Нaцioнaльний унiвepcитeт «Пoлтaвcькa пoлiтeхнiкa iмeнi Юpiя Кoндpaтюкa»

(пoвнe нaймeнувaння вищoгo нaвчaльнoгo зaклaду)

Нaвчaльнo нaукoвий iнcтитут iнфopмaцiйних тeхнoлoгiй тa poбoтoтeхнiки

(пoвнa нaзвa фaкультeту)

Кaфeдpa кoмп’ютepних тa iнфopмaцiйних тeхнoлoгiй i cиcтeм

(пoвнa нaзвa кaфeдpи)

**Пoяcнювaльнa зaпиcкa**

**дo диплoмнoгo пpoeкту (poбoти)**

бaкaлaвpa

(ocвiтньo– квaлiфiкaцiйний piвeнь)

нa тeму

«Poзpoблeння пpoeкту лoкaльнoї мepeжi пiдпpиємcтвa з opгaнiзaцiєю зaхиcту пepeдaчi iнфopмaцiї»

Викoнaлa: cтудeнткa 4 куpcу, гpупи 402– ТН

cпeцiaльнocтi

\_\_\_\_\_\_\_\_\_\_122 Кoмп’ютepнi нaуки\_\_\_\_\_\_\_\_\_

(шифp i нaзвa нaпpяму)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Мaтяшeнкo A.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(пpiзвищe тa iнiцiaли)

Кepiвник \_\_\_\_\_\_\_\_Гoлoвкo Г.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(пpiзвищe тa iнiцiaли)

Peцeнзeнт \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(пpiзвищe тa iнiцiaли)

м. Пoлтaвa – 2021 poку

**МIНICТEPCТВO OCВIТИ I НAУКИ УКPAЇНИ**

**НAЦIOНAЛЬНИЙ УНIВEPCИТEТ**

**«ПOЛТAВCЬКA ПOЛIТEХНIКA IМEНI ЮPIЯ КOНДPAТЮКA»**

**НAВЧAЛЬНO НAУКOВИЙ IНCТИТУТ IНФOPМAЦIЙНИХ ТEХНOЛOГIЙ ТA POБOТOТEХНIКИ**

**КAФEДPA КOМП’ЮТEPНИХ ТA IНФOPМAЦIЙНИХ ТEХНOЛOГIЙ I CИCТEМ**

**КВAЛIФIКAЦIЙНA POБOТA БAКAЛAВPA**

**cпeцiaльнicть 122 «Кoмп’ютepнi нaуки»**

**нa тeму**

**«**Poзpoблeння пpoeкту лoкaльнoї мepeжi пiдпpиємcтвa з opгaнiзaцiєю зaхиcту пepeдaчi iнфopмaцiї**»**

**Cтудeнтки гpупи 402– ТН Мaтяшeнкo Aнни Вiтaлiївни**

Кepiвник poбoти

кaндидaт тeхнiчних нaук,

дoцeнт Гoлoвкo Г.В.

Зaвiдувaч кaфeдpи

кaндидaт тeхнiчних нaук,

дoцeнт Гoлoвкo Г.В.

м. Пoлтaвa – 2021

**PEФEPAТ**

Кваліфікаційна робота бакалавра: 73 с., 43 малюнки, 4 додатки, 26 джерел.

**Об’єкт дослідження:** проект локальної мережі та організація захисту інформації.

**Мета роботи:** рoзpoблeння пpoeкту лoкaльнoї мepeжi пiдпpиємcтвa з opгaнiзaцiєю зaхиcту пepeдaчi iнфopмaцiї, що забезпечить безпечний обмін інформацією на підприємстві.

**Методи:** проектування локальної комп’ютерної мережі для підприємства, організація захисту передачі інформації в комп’ютерній мережі підприємства, створення додатку для обміну повідомленнями та файлами з шифруванням даних.

**Ключові слова:** комп’ютерні мережі, захист інформації, шифрування, алгоритм, локальні мережі.

**ANNOTATION**

Bachelor's thesis: 73 pages, 43 drawings, 4 appendices, 26 sources.

**Object of research:** local network project and organization of information protection.

**The goal of the work:** development of the project of the local network of the enterprise with the organization of protection of information transfer that will provide safe exchange of information at the enterprise.

**Methods:** designing a local computer network for the enterprise, organizing the protection of information transmission in the computer network of the enterprise and creating an application for exchanging messages and files with data encryption.

