Учреждение образования

«Белорусский государственный технологический университет»

**Кафедра информационных систем и технологий**

**«Отчёт по лабораторной работе №9»**

“Исследование криптографических хэш-функций”

**Выполнил:** студент 4 курса

1 группы специальности ИСИт

Палазник Арсений Викторович

**Проверил:** преподаватель

Сазонова Дарья Владимировна

Минск 2024

**Разработать оконное приложение, реализующее один из алгоритмов хеширования из указанного преподавателем семейства (MD или SHA; или иного). При этом можно воспользоваться доступными готовыми библиотеками. Язык программирования – на свой выбор. Приложение должно обрабатывать входные сообщения, длина которых определяется спецификацией на реализуемый алгоритм.**

Реализация приложения:

|  |
| --- |
| // Импортируем встроенный модуль crypto  const crypto = require('crypto');  const readline = require('readline');  // Создаем интерфейс для пользовательского ввода  const rl = readline.createInterface({  input: process.stdin,  output: process.stdout  });  // Функция для хеширования строки  function hashMessage(algorithm, message) {  try {  const hash = crypto.createHash(algorithm);  hash.update(message);  return hash.digest('hex');  } catch (error) {  console.error(`Ошибка при хешировании: ${error.message}`);  return null;  }  }  // Функция для оценки быстродействия алгоритма  function measurePerformance(algorithm, message, iterations) {  const startTime = process.hrtime.bigint();  for (let i = 0; i < iterations; i++) {  hashMessage(algorithm, message);  }  const endTime = process.hrtime.bigint();  const duration = Number(endTime - startTime) / 1e6; // в миллисекундах  return duration;  }  // Основной интерфейс приложения  rl.question('Введите алгоритм хеширования (например, md5, sha256, sha512): ', (algorithm) => {  rl.question('Введите сообщение для хеширования: ', (message) => {  // Выполняем хеширование  const hash = hashMessage(algorithm, message);  if (hash) {  console.log(`Хеш (${algorithm}): ${hash}`);  // Оцениваем быстродействие  const iterations = 100000;  const duration = measurePerformance(algorithm, message, iterations);  console.log(`Время выполнения ${iterations} итераций: ${duration.toFixed(2)} мс`);  console.log(`Среднее время на одну итерацию: ${(duration / iterations).toFixed(6)} мс`);  }  rl.close();  });  }); |

Результат выполнения:

