ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
Факультет Компьютерных Наук
Образовательная программа «Программная инженерия»

СОГЛАСОВАНО

Научный руководитель, доцент
Факультета Компьютерных Наук

_____ И. Н. Лесовская

«___» _____ 2026 г.

УТВЕРЖДЕНО

Академический руководитель
образовательной программы
«Программная инженерия», старший
преподаватель департамента
программной инженерии

_____ Н. А. Павлочев

«___» _____ 2026 г.

Подп. и дата	
Инв.№ дубл.	
Взам. инв.№	
Подп. и дата	
Инв.№ подл	

ШИФРОВАЛЬНАЯ МАШИНА «ЭНИГМА»: РЕАЛИЗАЦИЯ АЛГОРИТМОВ ШИФРОВАНИЯ.

Техническое задание

ЛИСТ УТВЕРЖДЕНИЯ

Исполнители:
Студентка группы БПИ-245
_____ / М. В. Горбачева /
«___» _____ 2026 г.

2026

УТВЕРЖДЕН

ШИФРОВАЛЬНАЯ МАШИНА «ЭНИГМА»: РЕАЛИЗАЦИЯ АЛГОРИТМОВ ШИФРОВАНИЯ.

Техническое задание

Листов 25

2026

Инв.№ подл	Подп. и дата	Взам. инв.№	Инв.№ дубл.	Подп. и дата

АННОТАЦИЯ

Техническое задание – это основной документ, оговаривающий набор требований и порядок создания программного продукта, в соответствии с которым производится разработка программы ее тестирование и приемка.

Настоящее Техническое задание на разработку «шифровальной машины «Энигма»: реализации алгоритмов шифрования» содержит следующие разделы: «Введение», «Основания для разработки», «Назначение разработки», «Требования к программе», «Требования к программной документации», «Технико-экономические показатели», «Стадии и этапы разработки», «Порядок контроля и приемки», приложения [7].

В разделе «Введение» указано наименование и краткая характеристика области применения программы.

В разделе «Основания для разработки» указан документ, на основании которого ведется разработка, и наименование темы разработки.

В разделе «Назначение разработки» указано функциональное и эксплуатационное назначение создаваемого программного продукта.

Раздел «Требования к программе» содержит указание на основные требования к функциональным характеристикам программы, к её надежности и к условиям эксплуатации, к составу и параметрам технических средств, к информационной и программной совместимости, к маркировке и упаковке, к транспортировке и хранению, а также специальные требования.

Раздел «Требования к программным документам» содержит указание на предварительный состав программной документации и специальные требования к ней.

Раздел «Технико-экономические показатели» содержит информацию об ориентировочной экономической эффективности разработки, экономические преимущества разработки программы.

Раздел «Стадии и этапы разработки» содержит информацию о стадиях разработки, этапах и содержании работ.

В разделе «Порядок контроля и приемки» указаны общие требования к приемке работы.

Настоящий документ разработан в соответствии с требованиями:

1. ГОСТ 19.101-77 [1]: Виды программ и программных документов.
2. ГОСТ 19.102-77 [2]: Стадии разработки.
3. ГОСТ 19.103-77 [3]: Обозначения программ и программных документов.
4. ГОСТ 19.104-78 [4]: Основные надписи.

5. ГОСТ 19.105-78 [5]: Общие требования к программным документам.
6. ГОСТ 19.106-78 [6]: Требования к программным документам, выполненным печатным способом.
7. ГОСТ 19.201-78 [7]: Техническое задание. Требования к содержанию и оформлению.

Изменения к данному Техническому заданию оформляются согласно ГОСТ 19.603-78 [12], ГОСТ 19.604-78 [13].

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ

1.1. Наименование программы

Наименование программы – «Шифровальная машина “Энигма”: реализация алгоритмов шифрования».

Наименование программы на английском языке – «Enigma cipher machine: implementation of encryption algorithms».

1.2. Краткая характеристика области применения программы

Программный эмулятор шифровальной машины «Энигма» — это образовательно-программное приложение, предназначенное для изучения принципов криптографии, исторического контекста Второй мировой войны и основ криптоанализа.

