

Промежуточный отчет по программному проекту

1. Основные планы и этапы проекта

1.1 Краткое описание проекта:

«Шифровальная машина “Энигма”: реализация алгоритмов шифрования» – это образовательно-программное приложение, эмулирующее работу настоящей немецкой шифровальной машины «Эннигма».

Название проекта:

«Шифровальная машина “Энигма”: реализация алгоритмов шифрования»

Цель проекта:

Разработать функциональное десктоп-приложение, эмулирующее работу шифровальной машины, предназначенное для изучения принципов криптографии, исторического контекста Второй мировой войны и основ криптоанализа. Данная программа будет полезна в учебном процессе при подготовке специалистов по информационной безопасности, в исследовательских целях для изучения исторических аспектов криптографии, или как демонстрационный инструмент на лекциях и семинарах по различным ИТ-дисциплинам.

Краткое описание задач:

1. Разработка технического задания, определение требований, определение архитектуры проекта.
2. Разработка десктоп-приложения на языке программирования C++.
3. Разработка программной документации: написание технического задания, пояснительной записки, программы и методики испытаний, текста программы, руководства оператора.
4. Испытания программы.
5. Защита проекта.

1.2 Планы и этапы выполнения проекта

Этап проекта	Описание работ	Ожидаемые результаты	Сроки выполнения
Обоснование необходимости разработки	Постановка задачи	Сбор исходных теоретических материалов и изучение исторических материалов об «Энигме»	13.11.25
Научно-исследовательский этап разработки	Выбор технологий и инструментов; Предварительный выбор методов решения задач; Определение требований к техническим и программным средствам; Обоснование возможности решения поставленной задачи	Определение требований для решения задачи	15.11.25 – 13.12.25

Этап проекта	Описание работ	Ожидаемые результаты	Сроки выполнения
Разработка и утверждение технического задания	Определение требований к программному продукту; Выбор языков программирования; Разработка и согласование технического задания с научным руководителем; Загрузка согласованного технического задания в SmartLMS	Готовое техническое задание для начала разработки приложения	15.11.25 – 16.12.25
Разработка программы	Предварительная разработка структуры программы; Программирование и отладка программы	Готовое приложение, реализующее все функции, определенные в техническом задании	17.12.25 – 15.04.26
Разработка программной документации	Разработка документов в соответствии с требованиями ГОСТ 19 ЕСПД (Единой системы программной документации)	Готовая программная документация	05.12.25 – 31.01.26
Испытания программы	Разработка, согласование и утверждение порядка в методики испытаний; Проведение испытаний программы в соответствии с утвержденными порядком и методикой; Корректировка программы и программной документации по результатам испытаний;	Тестирование программы и исправление недочетов	15.04.26 – 22.04.26
Подготовка и передача программы	Подготовка программы и программной документации для презентации и защиты; Представление разработанного программного продукта научному руководителю и получение отзыва; Загрузка Пояснительной записки в систему Антиплагиат через ЛМС НИУ ВШЭ; Загрузка материалов курсового проекта в ЛМС, дисциплина «Курсовой проект, 2 курс, ПИ»; Защита программного продукта комиссии	Получение готового продукта, реализующее все функции, и защита работы	22.04.26 – xx.05.26

2. Используемый технологический стек и его обоснование

2.1 Перечень используемых технологий

Технология/Инструмент	Описание	Причины выбора
C++	Язык программирования для разработки различных прикладных программ, в том числе, десктоп-приложений	Актуальный язык для создания крупных приложений в объектно-ориентированном стиле, представляя программу как совокупность взаимодействующих между собой классов и объектов
Qt 6.4+	Фреймфорк для создания интерфейса	Удобный и поддерживаемый фреймфорк для создания нативного внешнего вида и высокой производительности программы
Qwidget	Библиотека	Удобная и простая библиотека для работы с графическим интерфейсом
Visual Studio Code	Среда разработки для программирования на различных языках, в том числе, C++	Бесплатная и удобная среда разработки, поддерживающая множество плагинов и расширений
Git	Управление версиями	Удобство разработки в долгосрочной перспективе, сохранение версий разработки

2.2 Обоснование выбранного технологического стека

Использование языка программирования C++ и фреймворка Qt 6.4+ для реализации десктоп-приложения эмуляции шифровальной машины имеет следующие преимущества:

1. Простота и безопасность: C++ обеспечивает высокую читаемость и безопасность кода благодаря строгой типизации, что уменьшает количество ошибок на этапе компиляции.
2. Высокая производительность: C++ оптимизирован для работы с памятью, обеспечивая быстрые вычисления и отличное использование ресурсов устройства.
3. C++ является языком, позволяющим писать производительные программы, осуществляющие математические расчёты и реализующие вычислительные алгоритмы. Особенно хорошо пригождается в сфере криптографии и криptoанализа.
4. Документация: Обширные ресурсы и регулярные обновления делают разработку удобной и поддерживаемой.

5. Удобство реализации нативного графического интерфейса: фреймворк Qt 6.4+ позволяет создавать приятный интерфейс, при этом обеспечивая высокую производительность программы.

Visual Studio Code выбран в качестве среды разработки, потому что:

1. Предоставляет все необходимое для работы над проектом, включая редактор кода, дебаггер, симулятор устройств и инструменты для тестирования.
2. Баланс между производительностью и функциональностью: в отличие от тяжеловесных IDE, VS Code загружается за секунды и потребляет меньше ресурсов, но при этом предоставляет инструменты для профессиональной разработки.
3. Гибкость настройки под индивидуальные задачи: редактор позволяет настраивать среду разработки, используя расширения и встроенные инструменты.
4. Интегрированный терминал: возможность выполнять команды прямо внутри VS Code, что полезно при работе с инструментами CLI, запуске скриптов сборки или управления пакетами.
5. Доступен бесплатно.

Git используется, так как:

1. Контроль версий: Git позволяет отслеживать изменения в проекте, что помогает разработчикам работать с различными версиями приложения.
2. Удобство работы в долгосрочной перспективе: Git помогает выстраивать архитектуру в виде графа для удобства отслеживания версий проекта.
3. Безопасность данных: Все изменения сохраняются, что позволяет откатиться к предыдущим версиям при необходимости.

3. Критерии оценивания проекта

Критерий	Описание
Использование адаптивного дизайна	Будет использовано/Не будет использовано
Функциональность – Процент выполнения функциональных требований	Выполненные требования в процентах от общего количества
Функциональность – Количество реализованных функций	Абсолютное количество функций, которые работают правильно
Использование технологического стека – Процент использования функциональности стека (%)	Процент использования функциональности выбранного стека технологий
Документация и оформление – Полнота документации (%)	Процент от требуемого объема документации
Соблюдение сроков и плана – Процент выполнения работы в срок (%)	Процент задач, выполненных в срок
Соблюдение сроков и плана – Количество дней отклонения от плана	Общее число дней отклонения от плана