить условия  $d^2(F) \leq 2$ , чтобы найти самую лучшую новую матрицу.

5. Если  $\alpha \neq 0$  и  $S_\alpha \neq 0$ , то инвертируем бит на любое место и алгоритм заканчивается;
6. Если  $\alpha \neq 0$  и  $S_\alpha = 0$ , необходимо найти целое и минимальное число  $h > 1$ , где  $S_{h\alpha} \neq 0$  и  $S_{\alpha-h\alpha} \neq 0$ . В этой случае, использовать алгоритм СРТЕ не следует, из-за неоднозначности факта нахождения числа

- h. Если  $h$  найдено, то инвертируем значения битов матрицы в  $S_{h\alpha}$  и  $S_{\alpha-h\alpha}$ .
7. Если  $h$  не найдено, необходимо отметить, что матрица не содержит информацию. Смотрим  $S = s_1s_2s_3\dots s_r s_{r+1}$ :
- Если  $S$  – нечетное число,  $s_{r+1}=1$ , то матрица не содержит информацию.
  - Если  $S$  – четное число, то инвертируем бит, мы получим необходимое число.

Для восстановления информации, необходимо выполнить ряд последовательных шагов:

1. Вычислим матрицу:  $T = F \text{ (xor) } K$ .
2. Вычислим  $S = \sum_{i=1}^m \sum_{j=1}^n T * M \text{ mod } 2^{r+1}$ . Если  $S$  – нечетное, то  $F$  не содержит информацию, алгоритм закончивается.
3. Если  $S$  – четное, то сдвигаем один бит влево, получим  $S' = s_1s_2\dots s_r = [S/2]$ .  $S$  – это сокрытые последовательные битов.

### 2.4.3 Функция ValAndNext

Для сокрытия информации в изображении 24 бита, мы используем алгоритм CRTE. Необходимо выбрать последний бит в каждой точке изображения. Далее получим бинарную матрицу. Алгоритмы CRTE и MCRTE очень эффективны для 24-х битовых изображений. Каждая точка изображения содержит 24-бита (8 бит описывают красный цвет, 8 бит – зеленый и 8 бит – синий). Значит, мы получим всего 3 бинарных матриц. Но из-за качества изображения 24-бита очень много, поэтому, он редко используется для сокрытия.

Алгоритмы CRTE и MCRTE очень эффективны для 8-х битовых изображений потому что, бит изображения описывает только один цвет. Поэтому, если мы изменим один бит, то его очень легко визуально определить.

Для сокрытия информации в изображении, необходимо создать функции valandnext.

Допустим P – это множество цветов. Значит, P содержит всего  $2^8=256$  элементы.  $P = \{x_1, x_2, \dots, x_{256}\}$ , где  $x = \{x_R, x_G, x_B\}$ . Смотрим формулу о расстоянии:

$$d(x,y) = \sqrt{(xR - yR)^2 + (xG - yG)^2 + (xB - yB)^2}$$

Необходимо создать следующий элемент:  $P \rightarrow P$

$$d(x,y) = \min \{d(x,i) \mid i \in P\}$$

и переменную:  $p \rightarrow E$  ( $E = \{0,1\}$ ,  $val(x) + val(next(x)) = 1$ ).

Например: В таблице цвет, выбрали один любой цвет. Допустим – это a1. Значение  $val(a1) = 0$ . Необходимо найти цвет, у которого есть минимальное расстояние для a1. Допустим – это a2. Значит, a2 – это следующий a1,  $val(a2) = 1$ . Аналогично,  $val(a3) = val(a2) + 1$ . Процесс заканчивается, если цвет повторяется (рис.3).

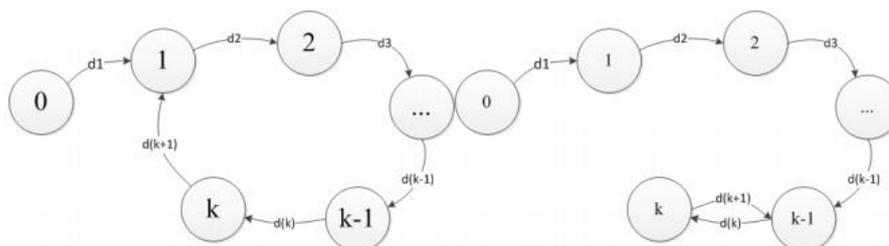


Рис.3. Алгоритм ValAndNext

Таким образом, свойства  $x$  и  $next(x)$  отличаются. Цель функции – инвертировать один бит и потом изменить бит A на следующий. Поэтому данные очень трудно обнаружить.

Далее представлена реализация алгоритмов на языке C#. Сначала необходимо создать класс CPTE. Создаем постоянную матрицу K, W (для

примера возьмем размер 8\*8), потом используем «StreamReader» и читаем значения.

```

for (int i = 0; i < 8; i++)
    {
        str = read.ReadLine();
for (int j = 0; j < 8; j++)
    {
        K[i, j] = Convert.ToInt16(str[2 * j].ToString());    }
    }

```

```

for (int i = 0; i < 8; i++)
    {
        str = read.ReadLine();
for (int j = 0; j < 8; j++)
    {
        W[i, j] = Convert.ToInt16(str.Substring(3 * j, 2));
    }
}
read.Dispose();
file.Dispose();

```

Далее вычислим матрицу:  $T = F \text{ (xor) } K$  ( $1 \text{ xor } 1 = 0, 0 \text{ xor } 0 = 0, 0 \text{ xor } 1 = 1, 1 \text{ xor } 0 = 0$ ) и  $S = \sum_{i=1}^m \sum_{j=1}^n T * M \bmod 2^{r+1}$ .

```

static int Xor(int x, int y)
    {
if (x == y) return 0;
elsereturn 1;
    }
for (int i = 0; i < 8; i++)
for (int j = 0; j < 8; j++)
    T[i, j] = Xor(F[i, j], K[i, j]);
    S = 0;
for (int i = 0; i < 8; i++)
for (int j = 0; j < 8; j++)
    S = S + T[i, j] * W[i, j];
    S = S % 128;

```

Проверим 3 случая  $x = 2^r$ . В этом шаге, мы проверим значения матрицы W и T. Результат возвращает значения матрицы S\_alpha.

```
if (a == x)
    {
for (int i = 0; i < 8; i++)
for (int j = 0; j < 8; j++)
if (W[i, j] == a)
        {
            empty = true;
return;
        }
    }
elseif (1 <= a && a < x)
    {
for (int i = 0; i < 8; i++)
for (int j = 0; j < 8; j++)
if (W[i, j] == a && T[i, j] == 0)
        {
            empty = true;
return;
        }
    }
elseif (x < a && a <= 2 * x - 1)
    {
for (int i = 0; i < 8; i++)
for (int j = 0; j < 8; j++)
if (W[i, j] == 2 * x - a && T[i, j] == 1)
        {
            empty = true;
return;
        }
    }
```

Если  $S\_alpha = 0$ , то инвертируем значения. Это простой процесс потому что, мы только инвертируем 1 на 0 или обратно. Кроме этого, необходимо создать функции для вычисления расстояния между цветами.

```
publicdouble KhoangCach(int i, int j)
```

```

    {
        return ((Red[i] - Red[j]) * (Red[i] - Red[j]) + (Green[i] - Green[j]) * (Green[i] - Green[j]) +
        (Blue[i] - Blue[j]) * (Blue[i] - Blue[j]));
    }

```

Далее создаем функции, которая возвращает цвет, у которого есть минимальное расстояние для цвета d. Если нет значения, то отметим значения -1 (значение val).

```

for (int i = 0; i < 256; i++)
    {
    if (i != d)
        {
        if (KhoangCach(i, d) <= min)
            {
            if (colorBinary[i] == -1) ret = i;
            elseif (KhoangCach(i, d) == min) ret = Math.Max(ret, -1);
            else ret = -1;
                min = KhoangCach(i, d);
            }
        }
    }
}

```

Итак, получили множество цветов, у которых есть значение -1. В этом случае, создаем функцию, которая возвращает цвет, у которого есть минимальное расстояние для цвета d.

```

for (int i = 0; i < 256; i++)
    {
    if (i != d)
        {
        if (colorBinary[i] != -2 && colorBinary[i] != -1)
            {
            if (KhoangCach(i, d) < min)
                {
                ret = i;
                min = KhoangCach(i, d);
            }
        }
    }
}

```

```
}
```

Для восстановления необходимо вычислить T и S.

```
for (int i = 0; i < 8; i++)
{
for (int j = 0; j < 8; j++)
{
T[i, j] = Xor(F_Mahoa[i, j], K[i, j]);
}
}
S = 0;
for (int i = 0; i < 8; i++)
for (int j = 0; j < 8; j++)
S = S + T[i, j] * W[i, j];
```

Итак, результат возвращается в виде текста (8 бит), поэтому необходимо изменить на тип «string».

```
for (int i = 0; i < chieudoc; i++)
for (int j = 0; j < chieungang; j++)
{
if (count > lap - 2) break;
count++;
for (int u = 0; u < 8; u++)
for (int v = 0; v < 8; v++) TG[u, v] = int.Parse(image[8 * i + u, 8 * j + v]);
strma += cpt.Giaima(TG);
}
MessageBox.Show("Текствизображения:\n" + strma);
string хауцои = "";
for (int i = 0; i < strma.Length / 7; i++)
{
хауцои += (char)BitToInt(strma.Substring(7 * i, 7));
}
}
```

Таким образом, алгоритм СРТЕ легче, чем алгоритм Alattar. Самое главное, это знать структуру изображения и обработать бит в строке.

## 3 РЕЗУЛЬТАТ РАБОТЫ

### 3.1 Результат защиты изображения ВМР

Вышеописанные теоретические положения были реализованы на практике в виде программного приложения «Watermarking Premium». Пользовательский интерфейс программы содержит закладки: «Обработка цифрового водяного знака».

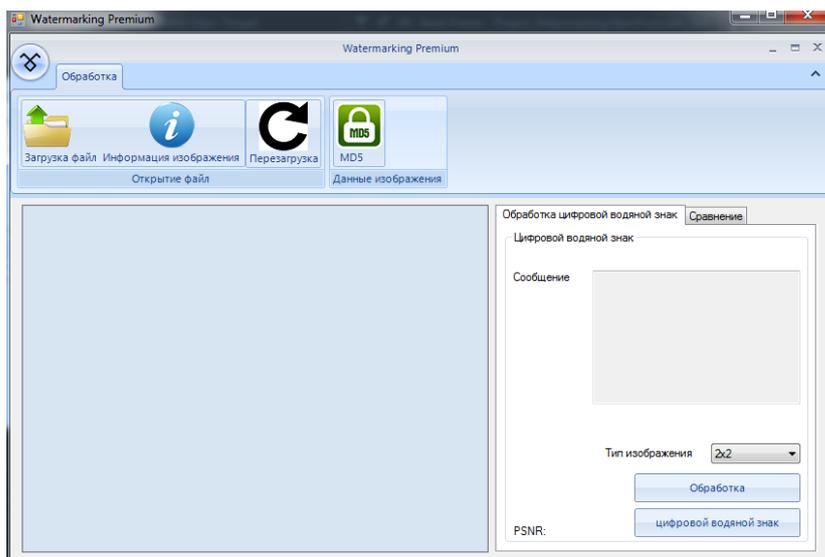


Рис.4. Интерфейс пользователя

Кнопка «Загрузка файл» укажет путь к файлу изображения. Выбрана текущая папка «AnhThuNghiem» и изображение по имени «baboon».

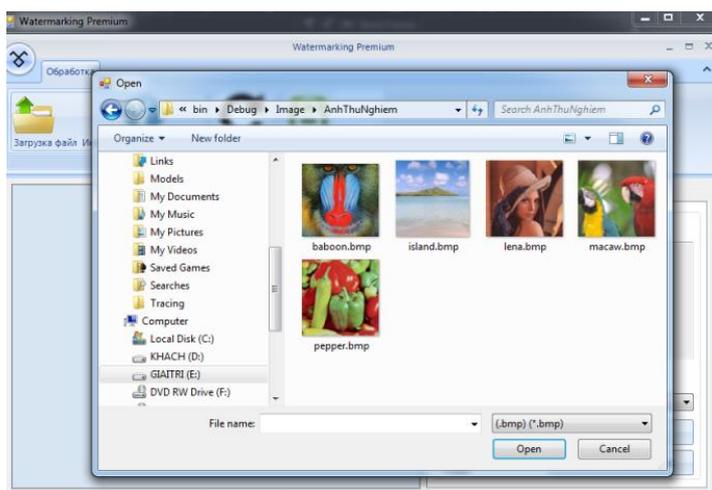


Рис.5. Папка изображения

Выбранное изображение показано в программе (рис.6).