В отличие от простых эмуляторов, разрабатываемый продукт включает не только точную реализацию алгоритма шифрования 3-роторной «Энигмы» Wehrmacht, но и интерактивный симулятор с визуализацией процесса шифрования, а также демонстрацию методов взлома, использовавшихся в Блетчли-Парке.

Основные проблемы, решаемые программой:

1. Отсутствие наглядных инструментов для изучения работы «Энигмы» в образовательном процессе;
2. Сложность понимания принципов криптоанализа без практической демонстрации;
3. Необходимость в инструменте для экспериментов с различными конфигурациями машины.

Проект нацелен на создание комплексного решения, которое может использоваться:

- В учебном процессе при подготовке специалистов по информационной безопасности;
- В исследовательских целях для изучения исторических аспектов криптографии;
- Как демонстрационный инструмент на лекциях и семинарах по различным IT-дисциплинам.

Существующие аналоги представляют собой либо простые эмуляторы шифрования без элементов криптоанализа, либо сложные исторические реконструкции без образовательной составляющей, без ясного изложения самой сути шифрования и дешифровки сообщений.

2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

2.1. Документ(ы), на основании которого(ых) ведется разработка

Разработка ведётся на основании учебного плана подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденной академическим руководителем программы темы курсового проекта.

2.2. Наименование темы разработки Наименование программы – «Шифровальная машина “Энигма”: реализация алгоритмов шифрования».

Наименование программы на английском языке – «Enigma cipher machine: implementation of encryption algorithms».

3. НАЗНАЧЕНИЕ РАЗРАБОТКИ

3.1. Функциональное назначение

Разрабатываемая программа предназначена для эмуляции работы 3-роторной шифровальной машины «Энигма» с последующим анализом её криптографических слабостей. Продукт должен предоставлять следующие функции:

Основной модуль эмуляции:

- Точная эмуляция 3-роторной «Энигмы» (роторы I-V, рефлекторы B/C, коммутационная панель);
- Настройка всех параметров машины: выбор роторов, их порядок, начальные позиции, кольцевые настройки (Ringstellung), соединения на коммутационной панели;
- Шифрование и дешифрование текстовых сообщений;
- Визуализация текущего состояния машины.

Интерактивный симулятор:

- Графическое представление машины, визуализация шифрования сообщения;
- История операций шифрования/дешифрования

Модуль криптоанализа:

- Демонстрация метода «груды» (crib attack) на примере известных исторических сообщений;
- Визуализация поиска «петель» (loops) для определения настроек роторов(под вопросом);
- Имитация атаки по известному открытому тексту;
- Статистический анализ зашифрованного текста.

Образовательный модуль:

- Историческая справка об «Энигме» и её роли во Второй мировой войне;
- Пошаговые инструкции по использованию машины;
- Примеры криптоанализа с объяснением методов.

3.2. Эксплуатационное назначение

Основными потребителями разрабатываемого приложения являются:

1. Студенты, обучающиеся на математических и IT-направлениях, преподаватели криптографии и смежных дисциплин, информационной безопасности;
2. Исследователи в области истории криптографии;
3. Школьники, которым может быть интересно узнать что-то новое или открыть для себя .

4. ТРЕБОВАНИЯ К ПРОГРАММЕ

4.1. Требования к функциональным характеристикам

4.1.1. Требования к составу выполняемых функций

Программа должна реализовывать следующие функции:

1. **Эмуляция шифровальной машины**

1.1. Поддержка конфигурации 3-роторной «Энигмы» Wehrmacht/Luftwaffe;

1.2. Реализация 5 стандартных роторов (I, II, III, IV, V) с исторически точными проводками;

1.3. Поддержка рефлекторов B и C;

1.4. Реализация коммутационной панели (Steckerbrett);

1.5. Шифрование и дешифрование текста на латинице (26 букв).

2. **Управление конфигурацией**

2.1. Выбор и расстановка роторов в нужном порядке;

2.2. Установка начальных позиций роторов (Grundstellung);

2.3. Настройка кольцевых установок (Ringstellung);

2.4. Настройка соединений на коммутационной панели;

2.5. Сохранение и загрузка конфигураций.

3. **Интерактивная визуализация**

3.1. Графическое отображение состояния всех компонентов машины;

3.2. отображение вводимого незакодированного сообщения с последующим выводом закодированного;

3.3. Отображение истории операций.

4. **Модуль криптоанализа**

4.1. **Хранилище данных для криптоанализа:**

4.1.1. **База известных текстов (crib database):** локальное хранилище в формате JSON, содержащее:

- Исторически известные «груды» (cribs) на немецком и английском языках;
- Шаблоны сообщений (message patterns) с указанием позиций вероятных cribs;
- Частотные таблицы букв для немецкого языка 1940-х годов;
- Словарь общих слов и фраз немецких военных сообщений.

4.2. **Методы криптоанализа, которые должны быть реализованы:**

4.2.1. **Метод «Груды» (Crib Attack):**

Алгоритм:

- Поиск петель (loops) между шифртекстом и предполагаемым открытым текстом;

- Построение и решение системы уравнений для нахождения возможных настроек роторов;

- Исключение невозможных конфигураций на основе свойств «Энигмы».

4.2.2. **Атака по известному открытому тексту (Known Plaintext Attack):**

4.2.2.1. **Вход:** полная пара (открытый текст ↔ шифртекст) длиной не менее 50 символов;

4.2.2.2.**Алгоритм:**

- Перебор настроек роторов (положений и колец) с использованием свойства «x →⁄ x»;

- Применение метода исключения для коммутационной панели;

4.2.2.3. **Выход:** вероятные настройки машины с оценкой достоверности.

4.2.3. **Статистический анализ:**

- **Анализ частотности**;

- **Проверка свойства «буква не шифруется сама в себя»**;

- **Диаграмма частотности букв:** сравнение частот зашифрованного текста с эталонными частотами немецкого языка.

5. **Образовательные функции**

5.1. Историческая справка об «Энигме»;

5.2. Пошаговые руководства по шифрованию и взлому;

5.3. Библиотека примеров исторических сообщений;

5.4. Режим «Сценарии» с историческими задачами.

6. **Экспорт и отчетность**

6.1. Экспорт результатов в текстовый формат;

6.2. Генерация отчетов о проведенных операциях;

Изм.	Лист	№ докум.	Подп.	Дата
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

4.1.2. Требования к организации входных данных

Входные данные для программы представляют собой:

1. **Конфигурационные данные:**
    1.1. Выбор роторов (I-V) и их порядок;
    1.2. Начальные позиции роторов (3 буквы от A до Z);
    1.3. Кольцевые настройки (3 цифры от 1 до 26);
    1.4. Пары соединений на коммутационной панели (до 10 пар букв);
    1.5. Выбор рефлектора (B или C).
2. **Текстовые данные:**
    2.1. Открытый текст для шифрования (латинские буквы, пробелы игнорируются);
    2.2. Шифртекст для дешифрования;
    2.3. Известный фрагмент текста (crib) для криптоанализа.
3. **Данные для криптоанализа:**
    3.1. Пара «открытый текст - шифртекст» для атаки;
    3.2. Предполагаемый crib и его позиция в шифртексте.
4. **Пользовательский ввод:**
    4.1. Нажатия кнопок в графическом интерфейсе;
    4.2. Ввод текста в текстовые поля;

4.1.3. Требования к организации выходных данных

Выходные данные программы:

1. **Результаты шифрования/дешифрования:**
    1.1. Преобразованный текст (шифртекст или открытый текст);
    1.3. Текущее состояние роторов после операции.
2. **Визуальные выходные данные:**
    2.1. Графическое представление состояния «Энигмы»;
    2.2. Вывод дешифрованного сообщения.
3. **Результаты криптоанализа:**
    3.1. Найденные возможные настройки роторов;
    3.2. Статистические отчеты о шифровании, дешифровке и взломе;

4.1.4. Требования к временным характеристикам

Требования к временным характеристикам:

1. Шифрование/дешифрование сообщения длиной 1000 символов должно выполняться не более чем за 1 секунду;
2. Отклик графического интерфейса на действия пользователя должен быть не более 100 мс;
3. Запуск программы (холодный старт) должен занимать не более 5 секунд;
4. Поиск возможных настроек методом crib attack для сообщения длиной 500 символов должен выполняться не более 30 секунд.

4.1.5. Требования к интерфейсу

Программа реализует графический интерфейс, состоящий из следующих основных компонентов:

1. **Главное окно программы:**
   - 1.1. Меню: Файл, Настройки, Помощь;
   - 1.2. Панель вкладок: «Эмулятор», «Криптоанализ», «Обучение»;
   - 1.3. Строка состояния: отображение текущего режима.
2. **Вкладка «Эмулятор»:**
   - 2.1. Панель настройки роторов:
     - 2.1.1. Выпадающие списки выбора роторов (левый, средний, правый);
     - 2.1.2. Поля для установки начальных позиций;
     - 2.1.3. Выбор рефлектора.
   - 2.2. Графическая панель коммутационной панели:
     - 2.2.1. 26 латинских букв;
     - 2.2.2. Возможность перетаскивания соединений между буквами;
     - 2.2.3. Отображение активных соединений.
   - 2.3. Панель ввода/вывода текста:
     - 2.3.1. Поле ввода открытого текста;
     - 2.3.2. Поле вывода шифртекста;
     - 2.3.3. Кнопки «Зашифровать», «Дешифровать», «Очистить».
   - 2.4. Визуализация машины:
     - 2.4.1. Схематичное изображение роторов с текущими позициями;

2.4.2. Панель истории операций.

3. **Вкладка «Криптоанализ»:**

3.1. Панель ввода данных для атаки:

3.1.1. Поле для шифртекста;

3.1.2. Поле для известного открытого текста (crib);

3.1.3. Поле для указания позиции crib.

3.2. Панель визуализации криптоанализа:

3.2.1. Графическое отображение поиска «петель»;

3.2.2. Таблица возможных настроек роторов;

3.3. Панель статистического анализа:

3.3.2. Частотный анализ текста.

4. **Вкладка «Обучение»:**

4.1. Историческая справка с изображениями;

4.2. Интерактивные уроки по работе с «Энигмой»;

4.3. Библиотека исторических примеров;

4.4. Режим «Сценарии» с задачами разной сложности.

4.2. Требования к надёжности

1. Программа не должна аварийно завершаться при любых входных данных;

2. При вводе некорректных данных (недопустимых символов, неверных настроек) программа должна выводить понятное сообщение об ошибке;

3. Возможность отмены последней операции (Undo);

4.3. Условия эксплуатации

4.3.1. Климатические условия эксплуатации

Климатические условия эксплуатации, при которых должна обеспечиваться корректная работа программы, должны соответствовать требованиям, предъявляемым к техническим средствам, реализующим данный программный продукт.

4.3.2. Требования к видам обслуживания

На персональном компьютере, где производится эксплуатация программы, необходимо обеспечить регулярные проверки оборудования и программного обеспечения на наличие сбоев и неполадок.

4.3.3. Требования к численности и квалификации персонала

Для пользования данным приложением достаточно одного человека. Пользователь должен обладать навыками работы с компьютером. Перед использованием программы он должен быть проинформирован о функционале программы.

4.4. Требования к составу и параметрам технических средств

Для надежной и бесперебойной работы программы требуется следующий состав технических средств:

1. Десктопное устройство с возможностью использования операционной системы Windows 10/11, macOS 10.15+ или Linux.

4.5. Требования к информационной и программной совместимости

4.5.1. Требования к информационным структурам и методам решения

Требования к информационным структурам и методам решения не предъявляются.

4.5.2. Требования к программным средствам, используемым программой

Для десктоп-приложения потребуется:

Компьютер с операционной системой Windows 10/11, macOS 10.15+ или Linux.

4.5.3. Требования к исходным кодам и языкам программирования

1. Ядро эмуляции должно быть реализовано на языке C++20 или выше в соответствии со стандартом ISO/IEC 14882:2020.

2. Для нативного внешнего вида и высокой производительности графический интерфейс десктоп-версии должен быть реализован с использованием различных фреймворков (предпочтителен фреймворке Qt 6.4+ и библиотеки QWidget).

3. Архитектура программы должна соответствовать следующим принципам:

3.1. Использование объектно-ориентированного программирования с четким разделением ответственности;

3.2. Применение паттернов проектирования для упрощения интерфейса эмуляции, для различных алгоритмов криптоанализа, для обновления GUI;

3.3. Разделение на модули: ядро эмуляции, модуль криптоанализа, графический интерфейс, утилиты.