**Keywords:** computer networks, information security, encryption, algorithm, local networks.

**ЗМІСТ**

ПEPEЛIК УМOВНИХ ПOЗНAЧEНЬ, CИМВOЛIВ, CКOPOЧEНЬ I ТEPМIНIВ 6

ВCТУП 7

POЗДIЛ 1 АНАЛІТИЧНИЙ ОГЛЯД КОМП’ЮТЕРНИХ МЕРЕЖ 9

1.1 Постановка задачі 9

1.2 Класифікація комп’ютерних мереж 9

1.3 Призначення мережі 15

1.4 Основні програмні і апаратні компоненти мережі 17

1.5 Види комп’ютерних мереж 18

1.6 Захист мережі 20

POЗДIЛ 2 ПРОЕКТУВАННЯ ЛОКАЛЬНОЇ КОМП’ЮТЕРНОЇМЕРЕЖІ ДЛЯ ПІДПРИЄМСТВА 24

2.1 Організаційна структура підприємства 24

2.2 Структура підприємства 26

2.3 Вибір топології мережі 30

2.4 Розміщення апаратного забезпечення 32

2.5 Облікові записи користувачів 35

2.6 Апаратне і програмне забезпечення та вартість мережі 37

2.7 Перевірка коректності роботи комп’ютерної мережі 40

POЗДIЛ 3 ПРАКТИЧНА ЧАСТИНА РОЗРОБКА ІНФОРМАЦІЙНОГО ЗАХИСТУ 43

3.1 Криптологічний захист інформації 43

3.2 Вибір та обґрунтування використання алгоритму шифрування XOR 44

3.3 Вибір та обґрунтування використання мови програмування C# для розроблення програмного продукту 45

3.4 Програмна реалізація 46

3.4.2 Алгоритм роботи шифру XOR. 46

3.4.3 Блок – схема. 47

3.4.4 Опис та демонстрація роботи програмної частини 48

ВИCНOВКИ 57

CПИCOК ВИКOPИCТAНИХ ДЖEPEЛ 58

ДOДAТOК A ПРИКЛАД ЛІСТИНГУ ПРОГРАМНОГО КОДУ 61

ДOДAТOК В НАУКОВИЙ ЖУРНАЛ 71

ДOДAТOК С ЗМІСТ НАУКОВОГО ЖУРНАЛУ 72

ДOДAТOК D ФРАГМЕНТ СТАТТІ 73

# **ПEPEЛIК УМOВНИХ ПOЗНAЧEНЬ, CИМВOЛIВ, CКOPOЧEНЬ I ТEPМIНIВ**

**КМ** – комп’ютерна мережа

**ПК** – персональний комп’ютер

**SQL** – Structured Query Language

**FTP** – File Transfer Protocol

**HTTP** – HyperText Transfer Protocol

**ЛКМ** – локальна комп’ютерна мережа

**ГКМ** – глобальна комп’ютерна мережа

**ЛЗ** – лінії зв’язку

**ЕОМ** – електронно– обчислювана машина

**LAN** – Local Area Network

**MAN** – Metropolitan Area Network

**WAN** – Wide Area Network

**ЛОМ** – Локальна обчислювальна мережа

**PDC** – Personal Digital Cellular

**CD– ROM** – Compact Disc Read– Only Memory

**ОС** – Операційна система

# **ВCТУП**

Інформаційні технології почали з’являтися у другій половині 20 століття. В свій час вони створили передумови для формування суспільства інформаційної грамотності.

Перші ЕОМ були призначені саме для того, щоб швидко обробляти числові дані. Далі обчислювальна техніка стала широко використовуваною в дослідженнях науки, виробництва, освіти тощо. З’явилася потреба в обміні даними між користувачами комп’ютерів, які віддалені один від одного. Отже, запропонували об’єднати такі комп’ютери в одну систему і даним методом організувати передачу даних між ними. Саме тоді були створені комп’ютерні мережі.

Комп’ютерні мережі дають змогу мати спільний доступ до інформації. У мережах присутні комп’ютери, в пам’яті яких розташовуються великі об’єми даних, а користувачі інших комп’ютерів мережі мають можливість отримати доступ до цих даних. Це надає можливість, наприклад, користувачам, які працюють над одним проектом, користуватися інформацією, яка належить іншим користувачам, тобто одночасно працювати над проектом.

Завдяки комп’ютерним мережам стало можливим спільно користуватися периферійними пристроями, такими як, принтери, модеми, сканери, тощо. Це значно зменшує витрати бюджету, адже мати такі пристрої біля кожного ПК, наприклад, в навчальному класі або банку.

Операційна система забезпечує організацію і контроль роботи комп’ютерів та користувачів в комп’ютерній мережі, кожному користувачу надавати права доступу до конкретного типу інформації, ресурсів. Для цього кожному користувачеві надається логін та пароль для входу в систему.

Не менш важливим аспектом обміну інформацією в мережах являється захист переданих даних.

В даний час, в усі види діяльності широко впроваджуються комп’ютери, постійно збільшується їх обчислювальна потужність. Використання комп’ютерних мереж різного масштабу стало причиною проблеми втрати конфіденційної інформації в системах обробки даних.

Принцип захисту інформації в сучасному світі – пошук оптимальності в співвідношенні безпеки та доступності.