Рис.6. Результат

После нажатия кнопки «MD5», результат шифрования появится на экране. Окно показывает цифровой знак изображения.

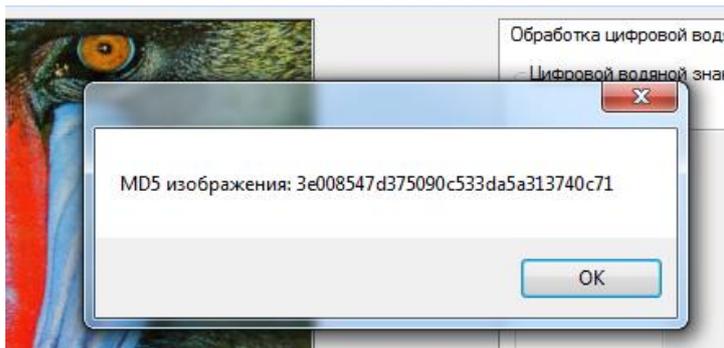


Рис.7. Информация об изображении

Далее необходимо разделить изображения на несколько частей. Для этого, следует выбрать значения в части «тип изображения» (2\*2, 3\*3, 4\*4, 5\*5). Для проверки, выбрано значение 3\*3.

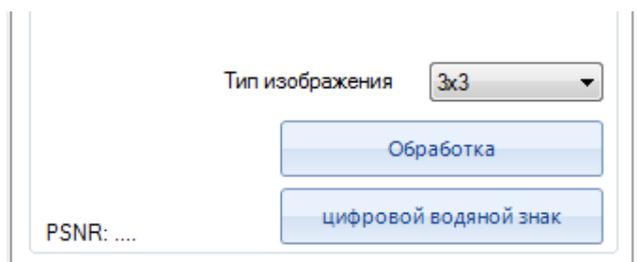


Рис.8. Процесс «цифровой водяной знак»

Нажив кнопку «Цифровой водяной знак», результат получен на рисунке 7. Качество оценивается правильно через коэффициента PSNR (PeakSignaltoNoiseRatio).

$$PSNR = 10\log_{10}\left(\frac{MAX^2}{MSE}\right)$$

MAX: максимальное значение каждого пикселя.

$$MSE = \frac{1}{m.n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$



Рис.9.Качество изображения

Новое изображение содержит цифровой водяной знак, называется «WaterMarking\_3x3\_baboon» и находится в новой папке «Anhthuyvan».

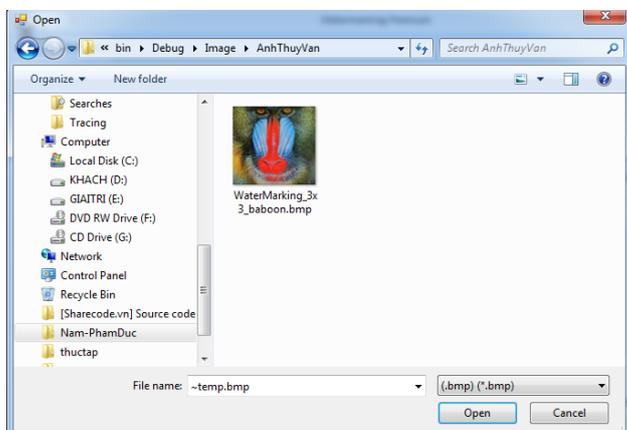


Рис.10. Новая папка и новое имя изображения

Визуально оба изображения выглядят одинаково.



Рис.11. До и после преобразования

В рамках имитации атаки на изображение используем программу «Paint» для редактирования изображения «WaterMarking\_3x3\_baboon», которое содержит цифровой водяной знак. Результат проверки атакованного изображения показан на рисунке 12.

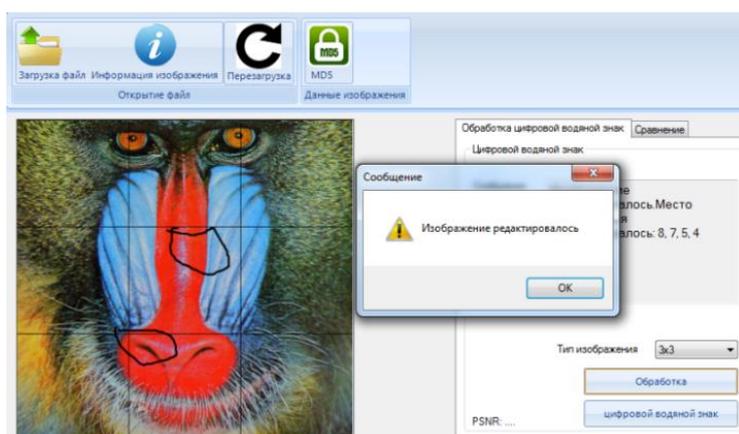


Рис.12. Результат работы

Измененное место изображения отмечено красным. Появится сообщение с указанием измененного участка. Участки 8,7,5,4 уже атаковались.



Рис.13. Результат проверки редактированного изображения

Если мы редактируем исходное изображение, то результат будет не верным, т.к. оно не содержит цифровой водяной знак.

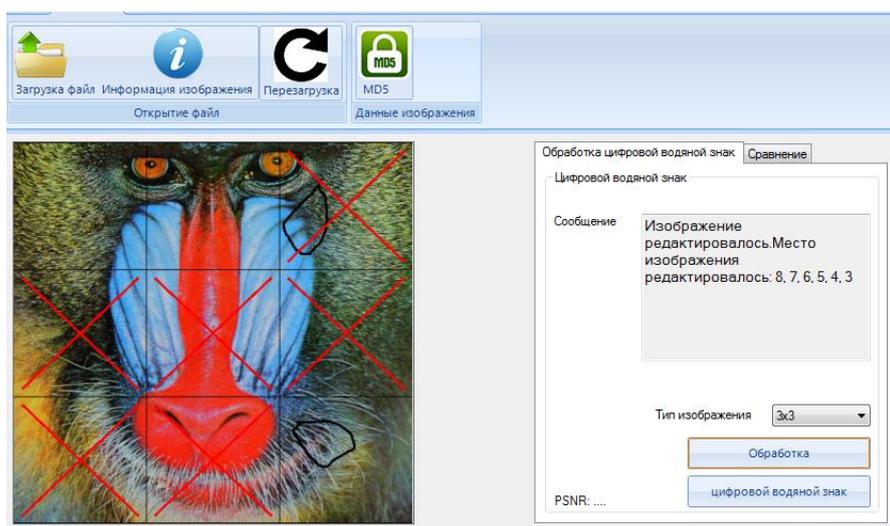


Рис.14. Результат проверки редактированного изображения

### 3.2 Результат защиты сообщения в изображениях

Вышеописанные теоретические положения были реализованы на практике в виде программного приложения «Скрытие информации в изображениях».

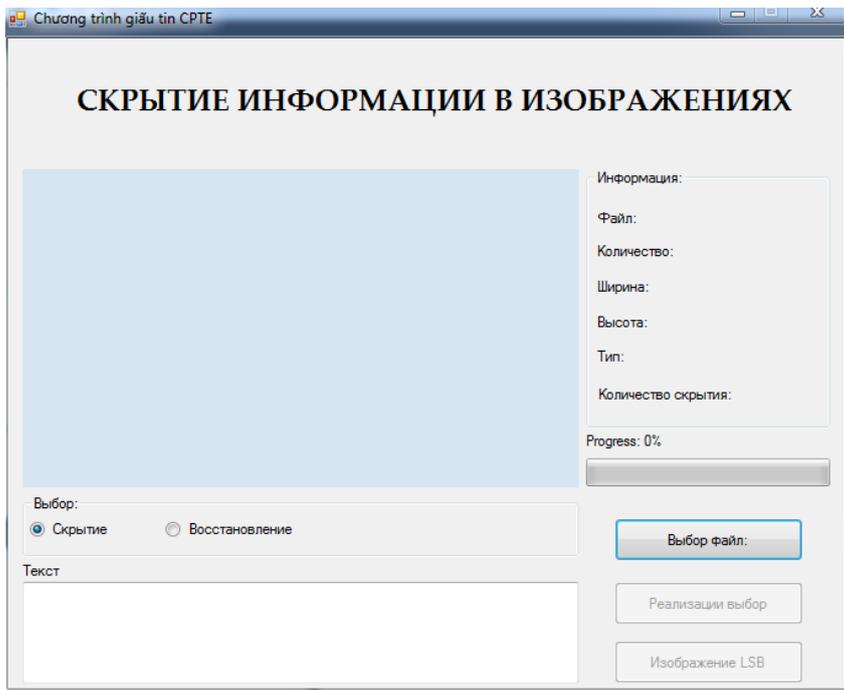


Рис.15. Пользовательский интерфейс

Пользовательский интерфейс программы позволяет скрыть, восстановить информации и прочитатть данные изображения.

Выбранное изображение показано в программе (рис. 16). Видно, что информация об изображении появится на экране. Сейчас, мы можем скрыть текст в изображении.

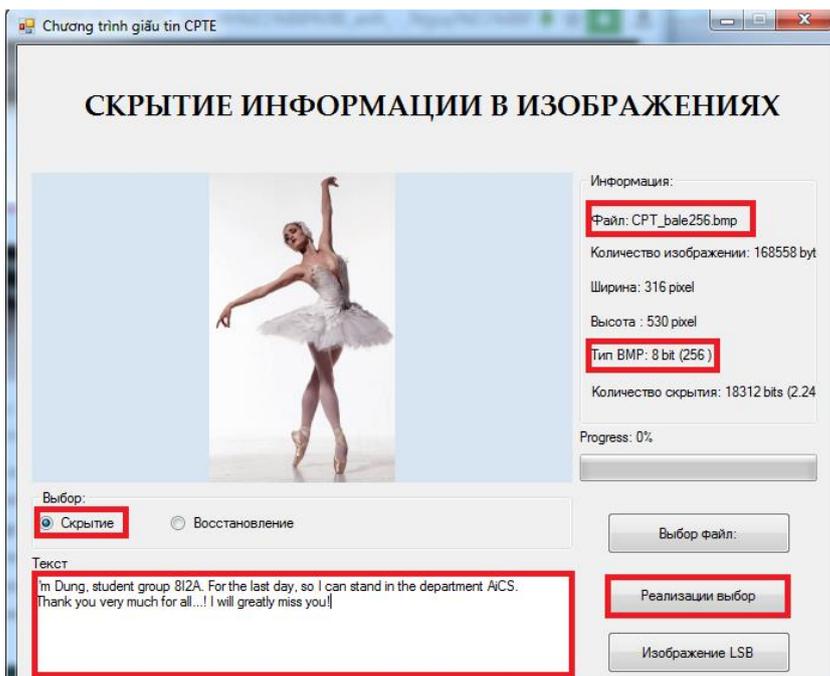


Рис.16. Информация об изображении

Текущая папка содержит всего 8 изображений.

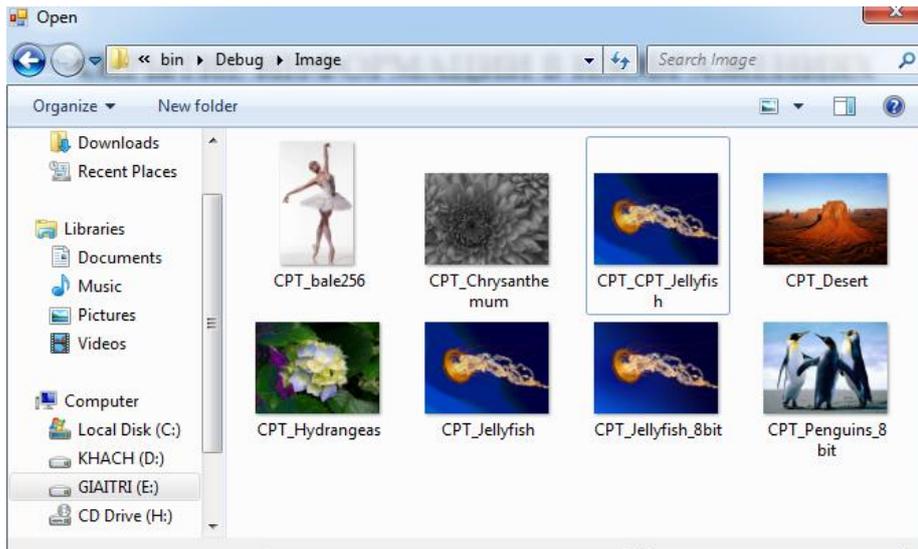


Рис.17. Выбор изображения из текущей папки

Выбрав «Скрытие» и нажав кнопку «Реализация выбора», получим результат, который представлен на рисунке 18:

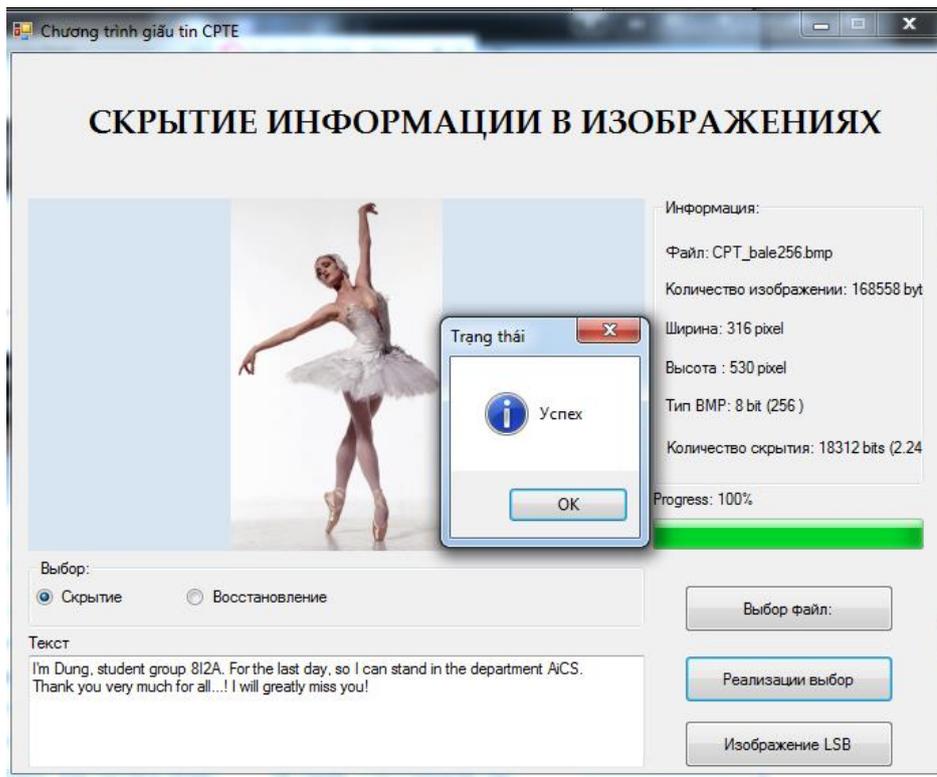


Рис.18. Результат работы

Новое изображение сохранено в текущей папке (рис. 19).

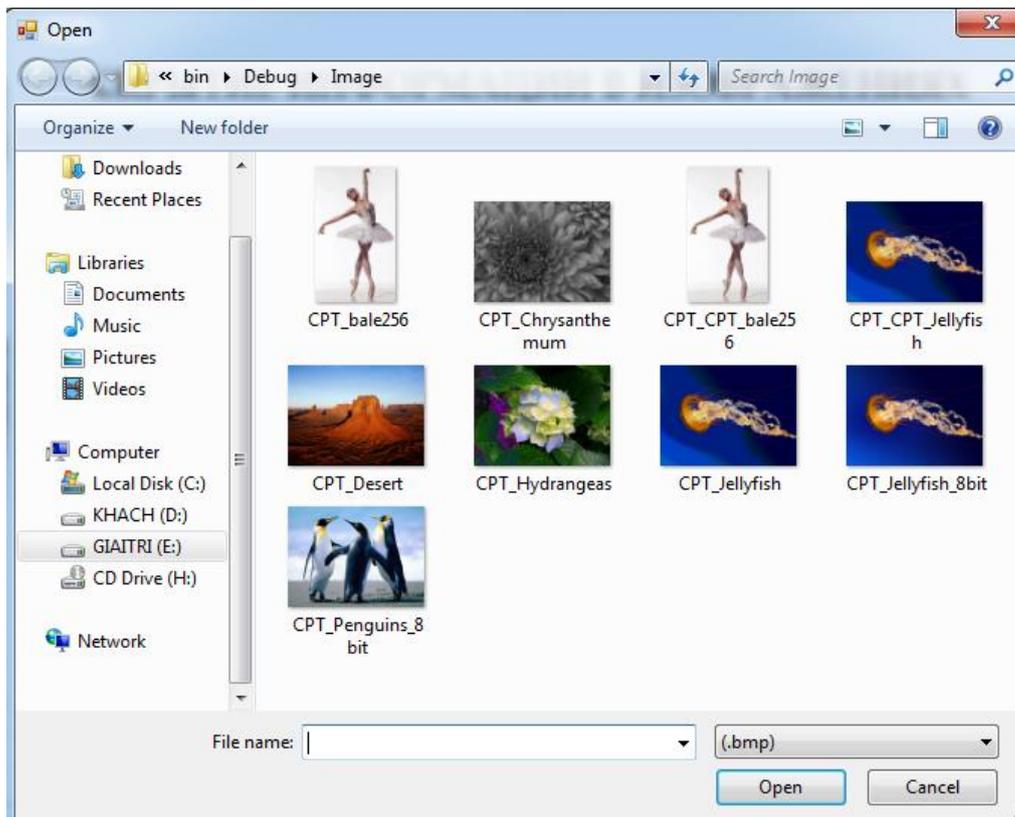


Рис.19. Текущая папка

Новое изображение называется «CPT\_CPT\_bale256». Далее восстановим информацию из изображения, выбрав «Восстановление» и нажав кнопку «Реализация выбора».

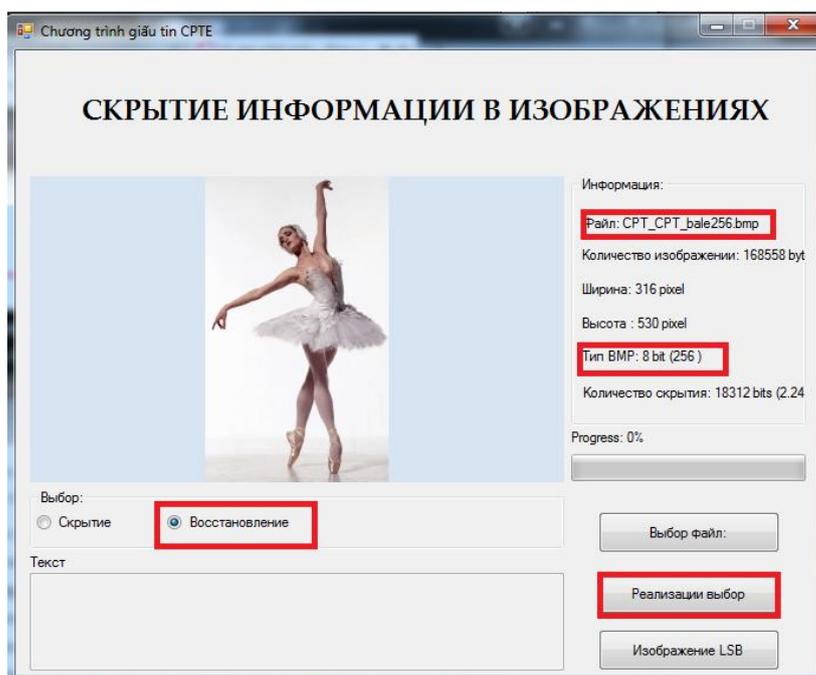


Рис.20. Восстановление изображения

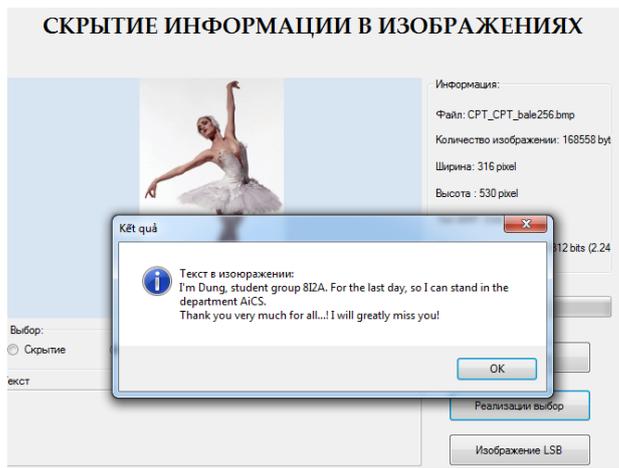


Рис.21. Отображение скрытого текста в изображении

Визуально оба изображения выглядят одинаково.



Рис.22. До и после восстановления



Рис.23. До и после восстановления

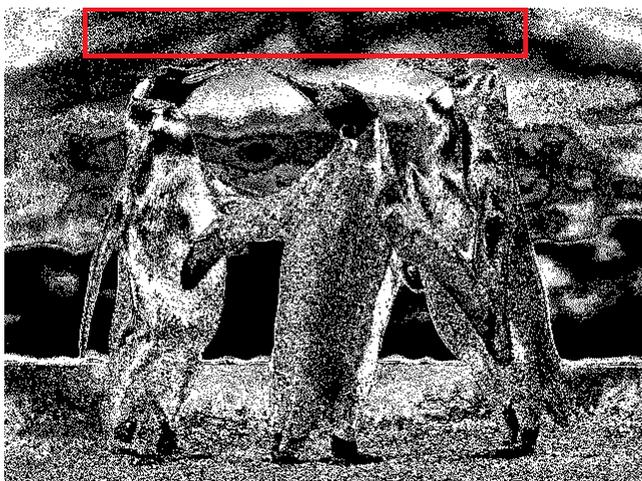


Рис.24. Довосстановления

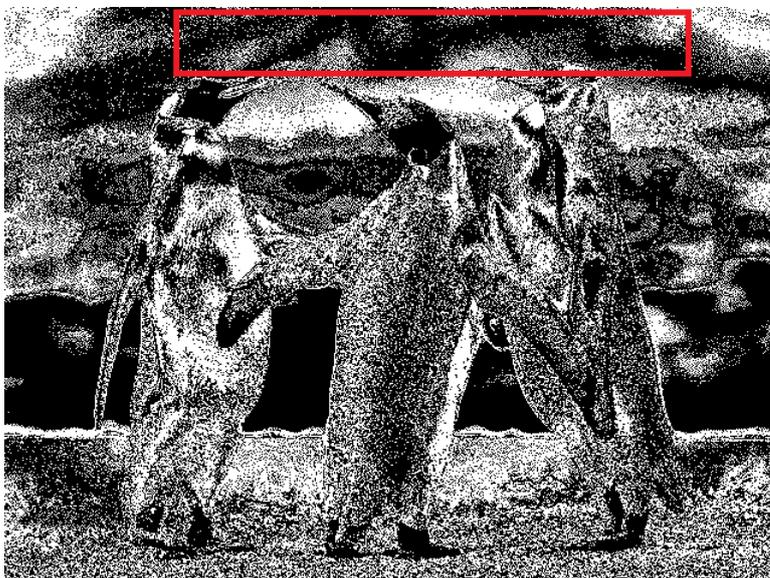


Рис.25. После восстановления

Итак, данная работа на текущем этапе носит характер исследовательской и может быть полезна студентам при изучении алгоритмов работ с изображениями и защиты данных. В дальнейшем планируется продолжить работу по направлению исследования алгоритмов скрытия информации не только в изображении, но и в видео или в звуке.

## 4 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ

### 4.1. Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

#### 4.1.1. Потенциальные потребители результатов исследования

Результат исследования - это не конкретная продукция для продажи конечным потребителям. Данная работа является исследованием способа сокрытия информации в изображениях формата BMP. Основным исполнителем работы является программист. Объектом для исследования работы является изображение BMP. Работа содержит 7 различных алгоритмов. Поскольку исследование не дает коммерческой продукции, не существует рынка для его результата. Потребителями могут являться студенты, которых интересуют анализ и обработка изображений BMP. Так же результат данного исследования полезен для людей, начинающих изучать сокрытия информации в изображениях формата BMP в среде VisualStudio 2012.

		Вид интернет-ресурса			
		Корпоративный сайт	Интернет-каталог	Интернет-магазин	Информационный портал
Размер обучения	Крупные		=====	//////	
	Средние		=====	//////	
	Мелкие		=====	//////	

Рис.24. Карта сегментирования рынка услуг по разработке интернет-ресурсов:

- ||||| Компании «ΠΙΕΤΟΟΣΚΑΡΙΟΥ» ;
- ===== Фирма РКК;
- ////// Компании ЗАО "ЗИПО".

#### 4.1.2. Анализ конкурентных технических решений

Анализ конкурентных технических решений определяется по формуле:

$$K \leftarrow \sum B \cdot B$$

где  $K$  – конкурентоспособность научной разработки или конкурента;

$V_i$  – вес показателя (в долях единицы);

$B_i$  – балл  $i$ -го показателя.

У результата исследования нет конкурентов. На таблице 1 приведен список критериев, которые позволяют оценить результат исследования. На основе этой таблицы можно сравнить результат данного исследования с результатами других работ.

Таблица 5

Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		$B_{\phi}$	$B_{k1}$	$B_{k2}$	$K_{\phi}$	$K_{k1}$	$K_{k2}$
1	2	3	4	5	6	7	8
Технические критерии оценки ресурсоэффективности							
1. Количество алгоритмов	0.1	4	4	5	0,4	0,4	0,5
2. Уровень детализации	0.15	5	4	4	0,75	0,6	0,6
3. Количество реализаций	0.1	4	3	4	0,4	0,3	0,4
4. Качество примеров, реализаций	0.15	4	5	3	0,6	0,75	0,45
5. Качество сравнительного анализа	0.3	4	4	4	1,2	1,2	1,2
6. Потребность в ресурсах	0.1	5	5	4	0,5	0,5	0,4
7. Потребность в времени	0.1	4	4	5	0,4	0,4	0,5
	1				4,25	4,15	4,05

По данным из таблицы 5, фактический результат нашего исследования получает 4.25 баллов из 5. Это хорошая оценка.

#### 4.1.3 Технология QuaD

Технология QuaD (QUality ADvisor) представляет собой гибкий инструмент измерения характеристик, описывающих качество новой разработки и ее перспективность на рынке и позволяющие принимать решение целесообразности вложения денежных средств в научно-исследовательский проект.

Для применения технологии QuaD используются таблица 6, приведённая далее.

Таблица 6 -Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Баллы	Максимальный балл	Относительное значение (3/4)	Средневзвешенное значение (5x2)
Количество алгоритмов	0.1	80	100	0,8	0,08
Уровень детализации	0.15	90	100	0,9	0,135
Количество реализаций	0.1	80	100	0,8	0,08
Качество примеров, реализаций	0.15	80	100	0,8	0,12
Качество сравнительного анализа	0.3	80	100	0,8	0,24
Потребность в ресурсах	0.1	90	100	0,9	0,09
Потребность в времени	0.1	75	100	0,75	0,075
<b>Итого</b>	<b>1</b>	<b>575</b>	<b>700</b>		<b>0,82</b>

Из таблицы **Error! Reference source not found.** получены результаты, дающие оценку качеству разработки. По этой оценки следует, что качество выше среднего.

#### 4.1.4 SWOT-анализ

**SWOT** – Strengths (сильные стороны), Weaknesses (слабые стороны), Opportunities (возможности) и Threats (угрозы) – представляет собой комплексный анализ научно-исследовательского проекта. SWOT-анализ применяют для исследования внешней и внутренней среды проекта.

Для SWOT анализа построена таблица 7.

Таблица 7 – SWOT анализ

	Сильные стороны научно-исследовательского проекта:	Слабые стороны научно-исследовательского проекта:
	С1 Присутствие большего количества алгоритмов	Сл1 Низкая скорость
	С2 Высокая степень детализаций	Сл2 Отсутствие некоторых алгоритмов
	С3 Хорошее качество реализации	Сл3 Качество сравнительного анализа
	С4 Низкая потребность в ресурсах	

<b>Возможности:</b> B1 Появление спроса на результат исследования B2 Внедрение результатов исследования в разработки программ	Появление спроса на результат исследования за счет высокой степени детализаций и хорошего качества реализации Внедрение результатов исследования в разработки программ за счет хорошего качества реализации	Широкое распространение результатов исследований Привлечение новых сотрудников для увеличения скорости исследования. Привлечение специалистов для повышения качества анализа
---	--	--

#### 4.2. Определение возможных альтернатив проведения научных исследований

Морфологическая матрица для программы представлена на таблице 8.

Таблица 8 - Морфологическая матрица для программы

	1	2	3
А. Прибор	Ноутбук	Компьютер	
Б. Программа	Visual Studio 2010	Visual Studio 2012	Visual Studio 2013
В. Програмирование язык	Java	C#	
Г. Алгоритм для выполнения задачу	CPT	CPTE	MCPTЕ
Д. Средство для сравнения качество алгоритмов	Время	Количество битов	

Наиболее желательных функционально конкретных решений:  
A1B2B2Г2Д2.

#### 4.3 Планирование научно-исследовательских работ

##### 4.3.1 Структура работ в рамках научного исследования

В данном разделе необходимо составить перечень этапов и работ в рамках проведения научного исследования, провести распределение исполнителей по видам работ. Исполнителями выбраны студент и научный руководитель.

Таблица 9 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Определение темы исследований	1	Выбор направления исследований	Руководитель
	2	Подбор и изучение материалов по выбранному направлению	студент
	3	Определение окончательной темы исследования	Руководитель, студент

	4	Определение задачи для исследования	Руководитель
	5	Календарное планирование работ по теме	Студент
<i>Проведение исследования</i>			
Теоретическое исследование	6	Подбор материалов по выбранной теме	Студент
	7	Изучение математических основ существующих алгоритмов сокрытия информации	Студент
	8	Сравнение алгоритмов	Студент
	9	Установка Visual Studio 2012	Студент
Практическое исследование	10	Поиск существующих реализаций алгоритмов	Студент
	11	Тестирование работы	Студент
	12	Сравнение алгоритмов	Студент
Утверждение результата исследования	13	Проверка результатов исследования	Руководитель
	14	Исправить ошибки	Студент
Оформление отчёта по НИР	15	Составление пояснительной записки	Студент
	16	Публикация работы	Студент

#### 4.3.2. Определение трудоемкости выполнения работ

Трудовые затраты в большинстве случаев образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоемкости работ каждого из участников научного исследования.

Трудоемкость выполнения научного исследования оценивается экспертным путем в человеко-днях и носит вероятностный характер, т.к. зависит от множества трудно учитываемых факторов. Для определения ожидаемого (среднего) значения трудоемкости  $t_{ожі}$  используется следующая формула:

$$t_{ожі} = \frac{3t_{\min i} + 2t_{\max i}}{5},$$

где  $t_{ожі}$  – ожидаемая трудоемкость выполнения  $i$ -ой работы чел.-дн.;

$t_{\min i}$  – минимально возможная трудоемкость выполнения заданной  $i$ -ой работы (оптимистическая оценка: в предположении наиболее благоприятного стечения обстоятельств), чел.-дн.;

$t_{\max i}$  – максимально возможная трудоемкость выполнения заданной  $i$ -ой работы (пессимистическая оценка: в предположении наиболее неблагоприятного стечения обстоятельств), чел.-дн.

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях  $T_{pi}$ , учитывающая параллельность выполнения работ несколькими исполнителями. Такое вычисление необходимо для обоснованного расчета заработной платы, так как удельный вес зарплаты в общей сметной стоимости научных исследований составляет около 65 %.

$$T_{pi} = \frac{t_{ожi}}{Ч_i},$$

где  $T_{pi}$  – продолжительность одной работы, раб.дн.;

$t_{ожi}$  – ожидаемая трудоемкость выполнения одной работы, чел.-дн.

$Ч_i$  – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

### 4.3.3 Разработка графика проведения научного исследования

Для разработки графика проведения научного исследования используется диаграмма Ганта.

Диаграмма Ганта – горизонтальный ленточный график, на котором работы по теме представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения данных работ.

Для удобства построения графика, длительность каждого из этапов работ из рабочих дней следует перевести в календарные дни. Для этого используется формула:

$$T_{ki} = T_{pi} \cdot k_{\text{кал}}, \quad (5)$$

где  $T_{ki}$  – продолжительность выполнения  $i$ -й работы в календарных днях;

$T_{pi}$  – продолжительность выполнения  $i$ -й работы в рабочих днях;

$k_{\text{кал}}$  – коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}}, \quad (6)$$

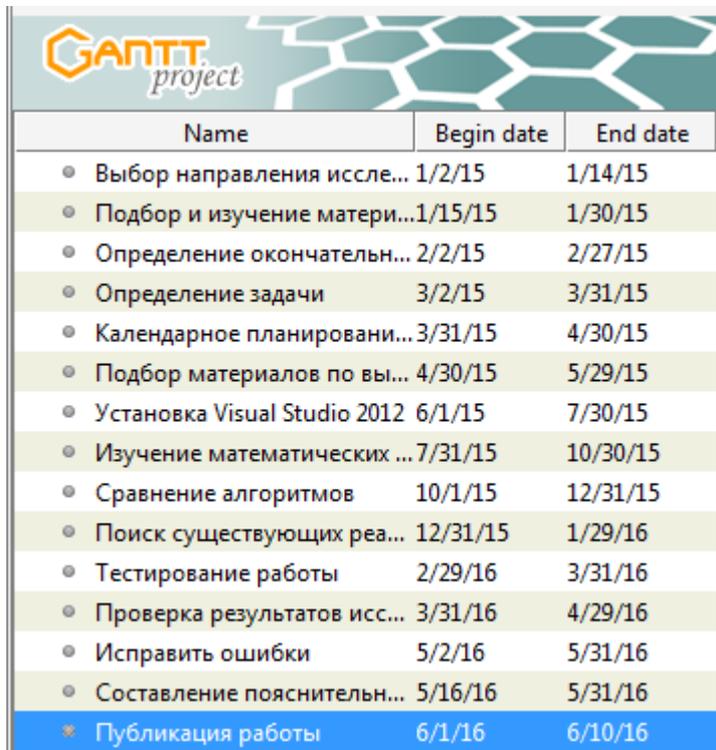
где  $T_{\text{кал}}$  – количество календарных дней в году;

$T_{\text{вых}}$  – количество выходных дней в году;

$T_{\text{пр}}$  – количество праздничных дней в году.

Для 2016 года количество календарных дней - 366, рабочих - 247, выходных - 119, праздничных - 14. Поэтому

$$k_{\text{кал}} = \frac{366}{366 - 119 - 14} = 1.57$$



Name	Begin date	End date
• Выбор направления иссле...	1/2/15	1/14/15
• Подбор и изучение матери...	1/15/15	1/30/15
• Определение окончательн...	2/2/15	2/27/15
• Определение задачи	3/2/15	3/31/15
• Календарное планировани...	3/31/15	4/30/15
• Подбор материалов по вы...	4/30/15	5/29/15
• Установка Visual Studio 2012	6/1/15	7/30/15
• Изучение математических ...	7/31/15	10/30/15
• Сравнение алгоритмов	10/1/15	12/31/15
• Поиск существующих реа...	12/31/15	1/29/16
• Тестирование работы	2/29/16	3/31/16
• Проверка результатов исс...	3/31/16	4/29/16
• Исправить ошибки	5/2/16	5/31/16
• Составление пояснительн...	5/16/16	5/31/16
• Публикация работы	6/1/16	6/10/16

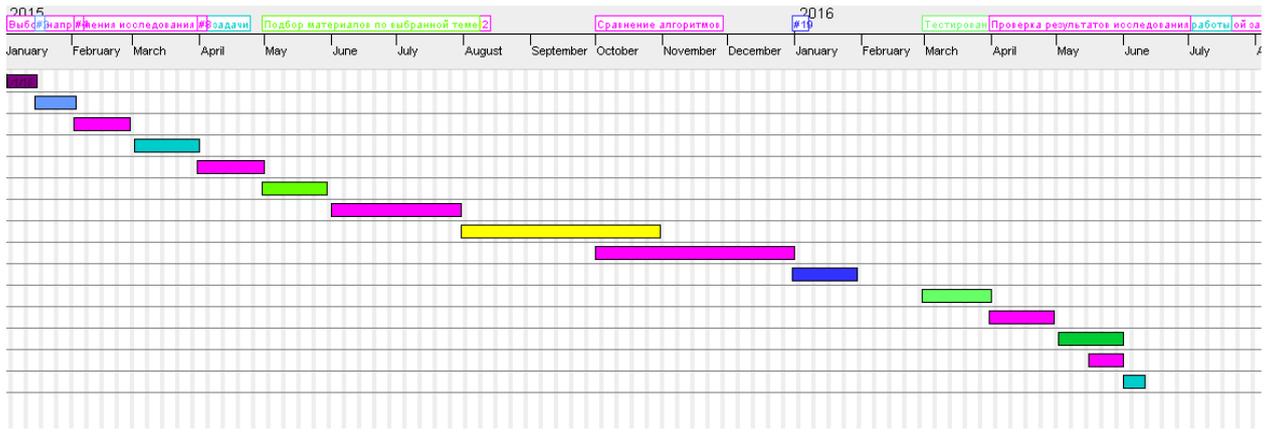


Таблица 10 – Временные показатели проведения научного исследования

Название работы	Трудоёмкость работ									Исполнители	Длительность работ в рабочих днях, $T_{pi}$			Длительность работ календарных дней, $T_{ki}$		
	$t_{min}$ , чел-дни			$t_{max}$ , чел-дни			$t_{ожі}$ , чел-дни				Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3							
Выбор направления исследования	2	2	2	2	3	4	2	2.4	2.8	Руководитель	2	2.4	2.8	3	4	5
Подбор и изучение материалов по выбранному направлению	15	15	15	16	18	16	15	16	15.4	Студент	15.4	16.2	15.4	23	24	23
Определение окончательной темы исследования	2	2	2	2	2	3	2	2	2.4	Руководитель	2	2	2.4	3	3	4
Определение задачи	1	1	1	1	2	2	1	1.4	1.4	Руководитель	1	1.4	1.4	2	3	3
Календарное планирование работе по теме	1	1	1	1	1	2	1	1	1.4	Студент	1	1	1.4	2	2	3
Подбор материалов по выбранной теме	10	10	10	12	12	12	11	11	10.8	Студент	10.8	10.8	10.8	16	16	16
Установка Visual Studio 2012	1	1	1	1	2	1	1	1.4	1	Студент	1	1.4	1	2	3	2
Изучение математических основ существующих алгоритмов сокрытия информации	17	17	17	19	21	20	18	19	18.2	Студент	17.8	18.6	18.2	27	28	27