4. Требования к структуре исходного кода:

4.1. Заголовочные файлы (.h, .hpp) должны содержать объявления классов и функций;

4.2. Файлы реализации (.cpp) должны содержать определения;

4.3. Использование пространств имен (namespaces) для логической группировки: `Enigma::Core`, `Enigma::GUI`, `Enigma::CryptoAnalysis`;

5. Требования к сборке и зависимостям:

8.1. Использование системы сборки CMake 3.16+ с поддержкой кроссплатформенности;

8.3. Поддержка сборки под Windows (MSVC), Linux (GCC/Clang), macOS (Clang).

6. Использование системы контроля версий Git.

4.5.4. Требования к защите информации и программы

1. Программа не предназначена для защиты реальных данных, только для образовательных целей;

2. Локальное хранение пользовательских данных без передачи в сеть;

4.6. Требования к маркировке и упаковке

Программа распространяется в виде:

1. Исходного кода в репозитории GitHub;
2. Исполняемого файла (.exe для Windows, .app для macOS);

4.7. Требования к транспортированию и хранению

Транспортировка осуществляется через скачивание с GitHub.

4.8. Специальные требования

Специальные требования к программе не предъявляются.

5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

5.1. Состав программной документации

1. Техническое задание (ГОСТ 19.201-78);
2. Пояснительная записка (ГОСТ 19.404-79);
3. Программа и методика испытаний (ГОСТ 19.301-79);
4. Текст программы (ГОСТ 19.401-78);
5. Руководство пользователя (ГОСТ 19.505-79);
6. Презентация проекта.

5.2. Специальные требования к программной документации

Документация должна быть выполнена в соответствии с ГОСТ 19.106-78; Пояснительная записка должна быть загружена в систему Антиплагиат через LMS «НИУ ВШЭ».

Документация и программа сдается в электронном виде в формате .pdf или .docx. в архиве формата;

За две недели до защиты комиссии все материалы курсового проекта:

- программная документация,
- программный проект,
- исполняемый файл,
- отзыв руководителя,
- отчет системы Антиплагиат

должны быть загружены одним или несколькими архивами в проект дисциплины «Курсовой проект» в личном кабинете в информационной образовательной среде SmartLMS НИУ ВШЭ.

6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

6.1. Предполагаемая потребность

Разрабатываемый эмулятор «Энигмы» с элементами криптоанализа заполняет нишу в образовательном ПО для криптографии. Существующие аналоги либо слишком просты (базовые эмуляторы шифрования), либо слишком сложны для учебных целей (профессиональные инструменты криптоанализа).

Программа будет востребована в учебных заведениях для преподавания криптографии, на исторических факультетах для изучения Второй мировой войны, в музеях криптографии или техники для интерактивных экспозиций.

6.2. Целевая аудитория

1. Студенты и преподаватели технических вузов;
2. Учителя и ученики профильных классов школ и интересующиеся;
3. Историки и реконструкторы;

6.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

Функция	Cryptool 2 эмуляция Энигмы	Python Enigma библиотека	Online Enigma веб-эмулятор	Enigma Simulator App Store	Our Project
Точная историческая эмуляция	+	+	+	+	+
Графический интерфейс	+	-	+	+	+
Визуализация работы	-	-	±	-	+
Модуль криптоанализа	+	-	-	-	+
Интерактивное обучение	-	-	-	-	+
Режим исторических сценариев	-	-	-	-	+
Итого	**3**	**1**	**2**	**2**	**6**

Таблица 1. Сравнение функциональных характеристик с аналогами

7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

7.1. Стадии разработки, этапы и содержание работ

Таблица 2 – Стадии и этапы разработки

Стадия разработки	Этап работ	Содержание работ	Сроки выполнения
Техническое задание	Обоснование необходимости разработки	Изучение исторических материалов об «Энигме»	13.11.25
		Анализ существующих аналогов	13.11.25
	Научно-исследовательский этап	Разработка математической модели «Энигмы»	15.11.25 – 03.12.25
		Изучение методов криптоанализа «Энигмы»	15.11.25 – 03.12.25
		Проектирование архитектуры программы	15.11.25 – 03.12.25
		Выбор технологий и инструментов	15.11.25 – 03.12.25
	Разработка ТЗ	Разработка технического задания	15.11.25 – 03.12.25
		Согласование ТЗ с научным руководителем	16.12.25