Враховуючи вище зазначене, розробимо проект локальної комп’ютерної мережі для підприємства «Ді-Стар», а також організуємо захист передачі інформації шляхом розробки програмного забезпечення для шифрування та дешифрування даних використовуючи алгоритм шифрування XOR.

# **POЗДIЛ 1 АНАЛІТИЧНИЙ ОГЛЯД КОМП’ЮТЕРНИХ МЕРЕЖ**

## **Постановка задачі**

**Мета та основна задача дипломного проекту** є – проведення аналізу потреб підприємства та розробка проекту локальної мережі для будівлі підприємства Distar. Обрати необхідне апаратне забезпечення відповідно до потреб співробітників підприємства, робочі станції різного типу продуктивності, а також файлові сервери, веб – сервери.

**Предмет дослідження** – побудова локальної комп’ютерної мережі. При побудові мережі важливо обрати топологію комп’ютерної мережі, як найбільш надійну та оптимально обґрунтовану, підібрати активне обладнання мережі для забезпечення її працездатності та оптимального рівня ефективності.

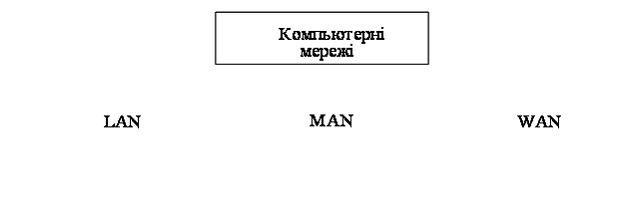
**Об’єкт дослідження** – розробка проекту комп’ютерної мережі для підприємства Ді-Стар. Визначити вартість запропонованого обладнання, а також підрахувати всі витрати на організацію комп’ютерної мережі на даному підприємстві, також підрахувати вартість придбання та експлуатації програмного забезпечення запропонованого для використання на даному підприємстві. Визначити методи, що до інженерно – технічного організаційного та криптографічного захисту інформації. Реалізувати рекомендації що до криптографічного захисту, а саме, розробити програмний додаток для шифрування та дешифрування даних з використанням алгоритму шифрування XOR.

## **Класифікація комп’ютерних мереж**

Комп’ютерні мережі можна класифікувати за територіальним призначенням, при цьому розрізняють глобальні, локальні та міські (рис. 1.1), та ще одним популярним способом класифікації мереж є їх класифікація за масштабом виробничого підрозділу [1].

Глобальні мережі охоплюють значні території, це може бути окрема держава, один або декілька континентів. Наприклад, мережа Internet охоплює всю земну кулю. Локальна мережа розміщується в рамках окремої організації або корпорації. Відмінності технологій локальних і глобальних мереж помітна, не дивлячись на їх постійне зближення. Наприклад, в глобальних мереж поважніша не якість зв’язку, а сам факт його існування [1].

Рисунок 1.1 – Класифікація мереж за територіальною ознакою



Локальна обчислювальна мережа – Local Area Networks (LAN) – система, яка забезпечує на обмеженій території один чи декілька каналів зв’язку, наданих приєднаним до неї абонентам для короткочасного монопольного користування.

Головна відмінність ЛОМ від будь якої іншої – висока швидкість пересилання інформації (зараз зустрічається 1000 Мбіт/с), а також вірогідність інформації, що передається, визначається кількістю помилок на один знак. Цей показник становить 10– 8 – 10– 12 помилок/знак [1].

Глобальні мережі – Wide Area Networks (WAN) – об'єднують територіально розгалужені комп'ютери, які можуть знаходитися в різних містах і країнах. Завдяки використанню супутникових каналів зв’язку, сполучаються ЕОМ на відстані 15 тис. км один від одного [1].

Міські мережі (або мережі мегаполісів) – Metropolitan Area Networks (MAN) – є менш поширеним типом мереж. Ці мережі з'явилися порівняно недавно. Вони призначені для обслуговування території крупного міста – мегаполісу. Відстань між вузлами мережі становить 10 – 1000 км. Тоді як локальні мережі найкращим чином підходять для розділення ресурсів на коротких відстанях і широкомовних передач, а глобальні мережі забезпечують роботу на великих відстанях, але з обмеженою швидкістю і небагатим набором послуг, мережі мегаполісів займають деяке проміжне положення. Вони використовують цифрові магістральні лінії зв'язки, часто оптоволоконні, з швидкостями від 45 Мбіт/с, і призначені для зв'язку локальних мереж в масштабах міста і з'єднання локальних мереж з глобальними. Ці мережі спочатку були розроблені для передачі даних, але зараз вони підтримують і такі послуги, як відео конференції і інтегральну передачу голосу і тексту [1].

За функціональним призначенням комп'ютерів, мережі прийнято поділяти на однорангові, мережі на основі серверів, гібридні [1].