Сравнение алгоритмов	7	7	7	9	10	9	7.8	8.2	7.8	Студент	7.8	8.2	7.8	12	13	12
Поиск существующих реализаций алгоритмов	7	7	7	10	8	9	8.2	7.4	7.8	Студент	8.2	7.4	7.8	13	11	12
Тестирование работы	5	5	5	7	6	7	5.8	5.4	5.8	Студент	5.8	5.4	5.8	9	8	9
Сравнение алгоритмов	9	9	9	11	12	11	9.8	10	9.8	Студент	9.8	10.2	9.8	15	16	15
Проверка результатов исследования	2	2	2	3	3	4	2.4	2.4	2.8	Руководитель	2.4	2.4	2.8	4	4	5
Исправить ошибки	5	5	5	5	6	7	5	5.4	5.8	Студент	5	5.4	5.8	8	8	9
Составление пояснительной записки	10	10	10	12	12	12	11	11	10.8	Студент	10.8	10.8	10.8	16	16	16
Публикация работы	1	1	1	1	1	1	1	1	1	Студент	1	1	1	2	2	2
Итого											102	104.6	105	157	161	163

#### 4.3.4 Бюджет научно-технического исследования (НТИ)

В данном разделе рассмотрены расходы, осуществлённые при разработке информационной системы.

##### 4.3.4.1 Расчет материальных затрат НТИ

Данная статья включает стоимость всех материалов, используемых при разработке проекта.

Таблица 11 – Материальные затраты

Наименование	Единица измерения	Количество			Цена за ед., руб.			Затраты на материалы, руб.		
		Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
ПК	Шт.				2400	6000	4200	2400	6000	4200
Итого								2400	6000	4200

В процессе работы использовался персональный компьютер (ПК). Самым дешёвым оказался вариант первого исполнения, самым дорогим – второго.

##### 4.3.4.2 Расчет затрат на специальное оборудование для научных (экспериментальных) работ

Работа используется только персональный компьютер. Поэтому, в этом пункте необходимо рассчитывать сумму амортизации. Эти затраты представлены в таблице. Амортизационные отчисления рассчитываются по формуле:

$$Z_{ам} = \frac{(C_i \cdot H_a)}{100\%},$$

где  $Z_{ам}$  – ежемесячная сумма амортизационных отчислений;

$C_i$  – цена (балансовая стоимость)  $i$ -го оборудования;

$H_a$  – норма амортизационных отчислений.

Срок полезного использования персонального компьютера - 1095 дней. Следует, что норма амортизационных отчислений равна:

$$N_a = \frac{1}{1095} * 100\% = 0.09132\%$$

Таблица 12 – Величина амортизационных отчислений

Наим.о борудо вания	Кол- во	С первонач., руб			А в мес., руб.			А за период, руб.		
		Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Компьютер	1				1161,6	1260,2	1210,9	6079,2	6763,4	6579,5
		2400	6000	4200						
Итого								<b>6079,2</b>	<b>6763,4</b>	<b>6579,5</b>

Стоимость покупки компьютера во втором варианте исполнения больше по сравнению с остальными вариантами, следует, что общая стоимость амортизации тоже наибольшая.

#### 4.3.4.3 Основная заработная плата исполнителей темы

Месячный должностной оклад работника:

$$Z_m = Z_{тс} \cdot (1 + k_{пр} + k_d) \cdot k_p,$$

где  $Z_{тс}$  – заработная плата по тарифной ставке, руб.;

$k_{пр}$  – премиальный коэффициент, равный 0,3 (т.е. 30% от  $Z_{тс}$ );

$k_d$  – коэффициент доплат и надбавок составляет примерно 0,2 – 0,5 (в НИИ и на промышленных предприятиях – за расширение сфер обслуживания, за профессиональное мастерство, за вредные условия: 15-20% от  $Z_{тс}$ );

$k_p$  – районный коэффициент, равный 1,3 (для Томска).

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{дн} = \frac{Z_m \cdot M}{F_d},$$

где  $Z_m$  – месячный должностной оклад работника, руб.;

$M$  – количество месяцев работы без отпуска в течение года:

- при отпуске в 24 раб.дня  $M = 11,2$  месяца, 5-дневная неделя;

- при отпуске в 48 раб.дней  $M=10,4$  месяца, 6-дневная неделя;

$F_d$  – действительный годовой фонд рабочего времени научно-технического персонала, раб.дн.

Таблица 13–Баланс рабочего времени

Показатели рабочего времени	Руководитель	Лаборант
Календарное число дней	366	366
Количество нерабочих дней	52+14=66	119+14=133
- выходные дни		
- праздничные дни		
Потери рабочего времени	48+24=72	24+20=44
- отпуск		
- невыходы по болезни		
Действительный годовой фонд рабочего времени	228	189

Основная заработная плата ( $Z_{осн}$ ) руководителя (лаборанта) рассчитывается по следующей формуле:

$$Z_{осн} = Z_{дн} \cdot T_p,$$

где  $Z_{осн}$  – основная заработная плата одного работника;

$T_p$ – продолжительность работ, выполняемых научно-техническим работником, раб. дн. (табл. 8);

$Z_{дн}$  – среднедневная заработная плата работника, руб.

Таблица 14 –Расчёт основной заработной платы

Исполнители 1	Разряд	$k_t$	$Z_{тс}$ , руб.	$k_{пр}$	$k_d$	$k_p$	$Z_m$ , руб	$Z_{дн}$ , руб.	$T_p$ , раб.дн.	$Z_{осн}$ , руб.
Руководитель	Проверить работу	600	15000	0,3	0,4	1,3	33150	1512,1	10	15121
Студент	Выполнить работу	600	8000	0,3	0,4	1,3	17680	1047,7	90	94293
Итого										109414

Исполнители 2	Разряд	$k_t$	$Z_{тс}$ , руб.	$k_{пр}$	$k_d$	$k_p$	$Z_m$ , руб	$Z_{дн}$ , руб.	$T_p$ , раб.дн.	$Z_{осн}$ , руб.
Руководитель	Проверить работу	600	15000	0,3	0,4	1,3	33150	1512,1	12	18145
Студент	Выполнить работу	600	8000	0,3	0,4	1,3	17680	1047,7	92	96388
Итого										114533

Исполнители 3	Разряд	$k_t$	$Z_{тс}$ , руб.	$k_{пр}$	$k_d$	$k_p$	$Z_m$ , руб	$Z_{дн}$ , руб.	$T_p$ , раб.дн.	$Z_{осн}$ , руб.
Руководитель	Проверить работу	600	15000	0,3	0,4	1,3	33150	1512,1	9	13608
Студент	Выполнить работу	600	8000	0,3	0,4	1,3	17680	1047,7	95	99531
Итого										113139

Из таблицы видно, что заработные платы в трех вариантах исполнения не сильно отличаются друг от друга. Заработная плата второго варианта исполнения самая большая.

#### 4.3.4.4 Дополнительная заработная плата исполнителей темы

Для дополнительной заработной платы используется коэффициент дополнительной заработной платы ( $k_{доп.}$ ), равный 0,15.

Расчет дополнительной заработной платы ведется по следующей формуле:

$$Z_{доп} = k_{доп} \cdot Z_{осн}$$

Результаты расчётов приведены в таблице 10.

Таблица 15 – Расчет дополнительной заработной платы

Исполнитель	Основная заработная плата, руб.			$k_{доп.}$	Дополнительная заработная плата, руб.		
	Исп.1	Исп.2	Исп.3		Исп.1	Исп.2	Исп.3
Руководитель	15121	18145	13608	0,15	2721,1	2721,75	2041,2
Студент	94293	96388	99531	0,15	14143,95	14458,2	14929,6
Итого					<b>16865,05</b>	<b>17179,95</b>	<b>16970,8</b>

В данной таблице ситуация аналогична основной заработной плате, то есть самым дорогим остаётся второй вариант.

#### 4.3.4.5 Отчисления во внебюджетные фонды (страховые отчисления)

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования (ФСС), пенсионного фонда (ПФ)

и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$Z_{\text{внеб}} = k_{\text{внеб}} \cdot (Z_{\text{осн}} + Z_{\text{доп}}),$$

где  $k_{\text{внеб}}$  – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

На 2014г. в соответствии с Федеральным законом от 24.07.2009 №212-ФЗ установлен размер страховых взносов равный 30%.

Таблица 1 – Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.			Дополнительная заработная плата, руб.		
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Руководитель	15121	18145	13608	2721,1	2721,75	2041,2
Студент	94293	96388	99531	14143,95	14458,2	14929,6
Коэффициент отчислений во внебюджетные фонды	0,3					
Итого						
Исполнение 1	<b>37883,715</b>					
Исполнение 2	<b>39513,885</b>					
Исполнение 3	<b>39032,94</b>					

#### 4.3.4.6 Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергии, почтовые и телеграфные расходы, размножение материалов и т.д. Их величина определяется по следующей формуле:

$$Z_{\text{накл}} = (\text{сумма статей } 1 \div 7) \cdot k_{\text{нр}}$$

где  $k_{\text{нр}}$  – коэффициент, учитывающий накладные расходы.

Величина коэффициента накладных расходов взята в размере 16%.

$$\text{Исп.1} = (42400+6079,2+109414+16865,05+37883,715)*0,16 = 34022,7$$

$$\text{Исп.2} = (46000+6579,5+114533+17179,95+39513,885)*0,16 = 36131,2$$

$$\text{Исп.3} = (44200+6579,5+113139+16970,8+39032,94)*0,16 = 35187,5$$

#### 4.3.4.7 Формирование бюджета затрат научно-исследовательского проекта

В данном пункте определяется конечный бюджет, основанный на расчётах в предыдущих пунктах. Бюджет приведён далее, в таблице 17.

Таблица 17 – Расчет бюджета затрат НИИ

Наименование статьи	Сумма, руб.		
	Исп.1	Исп.2	Исп.3
Материальные затраты НИИ	424	4600	4420
	00	0	0
Затраты на специальное оборудование для научных (экспериментальных) работ	6079,2	6763,4	6579,5
Затраты по основной заработной плате исполнителей темы	109414	114533	113139
Затраты по дополнительной заработной плате исполнителей темы	16865,05	17179,95	16970,8
Отчисления во внебюджетные фонды	37883,715	41904,39	39032,94
Накладные расходы	34022,7	39513,885	35187,5
Бюджет затрат НИИ	255994,8	271815,52	269122,474

Из таблицы видно, что общие бюджеты второго и третьего вариантов значительно больше чем первый вариант исполнения.

#### 4.4 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Определение эффективности происходит на основе расчёта интегрального показателя эффективности научного исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Интегральный показатель финансовой эффективности научного исследования получают в ходе оценки бюджета затрат трех (или более) вариантов исполнения научного исследования. Для этого наибольший интегральный показатель реализации технической задачи принимается за базу расчета (как знаменатель), с которым соотносятся финансовые значения по всем вариантам исполнения.

$$I_{фин}^{исп1} = 255994,8/271815,52 = 0,94;$$

$$I_{фин}^{исп2} = 271815,52/271815,52 = 1;$$

$$I_{фин}^{исп3} = 264122,474/271815,52 = 0,99;$$

Расчет интегрального показателя ресурсоэффективности приведён далее, в таблице 2.