Продолжение таблицы 2

Стадия разработки	Этап работ	Содержание работ	Сроки выполнения
Рабочий проект	Разработка программы	Реализация ядра эмуляции (алгоритмы шифрования)	05.12.25 – 25.12.25
		Разработка графического интерфейса	25.12.25 – 15.01.26
		Реализация модуля криптоанализа	15.01.26 – 31.01.26
	Разработка программной документации	Разработка документов по ГОСТ 19	05.12.25 – 31.01.26
	Испытания программы	Разработка методики испытаний	01.02.26 – 10.02.26
		Тестирование функциональности	10.02.26 – 17.02.26
		Тестирование исторической точности	17.02.26 – 25.02.26
	Корректировка по результатам тестов	Исправление ошибок	25.02.26 – 28.02.26
	Внедрение	Подготовка к защите	Подготовка презентации и демонстрационных материалов
			01.03.26 – 15.03.26
Получение отзыва научного руководителя	16.03.26		

Продолжение таблицы 2

Изм.	Лист	№ докум.	Подп.	Дата
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Стадия разработки	Этап работ	Содержание работ	Сроки выполнения
		Загрузка в систему Антиплагиат	01.04.26
		Загрузка материалов в LMS	07.04.26
		Защита перед комиссией	14.04.26

7.2. Сроки разработки и исполнители

Разработка должна быть завершена к 14.04.2026.

Исполнитель: Горбачева Маргарита Валерьевна – студентка группы БПИ-245.

8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

Контроль и приемка осуществляются в соответствии с документом «Программа и методика испытаний».

8.1. Виды испытаний

1. Функциональное тестирование всех модулей;
2. Тестирование исторической точности эмуляции;
3. Тестирование интерфейса на удобство использования;
4. Тестирование модуля криптоанализа на исторических примерах;
5. Нагрузочное тестирование.

8.2. Общие требования к приёмке работы

Программа принимается при условии:

1. Корректной работы всех заявленных функций;
2. Соответствия историческим прототипам;
3. Наличия полного комплекта документации;
4. Успешного прохождения всех видов испытаний.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. ГОСТ 19.101-77: Виды программ и программных документов.
2. ГОСТ 19.102-77: Стадии разработки.
3. ГОСТ 19.201-78: Техническое задание. Требования к содержанию и оформлению.
4. ГОСТ 19.301-79: Программа и методика испытаний.
5. ГОСТ 19.401-78: Текст программы.
6. ГОСТ 19.404-79: Пояснительная записка.
7. ГОСТ 19.505-79: Руководство оператора.
8. Singh S. The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography. — Anchor, 2000.
9. Copeland B. J. (Ed.) The Essential Turing. — Oxford University Press, 2004.
10. Rejewski M. Mathematical Solution of the Enigma Cipher
11. The Enigma Machine: How it works and how to break it. URL: https://www.codesandciphers.org.uk/enigma/

ПРИЛОЖЕНИЕ. ССЫЛКИ НА АНАЛОГИ

Приложение	Ссылка
Cryptool 2 (с модулем Enigma)	https://www.cryptool.org/
Python Enigma Library	https://pypi.org/project/enigma/
Online Enigma Emulator	https://observablehq.com/@tmcw/enigma-machine
Enigma Simulator для iOS	https://apps.apple.com/app/enigma-simulator/
Enigma Machine Simulator (Android)	https://play.google.com/store/apps/details?id=uk.co.ordnancesurvey.enigma
Virtual Enigma Machine	http://enigma.louisedade.co.uk/
Navy M4 Enigma Simulator	https://www.lysator.liu.se/~koma/enigma/

Дата обращения: 06.12.25.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Лист регистрации изменений									
Номера листов (страниц)					Всего листов (страниц в докум.)	№ документа	Входящий № сопроводительного докум. и дата	Подп.	Дата
Изм.	Измененных	Замененных	Новых	Аннулированных					