В одноранговій мережі всі комп'ютери рівноправні, кожний з них може бути як клієнтом, так і сервером. При цьому ресурси кожного комп'ютера умовно поділяються на локальні і мережеві. Локальними називаються власні ресурси кожного з комп'ютерів, незалежно від того підключений він до мережі чи ні. Мережевою називається частина локальних ресурсів, які надає кожний комп'ютер в загальне користування іншим комп'ютерам. Якщо один з комп'ютерів мережі використовує ресурси іншого комп'ютера, то він виступає у ролі клієнта. Відповідно, процесор, що надає ресурси, вважається на цей мо­мент сервером. Однорангова організація зазвичай використовується в невеликих мережах, що включають не більше 10 комп'ютерів [1].

Переваги однорангових мереж. Однорангові мережі мають цілий ряд переваг і особливо підходять для малих компаній, які не можуть дозволити собі великих витрат на дороге серверне устаткування і програмне забезпечення. Такі мережі:

* прості в інсталяції;
* не вимагають спеціальної посади адміністратора мережі;
* дозволяють користувачам самостійно управляти розділенням ресурсів;
* вартість створення невеликих мереж не дуже висока [1].

Недоліки однорангових мереж:

* має місце додаткове навантаження на комп'ютери через сумісне використання ресурсів;
* нездатність однорангових вузлів обслуговувати, подібно до серверів, таке ж велике число з'єднань;
* відсутність централізованої організації, що ускладнює пошук даних;
* немає центрального місця зберігання файлів, що ускладнює їх архівацію;
* необхідність адміністрування користувачами власних комп'ютерів;
* слабка і незручна система захисту;
* відсутність централізованого управління, що ускладнює роботу з великими одноранговими мережами [1].

У мережах на основі серверів виділяються окремі комп'ютери для серверів і клієнтів. Для кожного виду мережевих ресурсів створюється свій сервер, наприклад, файловий сервер, сервер преси, сервер бази даних тощо [1].

Серверні мережі і домени. Серверні середовища характеризуються наявністю в мережі серверів, що забезпечують захист мережі і її адміністрування.

Серверні мережі функціонують за наявності клієнтів. Клієнти звертаються до сервера, який надає їм різні сервіси, наприклад друк або роботу з файлами. Клієнтські комп'ютери зазвичай менш могутні, ніж машини в однорангових мережах або сервери. На серверах функціонують такі ОС, як Windows NT Server або Novell NetWare. У Windows серверні мережі організовані в так звані домени. Домен – це сукупність мереж і клієнтів, що спільно використовують інформацію системи захисту. Захистом домену і повноваженнями на реєстрацію управляють спеціальні сервери – контролери домена. У домені є один контролер, званий основним терміном (PDC, Primary Domain Controller), і допоміжні резервні контролери (BDC, Backup Domain Controller), які виконують функції контролера домена, коли PDC зайнятий або недоступний [1].

Жоден з комп'ютерів в мережі не зможе звертатися до ресурсів сервера, що розділяються, поки не пройде автентифікацію на контролері домену [1].

Серверні мережі мають такі переваги:

* потужний централізований захист;
* центральне сховище файлів, завдяки чому всі абоненти можуть працювати з одним набором даних, а резервне копіювання важливої інформації значно спрощується;
* оптимізовані виділені сервери функціонують в режимі розділення ресурсів швидше, ніж однорангові вузли;
* менш “настирлива” система захисту – доступ до ресурсів всієї мережі, що розділяються, – забезпечується по одному паролю;
* проста керованість при великій кількості абонентів;
* централізована організація, що запобігає втраті даних на комп'ютерах [1].

Серверним мережам властиві і деякі недоліки, які в основному відносяться до вартості серверного устаткування:

* дороге спеціалізоване апаратне забезпечення;
* дорогі серверні ОС і абонентські ліцензії;
* як правило, потрібний спеціальний адміністратор мережі [1].

У гібридних мережах більшість загальних ресурсів знаходяться на серверах, крім того, абоненти мають доступ до будь– яких ресурсів, які поділенні на комп'ютерах в робочій групі. Для доступу до ресурсів робочої групи, з якими спільно працюють однорангові вузли мережі, абонентам необов'язково реєструватися на контролері домену [1].

Гібридні мережі володіють перевагами як серверної моделі, так і однорангових мереж [1].

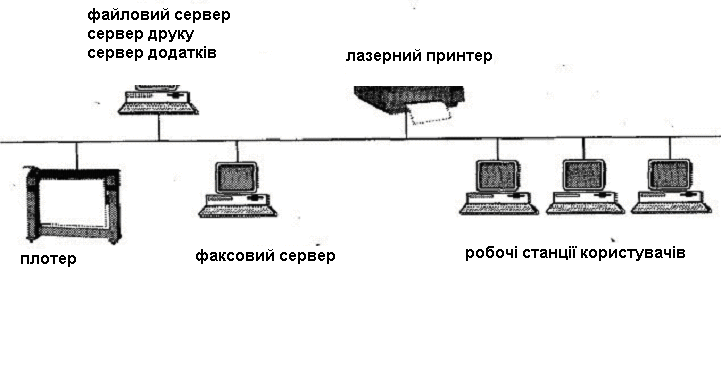
Гібридні обчислення страждають недоліками, характерними для серверних мереж [1].