Таблица 2 – Сравнительная оценка характеристик вариантов исполнения проекта

Критерии / Объект исследования	Весовой коэффициент параметра	Исп. 1	Исп. 2	Исп. 3
Надежность	0,2	4	4	4
Потребность в ресурсах памяти	0,15	4	4	4
Функциональная мощность (предоставляемые возможности)	0,2	4	3	4
Удобство	0,2	4	4	5
Скорость обработки данных	0,15	3	5	5
Безопасность	0,1	3	4	4
ИТОГО	1			

$$I_{рес}^{исп1} = \sum_{i=1}^6 \text{коэффициент}_i^{исп1} * \text{балл}_i^{исп1} = 3,75$$

$$I_{рес}^{исп2} = \sum_{i=1}^6 \text{коэффициент}_i^{исп2} * \text{балл}_i^{исп2} = 3,95$$

$$I_{рес}^{исп3} = \sum_{i=1}^6 \text{коэффициент}_i^{исп3} * \text{балл}_i^{исп3} = 4,35$$

Интегральный показатель эффективности вариантов исполнения разработки определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя.

$$I_{эфф}^{исп1} = \frac{I_{рес}^{исп1}}{I_{фин}^{исп1}} = 3,989$$

$$I_{\text{эфф}}^{\text{исп2}} = \frac{I_{\text{рес}}^{\text{исп2}}}{I_{\text{фин}}^{\text{исп2}}} = 3,95$$

$$I_{\text{эфф}}^{\text{исп3}} = \frac{I_{\text{рес}}^{\text{исп3}}}{I_{\text{фин}}^{\text{исп3}}} = 4,39$$

Далее все показатели сведены в одну таблицу **Error! Reference source not found.** Для сравнительной эффективности вариантов исполнения строка разделена на две строки, чтобы сравнение было с двумя оставшимися вариантами исполнения.

Таблица 19 – Сравнительная эффективность разработки

№ п/п	Показатели	Исп.1	Исп.2	Исп.3
1	Интегральный финансовый показатель разработки	0,94	1	0,99
2	Интегральный показатель ресурсоэффективности разработки	3,75	3,95	4,35
3	Интегральный показатель эффективности	3,989	3,95	4,39
4	Сравнительная эффективность вариантов исполнения	1,01 (с исп.2)	0,99 (с исп.1)	1,12(с исп.1)
		0,91 (с исп.3)	0,9 (с исп.3)	1,11 (с исп.2)

Таким образом, значений интегральных показателей эффективности вариантов исполнения 3 больше чем значений исполнения 1 и больше чем значений исполнения 2, поэтому выбрать эффективный вариант научного руководителя исполнения 3 для решения поставленной в проекте технической задачи.

## **5 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ**

### **Введение**

Обеспечение производственной и экологической безопасности является необходимым условием реализации любых проектов, в том числе конструкторских и исследовательских. Обеспечение безопасности, в общем, предполагает создание безопасных и благоприятных рабочих условий для всех лиц, задействованных в работах, предусмотренных проектом, а также условий, обеспечивающих экологическую безопасность окружающей среды.

Первичным этапом в задаче обеспечения безопасности труда является выявление возможных причин потенциальных несчастных случаев, производственных травм, профессиональных заболеваний, аварий и пожаров. Дальнейшими этапами являются разработка мероприятий по устранению выявленных причин и их реализация. Потенциальные причины и риски, а также конкретный набор мероприятий по их устранению, определяются спецификой выполняемых работ и априорными условиями труда (в частности, видом и состоянием рабочих мест исполнителей).

Данная работа является исследованием способа сокрытия информации в изображениях формата BMP. Основным исполнителем работы является программист. Объектом для исследования работы является изображение BMP. Работа содержит 7 различных алгоритмов.

На рабочем месте возможно возникновение вредных факторов, таких как: недостаточная освещенность рабочего места, повышение уровня шума, повышенная либо пониженная температура воздуха.

Также на данном рабочем месте могут иметь место проявления опасных факторов среды, например, поражение электрическим током. Возможной чрезвычайной ситуацией на рабочем месте является лишь возникновение пожара.

## 5.1 Производственная безопасность

Производственные условия на рабочем месте характеризуются наличием различных опасных и вредных производственных факторов, оказывающих негативное влияние на работников. Под вредными факторами, понимают такие факторы трудового процесса и рабочей среды, которые характеризуются потенциальной опасностью для здоровья, в частности способствуют развитию каких-либо заболеваний, приводят к повышенной утомляемости и снижению работоспособности. При этом, вредные факторы проявляются при определенных условиях таких как интенсивность и длительность воздействия. Опасные производственные факторы способны моментально оказать влияние на здоровье работника: привести к травмам, ожогам или к резкому ухудшению здоровья работников в результате отравления или облучения.

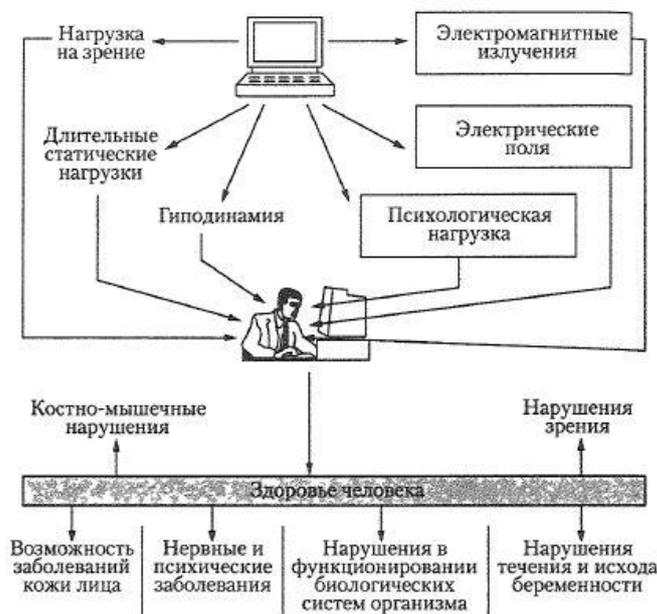


Рис.26. Факторы отрицательного воздействия компьютера на здоровье человека

Сегодня в стране более 50% лиц, работающих на ПЭВМ (персональной электронной вычислительной машине) или совмещающих эту работу с другой, имеют функциональные отклонения в состоянии

здоровья. Защита от отрицательного воздействия ПЭВМ является одной из важнейших медико-биологических и социальных задач. Факторы отрицательного воздействия ПЭВМ на человека показаны на рисунке 26.

### 5.1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения

#### 5.1.1.1 Отклонение показателей микроклимата в помещении

Одним из необходимых благоприятных условий труда является обеспечение в помещениях нормальных метеорологических условий, оказывающих существенное влияние на тепловое самочувствие человека. Метеорологические условия в производственных помещениях (микроклимат), зависят от ряда особенностей технологического процесса, а также внешних условий (климата, сезона, условий вентиляции и отопления).

Для безопасной работы необходимо соблюдать показатели микроклимата, в этом случае приведены оптимальные величины показателей микроклимата на рабочих местах производственных помещений по СанПиН 2.2.4.548-96 (табл. 20). Работа программиста относится к категории 1а, потому что уровень энергозатрат до 139 Вт.

Таблица 20 - Допустимые параметры микроклимата на рабочем месте

Период года	Категория работы	Температура воздуха, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	1а	20 - 25	15 - 75	0,1
Теплый	1а	21 - 28	15 - 75	0,1

Оптимальные значения перечисленных параметров для работ с ПК, установленные санитарными нормами, приведены в таблице 21

Таблица 21 - Оптимальные значения показателей микроклимата

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха
Холодный	22-24	19-26	10-80	0,1
Теплый	23-25	20-29	10-80	0,1

### **5.1.1.2 Превышение уровня шума**

Вредным производственным фактором также является шум, что связано с его негативным воздействием на организм человека. Воздействие шум снижает концентрацию внимания, нарушает физиологические функции. Под воздействием шума появляется усталость в связи с повышенными энергетическими затратами и нервно-психическим напряжением, ухудшается речевая коммуникация. Все перечисленное является причиной снижения работоспособности человека и приводит к падению производительности.

Основными источниками шума в проектируемом рабочем помещении исполнителей являются механические шумы, связанные с работой привода жесткого диска и вентилятора охлаждения корпуса системного блока и блока питания компьютера.

Уровень шума колеблется от 35 до 40 дБА. По СанПиН 2.2.2/2.4.1340-03 при выполнении основной работы на ПЭВМ уровень звука на рабочем месте не должен превышать 50 дБА.

### **5.1.1.3 Естественное освещение помещений**

Для работы за компьютером важное значение имеет освещение кабинета. Недостаточная освещенность приводит к снижению контрастной чувствительности, понижению остроты зрения. Освещение должно включать в себя как естественное, так и искусственное. Для источников искусственного освещения применяют люминесцентные лампы типа ЛБ. Минимальный размер объекта различия входит в диапазон 0,5 до 1,0, следовательно, работа относится к разряду IV. Подразряд Г, т.к. контраст объектов различия с фоном большой, сам фон светлый. В соответствии с СП 52.13330.2011 норма освещенности в кабинете должна быть  $E_n = 200$  лк [СП 52.13330.2011]

Пульсация при работе с компьютером не должна превышать 5% [СанПиН 2.2.2/2.4.1340-03]. Увеличение коэффициента пульсации освещенности снижает зрительную работоспособность, повышает утомляемость, воздействует на нервные элементы коры головного мозга и

фоторецепторные элементы сетчатки глаз. Для снижения пульсации необходимо использовать светильники, в которых лампы работают от переменного тока частотой 400 Гц и выше.

## **5.1.2 Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения**

### **5.1.2.1 Электробезопасность**

Одним из выявленным опасных факторов является поражение электрическим током, так как напряжение считается безопасным при  $U < 42$  В, а вычислительная техника питается от сети 220 В 50 Гц. Ток является опасным, так как 20 – 100 Гц – ток наиболее опасен. Поэтому результатом воздействия на организм человека электрического тока могут быть электрические травмы, электрические удары, и даже смерть [ГОСТ Р 12.1.009-2009].

Виды электротравм: местные электротравмы, к ним относятся: электрический ожог, электрические знаки, металлизация кожи, механические повреждения). Особую опасность представляют электрические травмы, которые выглядят в виде ожогов. Электрический ожог возникает на том месте тела человека, в котором контакт происходит с токоведущей частью электроустановки. Электроожоги сопровождаются кровотечениями, омертвением отдельных участков тела. Лечатся они гораздо труднее и медленнее обычных термических.

В результате механического повреждения могут разорваться кровеносные сосуды, нервные ткани, а также случаются вывихи суставов и даже переломы костей. Такие повреждения могут возникнуть в результате сокращений мышц под действием тока, который проходит через тело человека.

Электрические знаки в основном безболезненны, они могут возникнуть у 20% пострадавших от тока. Иногда электрические знаки выглядят в виде царапин, ушибов, бородавок, мозолей, также они

представляют собой серые или бледно-желтые пятна круглоовальной формы с углублением в центре. Чтобы защититься от поражения током, необходимо:

- обеспечить недоступность токоведущих частей от случайных прикосновений;
- электрическое разделение цепи;
- устранять опасности поражения при проявлении напряжения на разных частях;

При работе с компьютером прикосновения к его элементам могут возникнуть токи статического электричества, которые в свою очередь имеют свойство притягивать пыль и мелкие частицы к экрану. Пыль на экране ухудшает видимость, а при подвижности воздуха может попасть на кожу лица и в легкие, что вызывает заболевание кожи и дыхательных путей.

Есть специальные шнуры питания с заземлением и экраны для снятия статического электричества, это поможет защититься от статического электричества, а также необходимо проводить регулярную влажную уборку рабочего помещения. Мониторы являются источниками интенсивных электромагнитных полей. Электромагнитные поля могут вызывать изменения в клетках. Длительное воздействие низких частот ЭВМ вызывает нарушения сердечнососудистой и центральной нервной системы, небольшие изменения в составе крови. Возможно возникновение катаракты глаз, злокачественных опухолей при интенсивном длительном воздействии.

По электробезопасности рабочее место относится к помещениям без повышенной опасности поражения людей электрическим током характеризуются отсутствием условий, создающих повышенную или особую опасность. К ним относятся жилые помещения, лаборатории, конструкторские бюро, заводоуправление, конторские помещения и другие.

Степень воздействия зависит от продолжительности работы и индивидуальных особенностей организма. Для снижения уровня воздействия, необходимо:

- экранирование экрана монитора;

- соблюдать оптимально расстояние от экрана;
- рационально размещать оборудование (если имеется несколько компьютеров, то расстояние между боковыми и задними стенками компьютеров должно быть 1,22 м);
- организовывать перерывы 10-15 минут через каждые 45-60 минут работы [СанПиН 2.2.2/2.4.1340-03].

#### **5.1.2.2 Пожаровзрывобезопасность**

Одними из наиболее вероятных и разрушительных видов ЧС являются пожар или взрыв на рабочем месте. Пожарная безопасность представляет собой единый комплекс организационных, технических, режимных и эксплуатационных мероприятий по предупреждению пожаров и взрывов.