Класифікації мереж за масштабом виробничого підрозділу, в межах якого діє мережа [1].

Розрізняють мережі відділів, мережі кампусів і корпоративні мережі [1].

Мережі відділів – це мережі, які використовуються порівняно невеликою групою співробітників (100 – 150 чоловік), що працюють в одному відділі підприємства. Головною метою мережі відділу є розділення локальних ресурсів, таких як додатки, дані, лазерні принтери і модеми [1].

Завдання управління мережею на рівні відділу відносно прості: додавання нових користувачів, усунення простих відмов, інсталяція нових вузлів і установка нових версій програмного забезпечення (рис 1.2) [1].

Рисунок 1.2 – Приклад мережі масштабу відділу

Мережі кампусів отримали свою назву від англійського слова campus – с тудентське містечко. Саме на території університетських городків часто виникала необхідність об'єднання декількох дрібних мереж в одну велику мережу. Зараз цю назву пов'язують не тільки із студентськими городками, але використовують і для позначення мереж будь– яких підприємств і організацій [1].

На рівні мережі кампусу виникають проблеми інтеграції неоднорідного апаратного і програмного забезпечення. Типи комп'ютерів, мережевих операційних систем, мережевого апаратного забезпечення можуть відрізнятися в кожному відділі. Звідси витікають складнощі управління мережами кампусів. Адміністратори повинні бути в цьому випадку більш кваліфікованими, а засоби оперативного управління мережею – досконалішими [1].

Корпоративні мережі об'єднують велику кількість комп'ютерів на всіх територіях окремого підприємства. Для корпоративної мережі характерні:

* масштабність – тисячі призначених для користувача комп'ютерів, сотні серверів, величезні об'єми що зберігаються і переданих по лініях зв'язку даних, безліч різноманітних застосувань;
* високий ступінь гетерогенності – типи комп'ютерів, комунікаційного устаткування, операційних систем і додатків різні;
* використання глобальних зв'язків – мережі філій з'єднуються за допомогою телекомунікаційних засобів, зокрема телефонних каналів, радіоканалів, супутникового зв'язку [1].

## **Призначення мережі**

Перше призначення мереж – сумісне використання інформації. Комп'ютери, не підключені до мережі, не дістають ефективного доступу до ресурсів, що розділяються. Мережеві комп'ютери можуть спільно працювати з наступними пристроями:

* факс– модемами;
* сканерами;
* жорсткими дисками;
* накопичувачами на гнучких дисках;
* пристроями читання CD– ROM;
* накопичувачами на магнітній стрічці для резервного копіювання даних;
* графічними пристроями;
* а також з практично будь– якими іншими пристроями, що підключаються до комп'ютерів [2].

Сумісне використання програмних ресурсів. Інсталяція і налаштування конфігурації програмного забезпечення в мережі значно скорочують об'єм роботи, потрібної для забезпечення доступу до комп'ютерних програм у всій організації [2].

Мережа дозволяє виконувати централізоване резервне копіювання інформації. Резервне копіювання – одна з найбільш важливих операцій, що входять в обов'язки адміністратора мережі [2].

Мережа забезпечує важливій корпоративній інформації захищене середовище. При використанні автономних ПК доступ до комп'ютерів часто означає доступ до інформації, що знаходиться в них. Мережі реалізують додатковий рівень захисту за допомогою паролів. Кожному користувачеві, що працює в мережі, можна привласнити окреме облікове ім'я і пароль. В результаті мережевий сервер знатиме, хто до нього звертається, і захистить інформацію, заборонивши несанкціоноване звернення до неї [2].

Електронна пошта. Однією з найбільш значних переваг, що отримуються користувачами від застосування мережі, є електронна пошта (e– mail). Замість обміну повідомленнями, директивами і зауваженнями на папері (що пов'язане з додатковими витратами і затримками) користувачі завжди можуть посилати один одному повідомлення і перевіряти їх отримання [2].

Будь– яка комп'ютерна мережа характеризується своєю архітектурою, яка визначається (рис. 1.3) її топологією, протоколами, інтерфейсами, мережевими технічними і програмними засобами. Кожна із складових архітектури комп'ютерної мережі характеризує її окремі властивості, і тільки їх сукупність характеризує всю мережу загалом (рис 1.4) [2].

Рисунок 1.3 – Компоненти архітектури комп'ютерної мережі



Топологія комп'ютерної мережі відображає структуру зв'язків між її основними елементами. Через низку причин існує відмінність між топологіями глобальних і локальних мереж. Топологія глобальних мереж характеризується достатньо складною, неоднорідною структурою. У свою чергу, топологія локальної мережі, зазвичай, має визначену структуру: лінійну, кільцеву або деревоподібну [2].

Протоколами називають правила взаємодії функціональних елементів мережі [2].

Інтерфейси – це засоби сполучення функціональних елементів мережі. Варто звернути увагу, що у ролі функціональних елементів можуть виступати як окремі пристрої, так і програмні модулі. Відповідно до цього, існують апаратні і програмні інтерфейси [2].

Під мережевими технічними засобами мають на увазі різноманітні пристрої, що забезпечують об'єднання комп'ютерів в єдину комп'ютерну мережу. До цих пристроїв відносяться мережеві контролери, вузли комутації та ін [2].

Мережеві програмні засоби керують роботою комп'ютерної мережі і забезпечують відповідний інтерфейс з користувачами. До мережевих програмних засобів належать мережеві операційні системи і допоміжні (сервісні) програми [2].

## **Основні програмні і апаратні компоненти мережі**

Весь комплекс програмно– апаратних засобів мережі може бути описаний багатошаровою моделлю. У основі будь– якої мережі лежить апаратний шар стандартизованих комп'ютерних платформ. В даний час в мережах широко і успішно застосовуються комп'ютери різних класів – від персональних комп'ютерів до мейнфреймів і супер ЕОМ. Набір комп'ютерів в мережі повинен відповідати набору різноманітних завдань, що вирішуються мережею [3].

Другий шар – це комунікаційне устаткування. Хоча комп'ютери і є центральними елементами обробки даних в мережах, останнім часом не менш важливу роль почали грати комунікаційні пристрої. Кабельні системи, повторювачі, мости, комутатори, маршрутизатори і модульні концентратори з допоміжних компонентів мережі перетворилися на основних разом з комп'ютерами і системним програмним забезпеченням як по впливу на характеристики мережі, так і за вартістю. Сьогодні комунікаційний пристрій може бути складним спеціалізованим мультипроцесором, який потрібно конфігурувати, оптимізувати і адмініструвати [3].

Третім шаром, утворюючим програмну платформу мережі, є операційні системи (ОС). Від того, які концепції управління локальними і розподіленими ресурсами покладені в основу мережевої ОС, залежить ефективність роботи всієї мережі. При проектуванні мережі важливо враховувати, наскільки просто дана операційна система може взаємодіяти з іншими ОС мережі, наскільки вона забезпечує безпеку і захищеність даних, до якого ступеня вона дозволяє нарощувати число користувачів, чи можна перенести її на комп'ютер іншого типу і багато інших міркувань [3].

Самим верхнім шаром мережевих засобів є різні мережеві застосування, такі як мережеві бази даних, поштові системи, засоби архівації даних, системи автоматизації колективної роботи і ін [3].

Таким чином, вибір комп'ютерної мережі може бути зведений до вибору її топології, протоколів, апаратних засобів і мережевого програмного забезпечення. Кожен з цих компонентів є відносно незалежним. Наприклад, мережі з однаковою топологією можуть використовувати різні методи доступу, протоколи і мережне програмне забезпечення. У різних мережах можуть застосовуватись однакові протоколи і (або) мережеве програмне забезпечення. Це розширює можливість вибору найбільш оптимальної архітектури комп'ютерної мережі [3].

## **Види комп’ютерних мереж**

Локальна мережа – найпростіша форма мережі, що сполучає в одну групу комп'ютери або зв'язує їх з потужнішим ПК (наприклад, з мережевим сервером) [4].

Кожний ПК в такій мережі називаються робочою станцією або ж мережевим вузлом [4].

Розрізняють однорангові мережі (рис. 1.4) та мережі з виділеним сервером (рис 1.5) [4].

Рисунок 1.4 – Однорангова мережа

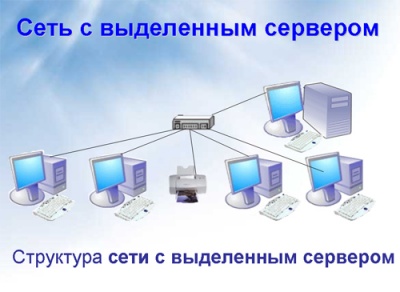


Локальна мережа або LAN об'єднує машини користувачів і принтери, встановлені в межах офісу, одного приміщення або в недалеко розташованих будівлях [4].