Причинами возгораний в рабочей зоне являются:

- резкие перепады напряжения;
- короткое замыкание в проводке, когда рубильник не отключен;
- пожар в соседней аудитории;
- короткое замыкание в розетке;

#### **5.2 Экологическая безопасность**

Непосредственно с выполнением данной работы, могут быть связаны негативно влияющие на экологию факторы, сопутствующие эксплуатации ПК. В частности, аспектами негативного влияния являются, отходы и выбросы, имеющие место на этапе производства ПК, а также отходы, связанные с неполной их утилизацией. Кроме того, компьютерная техника является набором приборов, потребляющих электроэнергию, в связи с чем, нерациональное их использование может быть также расценено, как необоснованная нагрузка на окружающую среду.

Эксплуатация компьютерной техники может сопровождаться следующими негативными факторами влияния на окружающую среду:

- локальное повышение электромагнитного и радиоактивного фона;

- повышение интенсивности звукового фона (слышимый шум, инфра- и ультразвуки);
- образование твердых отходов (компьютерный лом, бумага и т.п.) и жидких отходов (сточные воды);
- неоправданное потребление электроэнергии (связано с использованием ПК не на полную мощность в течение всего его рабочего времени) и прочее.

Также в СанПиН 2.2.2/2.4.1340-03, даются следующие общие рекомендации по снижению опасности для окружающей среды, исходящей от компьютерной техники:

- применять оборудование, соответствующее санитарным нормам и стандартам экологической безопасности;
- применять расходные материалы с высоким коэффициентом использования и возможностью их полной или частичной регенерации;
- отходы в виде компьютерного лома утилизировать;
- использовать экономные режимы работы оборудования.

На основе выполненного выше анализа, стоит отметить, что современные ПК, практически не оказывают негативного влияния на окружающую среду, посредством электромагнитных (в разных диапазонах частот спектра) излучениях. Кроме того, для современных ПК характерен низкий уровень производимых шумов. Таким образом, при дальнейшем рассмотрении проблемы целесообразно остановиться на последних двух факторах влияния. При использовании данных рекомендаций, возможно, существенно сократить наносимый экологии вред и снизить действующие значения вредных факторов до приемлемого минимума.

### **5.3 Безопасность в чрезвычайных ситуациях**

Наиболее типичная чрезвычайная ситуация - это пожар.

#### **5.3.1 Инструкция по помещению безопасности**

В помещениях запрещается:

- Применять нестандартные электроприборы некалиброванные плавкие вставки, электроприборы, имеющие неисправности, которые в соответствии с инструкцией по эксплуатации могут привести к пожару.
- Пользоваться электрокипятильниками, электронагревательными приборами для обогрева помещений.
- Пользоваться электроплитками, электрическими чайниками и кофеварками, не имеющими устройства тепловой защиты, без подставок из негорючих теплоизоляционных материалов.
- Устанавливать электроприборы на подоконниках, на другие электроприборы, на пол, на неустойчивое основание.
- Применять переходники, временную электропроводку для подключения холодильников, электронагревательных приборов.
- Подключать два и более потребителя электроэнергии к одному источнику электропитания.
- Самовольно проводить электромонтажные работы.
- Хранить пожароопасные вещества и материалы.
- Курить.
- Пользоваться открытым огнем.

### **5.3.2 Обязанности проживающих в помещениях**

- Знать внутреннюю планировку здания помещения, расположение лестничных клеток, основных и запасных эвакуационных выходов, средств пожаротушения (огнетушителей, внутренних пожарных кранов).
- Уметь пользоваться средствами пожаротушения, знать их тактико-технические данные.
- Неукоснительно подчиняться сигналам оповещения о пожаре, срочно покидать помещения.

### **5.3.3 Необходимые действия при возникновении пожара в помещении**

- Немедленно сообщить в службу спасения по телефону 01.

- Приступить к тушению пожара имеющимися в помещении средствами пожаротушения.

- Если ликвидировать очаг пожара своими силами не представляется возможным, выйти из помещения и закрыть дверь, не запирая ее на замок.

#### **5.4. Правовые и организационные вопросы обеспечения безопасности**

##### **5.4.1 Правовые нормы трудового законодательства**

Продолжительность рабочего дня не должна превышать 40 часов в неделю. Возможно сокращение рабочего времени. Для работников, возраст которых меньше 16 лет – не более 24 часа в неделю, от 16 до 18 лет – не более 35 часов, как и для инвалидов I и II группы. Также рабочее время зависит от условий труда: для работников, работающих на рабочих местах с вредными условиями для жизни - не больше 36 часов в неделю.

Светопроемы должны быть ориентированы преимущественно на север и северо-восток. В рабочем кабинете имеется три окна, два ориентированные на север и одно на северо-запад. Площадь на одно рабочее место с ПЭВМ для взрослых пользователей должна составлять не менее 6,0м<sup>2</sup>. Требования к организации и оборудованию рабочих мест с ПЭВМ по СанПиН 2.2.2/2.4.1340-03

Таблица 22

Требование	Требуемые значения параметров	Значения параметров в комнате общежития
Высота рабочей поверхности стола	680 – 800 мм	740 мм
Расположение монитора от глаз пользователя	600 – 700 мм	640 мм
Расположение клавиатуры на поверхности стола от края	100 – 300 мм	190 мм
Высота стула над полом (для роста 161-170 см)	420 мм	420 мм
Угол наклона монитора	0 – 30 градусов	5 градусов

Рабочее место также необходимо оборудовать подставкой для ног, которая отсутствует на рабочем месте. Рабочий стул должен быть подъемно – поворотным, для регулировки высоты и угла наклона.

#### 5.4.2 Организационные мероприятия обеспечения безопасности

Рабочее место должно обеспечивать возможность удобного выполнения работ, учитывать размеры рабочей зоны, а также необходимость передвижения в ней работающего. Невыполнение требований к расположению и компоновке рабочего места может привести к получению работником производственной травмы или развития у него профессионального заболевания. Рабочее место должно соответствовать требованиям ГОСТ 12.2.032-78 «ССБТ. Общие эргономические требования».

Конструкция оборудования и рабочего места при выполнении работ в положении сидя должна обеспечивать оптимальное положение работающего, которое достигается регулированием высоты рабочей поверхности, высоты сидения, оборудованием пространства для размещения ног и высотой подставки для ног. Оптимальные параметры рабочего места при работе с ЭВМ представлены в таблице 23:

Таблица 23

Параметры	Значение параметра	Реальные значения
Высота рабочей поверхности стола	От 600 до 800 мм	700
Высота от стола до клавиатуры	Около 20 мм	20
Высота клавиатуры	600-700, мм	600
Удаленность клавиатуры от края стола	Не менее 80 мм	100
Удаленность экрана монитора от глаз	500-700, мм	500
Высота сидения	400-500, мм	500
Угол наклона монитора	0-30, град.	20
Наклон подставки ног	0-20, град.	0

Конструкция рабочего кресла поддерживает рациональную рабочую позу, а также позволяет изменить позу, чтобы снизить статическое напряжение мышц шейно-плечевой области и спины, это позволяет предотвратить утомление. Не рекомендуется располагать компьютеры рядом друг с другом в целях уменьшения действия переменного электрического поля.

## ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы были исследованы алгоритмы для сокрытия информации в изображениях формата BMP. После исследования, мы получили знания о структуре изображения BMP, знания о преобразовании битов в изображении BMP и сокрытии информации в изображении. Были получены знания о цифровых водяных знаках, об алгоритмах шифрования, в частности, MD5, защите цифровых изображений и оптимизации качества изображения после сокрытия информации.

Формат изображения BMP не очень сложный, но очень хороший для сокрытия информации. Самый сложный алгоритм – это алгоритм для задачи «цифровые водяные знаки», потому что необходимо разделить изображения и потом найти все точки, которые лежат в промежутке [0;255].

Алгоритмы были реализованы в виде приложений «СРТЕ» и «WatermakingPremium» на языке C#. Разработанные программы позволяют защитить информацию в изображениях формата BMP: мы можем скрыть информацию и потом восстановить данные. В дальнейшем планируется продолжить работу над улучшением качества изображений.

## СПИСОК ПУБЛИКАЦИЙ

1. 1. Чан Т. -. Проверка подлинности электронных подписей в PDF файлах // Технологии Microsoft в теории и практике программирования: сборник трудов XII Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых, Томск, 25-26 Марта 2015. - Томск: ТПУ, 2015 - С. 245-246
2. 2. Чан Т. -. Анализ и реализация алгоритмов структуры данных в изображениях формата BMP // Технологии Microsoft в теории и практике программирования: сборник трудов XIII Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых, Томск, 25-26 Марта 2016. - Томск: ТПУ, 2015

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. BMP[Электронный ресурс] URL: <https://ru.wikipedia.org/wiki/BMP>
2. Thuật toán CPT[Электронный ресурс] URL: <http://diendan.congdongcviet.com/threads/t84247::giau-tin-trong-anh-den-trang-bang-thuat-toan-cpt-nhu-the-nao.cpp>
3. Прячем текст в Bitmap[Электронный ресурс] URL: <https://habrahabr.ru/post/115673/>
4. Форматы графических файлов [Электронный ресурс] URL: <http://kunegin.narod.ru/ref1/code/6.htm>
5. Kỹ thuật giấu tin trong ảnh Steganography[Электронный ресурс] URL: <http://www.van-binh.com/2015/11/ky-thuat-giau-tin-trong-anh.html>
6. Формат BMP-файла [Электронный ресурс] URL: <http://pascal.sources.ru/articles/099.htm>
7. MD5 [Электронный ресурс] URL: <https://ru.wikipedia.org/wiki/MD5>
8. Các thuật toán giấu tin trong ảnh – Phạm Đức Nam[Книжный ресурс]
9. Thủy văn thuật toán nghịch và ứng dụng[Книжный ресурс]
10. Đồ án tốt nghiệp[Электронный ресурс] URL: [http://lib.hpu.edu.vn/bitstream/handle/123456789/16756/21\\_NgoVanHiep\\_CT1201.pdf?sequence=1](http://lib.hpu.edu.vn/bitstream/handle/123456789/16756/21_NgoVanHiep_CT1201.pdf?sequence=1)

## ПРИЛОЖЕНИЕ А.(ОБЯЗАТЕЛЬНОЕ)

### СТРУКТУРА ИЗОБРАЖЕНИЯ BMP

```
public ImageInfo(string FileDir,string FileName,bool allInfo)
{
try
{
if (File.Exists("Image/~temp.bmp")) File.Delete("Image/~temp.bmp");
File.Copy(FileDir, "Image/~temp.bmp");
FileStream fstream = newFileStream("Image/~temp.bmp", FileMode.Open);
BinaryReader read = newBinaryReader(fstream);
    fstream.Position = 0;
    byte0 = read.ReadByte();
    byte1 = read.ReadByte();
    fstream.Position = 2;
    byte0 = read.ReadByte(); byte1 = read.ReadByte();
    byte2 = read.ReadByte(); byte3 = read.ReadByte();
    sizeofFile = byte3 * Math.Pow(256, 3) + byte2 * Math.Pow(256, 2) + byte1 *
Math.Pow(256, 1) + byte0;
    fstream.Position = 6;
    byte0 = read.ReadByte(); byte1 = read.ReadByte(); PositionByte = (int)(byte1 *
Math.Pow(256, 1) + byte0);
    fstream.Position = 10;
    byte0 = read.ReadByte(); byte1 = read.ReadByte();
    byte2 = read.ReadByte(); byte3 = read.ReadByte();
    d_PointerDataImage = byte3 * Math.Pow(256, 3) + byte2 * Math.Pow(256, 2) +
byte1 * Math.Pow(256, 1) + byte0;
    fstream.Position = 14;
    byte0 = read.ReadByte(); byte1 = read.ReadByte();
    byte2 = read.ReadByte(); byte3 = read.ReadByte();
    KichthuocInfoheader = byte3 * Math.Pow(256, 3) + byte2 * Math.Pow(256, 2) +
byte1 * Math.Pow(256, 1) + byte0;
    PointerToTableColor = KichthuocInfoheader + 14;
    byte0 = read.ReadByte(); byte1 = read.ReadByte();
    byte2 = read.ReadByte(); byte3 = read.ReadByte();
    width = byte3 * Math.Pow(256, 3) + byte2 * Math.Pow(256, 2) + byte1 *
Math.Pow(256, 1) + byte0;
```

```

byte0 = read.ReadByte(); byte1 = read.ReadByte();
byte2 = read.ReadByte(); byte3 = read.ReadByte();
height = byte3 * Math.Pow(256, 3) + byte2 * Math.Pow(256, 2) + byte1 *
Math.Pow(256, 1) + byte0;
fstream.Position = 28;
byte0 = read.ReadByte();
byte1 = read.ReadByte();
bitperpixel = byte1 * (256) + byte0;
Console.WriteLine(bitperpixel);
Colors = Math.Pow(2, bitperpixel);
FileLocation = FileDir;
Name = FileName;
Size = sizeofFile;
Width = (int)width;
Height = (int)height;
BitPerPixel = bitperpixel;
NumberColor = Colors;
if(allInfo)
{
    fstream.Position = (int)d_PointerDataImage;
    image = newstring[(int)height, (int)width];
    pixelMatrix = newint[(int)height][];