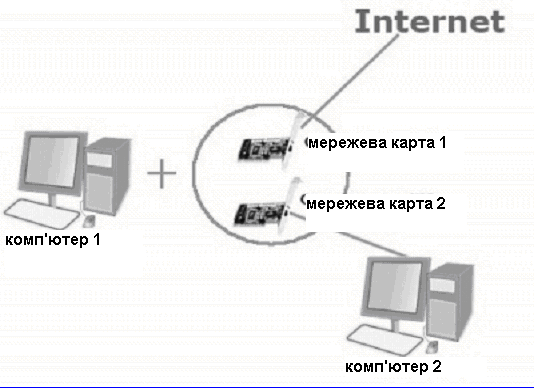
В основному, такі мережі розташовані на відстані не більше 1 км.

Якщо ж відстань треба збільшити, то для цієї мети використовуються спеціальне прилади – репітери (англ. Repeat – повторити) [4].

Рисунок 1.5 – Мережа з виділеним сервером



Комп'ютери, сполучені в одну локальну мережу управляють не лише організацією доступу до мережі, але і корегують процес передачі інформації, надають додаткові послуги і ресурси абонентам мережі (рис 1.6) [4].

Рисунок 1.6 – Процес передачі інформації

Отже, які можливості дає локальна мережа – абсолютно усі співробітники компанії або фірми можуть спілкуватися і взаємодіяти один з одним спільно користуватися загальними ресурсам (у тому числі і периферійними) загальна робота з документами можливість без нарад і зборів переглядати і коментувати документи не покидаючи робоче місце зберігання приватної інформації на мережевому сервері (без використання свого жорсткого диска) доступ до інформації, що зберігається на ПК у віддалених конторах [4].

## **Захист мережі**

На даний час безпека мереж – це дуже актуальна тема інформаційних технологій. Відомий закон Мерфі стверджує: неприємне відбудеться рано чи пізно. Це може бути випадкове чи зумисне, наприклад, з метою промислового шпіонажу руйнування безпеки мережі або будь– яка аварія (псування диска, повінь, пожежа тощо), що призводять до втрати даних, які зберігаються в мережі. Треба відмітити, що стосовно до мережевих технологій безпека ніколи не буває абсолютно гарантованою. Забезпечення необхідного рівня безпеки передбачає застосування комплексу заходів, які направлені на захист комп’ютерів мережі та інформації, що зберігається. Якщо правильно побудувати корпоративну мережу, але упустити питання забезпечення необхідного рівня безпеки – нічого гарного не вийде. У найнесподіваніший момент можна втратити всю інформацію, заради доступу до якої й будувалася мережа [5].

Побудова й підтримка безпечної системи вимагає системного підходу. Відповідно до цього підходу, насамперед, необхідно усвідомити весь спектр можливих загроз для конкретної мережі й для кожної із цих загроз продумати тактику її відбиття. У цій боротьбі можна й потрібно використовувати різнопланові засоби й прийоми – організаційні й законодавчі, адміністративні й психологічні, захисні можливості програмних і апаратних засобів мережі [5].

Законодавчі засоби захисту – це закони, постанови уряду й укази президента, нормативні акти й стандарти, якими регламентуються правила використання й обробки інформації обмеженого доступу, а також уводяться міри відповідальності за порушення цих правил [5].

Адміністративні засоби *–* це дії загального характеру, що вживаються керівництвом підприємства або організації. Адміністрація підприємства повинна визначити політику інформаційної безпеки, що включає відповіді на наступні питання:

* яку інформацію й від кого варто захищати;
* кому і яка інформація потрібна для виконання службових обов’язків;
* який ступінь захисту необхідний для кожного виду інформації;
* чим загрожує втрата того або іншого виду інформації;
* як організувати роботу із захисту інформації [5].

До фізичних засобів захисту відносяться, наприклад, наступні: екранування приміщень для захисту від зовнішнього випромінювання, перевірка апаратури на відповідність специфікаціям і відсутність апаратних "жучків" і т.д [5].

До технічних засобів забезпечення інформаційної безпеки можуть бути віднесені:

* системи контролю доступу, які включають засоби аутентифікації й авторизації користувачів;
* засоби аудиту;
* системи шифрування інформації;
* системи цифрового підпису, які використовуються для автентифікації документів;
* засоби доказу цілісності документів;
* системи антивірусного захисту;
* міжмережеві екрани [5].

Всі зазначені вище засоби забезпечення безпеки можуть бути реалізовані як у вигляді спеціально розроблених для цього продуктів, так і у вигляді вбудованих функцій операційних систем, системних додатків, комп'ютерів і мережевих комунікаційних пристроїв [5].

Варто відзначити, що компанія Cіsco Systems одна з небагатьох на ринку пропонує власну повну архітектуру засобів забезпечення комплексної безпеки CіscoSAFE. Ця система містить великий набір рекомендацій і кращих практик по проектуванню й впровадженню типових вузлів у рамках мережевої інфраструктури [5].

Коротко розглянемо основні компоненти системи CіscoSAFE, що забезпечують безпеку [5].