```

## ПРИЛОЖЕНИЕ Б. (ОБЯЗАТЕЛЬНОЕ)

### ЛИСТИНГ КОДА АЛГОРИТМА «ALATTAR»

```
public static bool KiemtraKhaMoAlattar(int[] pixel, int[] bit)
{
    int[] result = NhungTin_Alattar(pixel, bit);
    for (int i = 0; i < pixel.Count(); i++)
        if (result[i] < 0 || result[i] > 255) return false;
    return true;
}

public static int[] NhungTin_Alattar(int[] pixel, int[] bit)
{
    int n = pixel.Count();
    int[] tempPixel = new int[n];
    int[] resultPixel = new int[n];
    int sumU = 0, sumV = 0, temp = 0;
    for (int i = 0; i < n; i++) sumU += pixel[i];
    //-----Tính các vector v
    tempPixel[0] = (int) Math.Floor((sumU * 1.0) / n);
    for (int i = 1; i < n; i++)
    {
        tempPixel[i] = 2 * (pixel[i] - pixel[0]) + bit[i - 1];
    }
    for (int i = 1; i < n; i++) sumV += tempPixel[i];
    temp = (int) Math.Floor((sumV * 1.0) / n);
    resultPixel[0] = tempPixel[0] - temp;
    for (int i = 1; i < n; i++)
    {
        resultPixel[i] = tempPixel[i] + resultPixel[0];
    }
    return resultPixel;
}

public static int[] Khoiphuctin_Alattar(int[] pixel)
{
    int[] bit = new int[pixel.Count() - 1];
    int[] tempPixel = new int[pixel.Count()];
}
```

```

int[] resultPixel = new int[pixel.Count()];
int sumV1 = 0, sumV2 = 0;
int n = pixel.Count();

for (int i = 0; i < n; i++) sumV1 += pixel[i];
    tempPixel[0] = (int)Math.Floor((sumV1*1.0) / n);
for (int i = 1; i < n; i++)
    {
        bit[i - 1] = (pixel[i] - pixel[0]) % 2;
        tempPixel[i] = (int)Math.Floor(((pixel[i] - pixel[0])*1.0) / 2);
    }
for (int i = 1; i < n; i++) sumV2 += tempPixel[i];
    resultPixel[0] = tempPixel[0] - (int)Math.Floor((sumV2*1.0)/n);

for (int i = 1; i < n; i++) resultPixel[i] = tempPixel[i] + resultPixel[0];
return resultPixel;
    }

```

## ПРИЛОЖЕНИЕ В. (ОБЯЗАТЕЛЬНОЕ)

### ЛИСТИНГ КОДА АЛГОРИТМА «DIFFERENCE EXPANSION»

```
public static bool KiemtraKhaMo(int[] pixel, int[] bit)
{
    int[] result = NhungTinMoRongHieu(pixel, bit);
    for (int i = 0; i < pixel.Count(); i++)
        if (result[i] < 0 || result[i] > 255) return false;
    return true;
}

public static bool KiemtraKhaMo(int[] pixel)
{
    int n = pixel.Count() - 1;
    int[] bit = new int[n]; string str = "";
    for (int u = 0; u < Math.Pow(2, n); u++)
    {
        str = Functions.ConvertToBinaryBit((byte)u, n);
        for (int i = 0; i < str.Length; i++)
        {
            if (str[i] == '1') bit[i] = 1;
            else bit[i] = 0;
        }
        int[] result = NhungTinMoRongHieu(pixel, bit);
        for (int i = 0; i < pixel.Count(); i++)
            if (result[i] < 0 || result[i] > 255) return false;
    }
    return true;
}

public static int[] NhungTinMoRongHieu(int[] pixel, int[] bit)
{
    int[] tempPixel = new int[pixel.Count()];
    int[] resultPixel = new int[pixel.Count()];
    for (int i = 0; i < pixel.Count(); i++)
        tempPixel[i] = pixel[i];
    //-----
    Array.Sort(tempPixel);
    int a = tempPixel[(int)Math.Floor(tempPixel.Count() * 1.0 / 2)];
}
```

```

int k = TimChiSo_Thuan(pixel, a);
//-----
int index = 0;
for (int i = 0; i < pixel.Count(); i++)
if (i == k) resultPixel[i] = a;
else
    {
if (pixel[i] >= a) resultPixel[i] = a + (2 * Math.Abs(pixel[i] - a) + bit[index]);
else resultPixel[i] = a - (2 * Math.Abs(pixel[i] - a) + bit[index]);
        index++;
    }
return resultPixel;
    }
publicstaticint[] KhoiphucMoRongHieu(int[] pixel)
    {
int[] tempPixel = newint[pixel.Count()];
int[] resultPixel = newint[pixel.Count()];
for (int i = 0; i < pixel.Count(); i++)
        tempPixel[i] = pixel[i];
//-----
Array.Sort(tempPixel);
int j = TimChiSo_Nguoc(tempPixel);
int a;
if (j == -1) a = tempPixel[0]; else a = tempPixel[j];
int k = TimChiSo_Thuan(pixel, a);
for (int i = 0; i < pixel.Count(); i++)
if (i == k) resultPixel[i] = a;
else
if (pixel[i] >= a) resultPixel[i] = a + (int)Math.Floor((Math.Abs(pixel[i] - a)*1.0/2));
else resultPixel[i] = a - (int)Math.Floor((Math.Abs(pixel[i] - a)*1.0/ 2));
return resultPixel;
    }
publicstaticint[] TrichTinMoRongHieu(int[] pixel)
    {
int[] tempPixel = newint[pixel.Count()];
int[] resultBit = newint[pixel.Count()-1];
for (int i = 0; i < pixel.Count(); i++)
        tempPixel[i] = pixel[i];

```

```
Array.Sort(tempPixel);
int j = TimChiSo_Nguoc(tempPixel);
int a;
if (j == -1) a = tempPixel[0]; else a = tempPixel[j];
int k = TimChiSo_Thuan(pixel, a);
int index = 0;
for (int i = 0; i < pixel.Count(); i++)
if (i != k)
    {
        resultBit[index] = Math.Abs(pixel[i] - a)%2;
        index++;
    }
return resultBit;
}
```

## ПРИЛОЖЕНИЕ Г. (РЕКОМЕНДУЕМОЕ)

### ЛИСТИНГ КОДА АЛГОРИТМА ДЛЯ ЭЛЕМЕНТОВ НЕ РАШИРЯЕМ

#### БХ

```
public static int[] NhungTinChenBitThap(int[] pixel, int[] bit)
{
    int[] tempPixel = new int[pixel.Count()];
    int[] resultPixel = new int[pixel.Count()];
    for (int i = 0; i < pixel.Count(); i++) tempPixel[i] = pixel[i];
    //-----
    Array.Sort(tempPixel);
    int a = tempPixel[tempPixel.Count() / 2];
    //-----
    int index = 0;
    for (int i = 0; i < pixel.Count(); i++)
    if (Math.Abs(pixel[i] - a) <= 1) resultPixel[i] = pixel[i];
    else
    {
        if (pixel[i] >= a + 2)
        {
            resultPixel[i] = a + (2 * (Math.Abs(pixel[i] - a) / 2) + bit[index]);
        }
        else resultPixel[i] = a - (2 * (Math.Abs(pixel[i] - a) / 2) + bit[index]);
        index++;
    }
    return resultPixel;
}

public static int[] NhungTinChenBitThap(int[] pixel, int[] bit, ref int[] LSB)
{
    int[] tempPixel = new int[pixel.Count()];
    int[] resultPixel = new int[pixel.Count()];
    for (int i = 0; i < pixel.Count(); i++) tempPixel[i] = pixel[i];
    //-----
    Array.Sort(tempPixel);
    int a = tempPixel[tempPixel.Count() / 2];
    //int k = TimChiSo_Thuan(pixel, a);
    //-----
}
```

```

int index = 0;
for (int i = 0; i < pixel.Count(); i++)
if (Math.Abs(pixel[i]-a)<=1) resultPixel[i] = pixel[i];
else
    {
if (pixel[i] >= a + 2)
    {
        resultPixel[i] = a + (2 * (Math.Abs(pixel[i] - a) / 2) + bit[index]);
    }
else resultPixel[i] = a - (2 * (Math.Abs(pixel[i] - a) / 2) + bit[index]);
    LSB[index] = Math.Abs(pixel[i] - a) % 2;
    index++;
    }
return resultPixel;
    }

```

## ПРИЛОЖЕНИЕ Д. (РЕКОМЕНДУЕМОЕ)

### ЛИСТИНГ КОД ФУНКЦИИ VALANDNEXT

```
class ValAndNext
{
    public int[] colorBinary = new int[256];
    public int[] colorNext = new int[256];
    int[] Red = new int[256];
    int[] Green = new int[256];
    int[] Blue = new int[256];
    int[] Root = new int[256];
    int RootIndex = -1;
    public void Display()
    {
        for (int i = 0; i < 256; i++)
        {
            strColor = string.Format("Color[{0}]:\t{1}\t{2}\t{3}\t{4}\t{5}\t{6}", i, Red[i],
Green[i], Blue[i], colorBinary[i], colorNext[i], KhoangCach(i, colorNext[i]));
            Console.WriteLine(strColor);
        }
    }
    public int Min(int d)
    {
        double min = 256 * 256 * 4;
        int ret = -1;
        for (int i = 0; i < 256; i++)
        {
            if (i != d)
            {
                if (KhoangCach(i, d) <= min)
                {
                    if (colorBinary[i] == -1) ret = i;
                    elseif (KhoangCach(i, d) == min) ret = Math.Max(ret, -1);
                    else ret = -1;
                    min = KhoangCach(i, d);
                }
            }
        }
    }
}
```

```

        }

    }
}
return ret;
}
public int Min_QuayVong(int d)
{
    double min = 256 * 256 * 4;
    int ret = -1;
    for (int i = 0; i < 256; i++)
    {
        if (i != d)
        {
            if (KhoangCach(i, d) <= min)
            {
                if (colorBinary[i] == -2) ret = i;
                elseif (KhoangCach(i, d) == min) ret = Math.Max(ret, -1);
                else ret = -1;
                min = KhoangCach(i, d);
            }
        }
    }
    return ret;
}
public int Min_NoiVaoNhanhGoc(int d)
{
    double min = 256 * 256 * 4;
    int ret = -1;
    for (int i = 0; i < 256; i++)
    {
        if (i != d)
        {
            if (colorBinary[i] != -2 && colorBinary[i] != -1)
            {
                if (KhoangCach(i, d) < min)
                {

```



```

int Color = Root[RootIndex];
while (Min(Color) != -1)
    {
        colorNext[Color] = Min(Color);
        colorBinary[Color] = -2;
        Color = Min(Color);
    };
if (Min_QuayVong(Color) != -1)
    {
for (int i = 0; i < 256; i++)
    {
if (colorNext[i] == Color) { colorNext[Color] = i; colorBinary[Color] = -2; break; }
    }
        Color = Root[RootIndex];
Console.WriteLine("Goc la:" + Color);
int dem = 0;
while (colorBinary[Color] == -2)
    {
        colorBinary[Color] = dem % 2;
        Color = colorNext[Color];
        dem++;
    }
elseif (Min_NoiVaoNhanhGoc(Color) != -1)
    {
        colorNext[Color] = Min_NoiVaoNhanhGoc(Color);
        colorBinary[Color] = -2;
        Color = Root[RootIndex];
int dem = 0;
while (colorBinary[Color] == -2)
    {
        colorBinary[Color] = dem % 2;

        Color = colorNext[Color];
Console.WriteLine("Color khi này là:" + Color);
        dem++;
    };
if ((dem - colorBinary[Color]) % 2 == 1)

```

```

        {
int ColorEnd = Color;
        Color = Root[RootIndex];
        dem = 1;
while (Color != ColorEnd)
        {
            colorBinary[Color] = dem % 2;
            Color = colorNext[Color];
            dem++;
        };
    }
}

publicdouble KhoangCach(int i, int j)
{
return ((Red[i] - Red[j]) * (Red[i] - Red[j]) + (Green[i] - Green[j]) * (Green[i] - Green[j]) +
(Blue[i] - Blue[j]) * (Blue[i] - Blue[j]));
}

publicvoid Input(string DirFile)
{
for (int i = 0; i < 256; i++)
{
    Root[i] = -1;
    colorBinary[i] = -1;
    colorNext[i] = -1;

};
ReadImage(DirFile);
}

```