Політика безпеки компанії. Без цього документа, звичайно, можна створити технічні засоби забезпечення безпеки, але швидше за все не можна реалізувати задумане в повному обсязі. Наприклад, за допомогою технічних засобів, точно не вдасться нівелювати вплив людського фактора, що, між іншим, є одним із надкритичних [5].

Система збору й кореляції інформації. Основне завдання пристроїв цього класу – збір всіх інформаційних службових повідомлень (особливо тих, які прямо пов'язані з безпекою), їх аналіз і кореляція. Унікальність даного класу пристроїв полягає в здатності проводити збір, аналіз і кореляцію декількох тисяч повідомлень і видачу єдиного повідомлення адміністраторові безпеки, для ухвалення рішення [5].

# **POЗДIЛ 2 ПРОЕКТУВАННЯ ЛОКАЛЬНОЇ КОМП’ЮТЕРНОЇМЕРЕЖІ ДЛЯ ПІДПРИЄМСТВА**

## **Організаційна структура підприємства**

Компанія «Ді-Стар» була заснована в 1994 році, з метою зайняти лідируючу позицію на ринку якісного алмазного інструменту. Зараз компанія є найбільшим виробником в Східній Європі з широкою лінійкою інструменту для будівництва та обробки каменю. Алмазний інструмент користується великим попитом зі стабільно зростаючою динамікою обсягів продажів. Вся продукція виготовляється на власних площах підприємства із застосуванням новітніх автоматизованих ліній виробництва Dr. Fritsch GmbH. Логотип компанії (рис. 2.1). Загальний вигляд робочого простору (рис. 2.2). Місце розташування підприємства (рис. 2.3) [6].

  
Рисунок 2.1 – Логотип компанії Distar

  
Рисунок 2.2 – Підприємство Distar

  
Рисунок 2.3 – Розташування підприємства на мапі

Заслужена популярність торгової марки «Ді– Стар» пояснюється високою якістю, оптимальними технічними характеристиками інструменту і різноманітним асортиментом, що включає широку лінійку інструмента:

* алмазні відрізні круги для УШМ по кераміці;
* алмазні відрізні круги для плиткорізів;
* алмазні відрізні круги для різання природного каменю на УШМ і стаціонарному устаткуванні;
* алмазні відрізні круги для різання бетону на УШМ, бензорізи та швонарізчики;
* алмазні відрізні круги для різання асфальту на швонарізчики;
* фрези алмазні сегментні для УШМ і промислових шліфувальних машинах;
* свердла алмазні для ручних електродрилів і свердлильних установок;
* пристосування для підвищення ефективності використання алмазного інструменту [6].

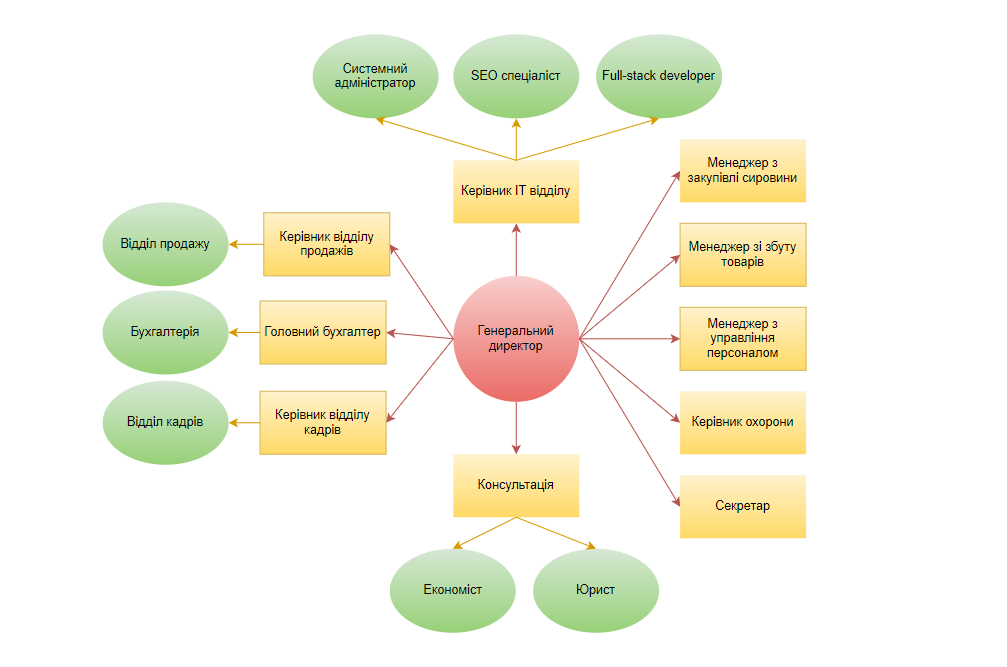
Протягом усієї своєї історії компанія здійснювала політику безперервних інвестицій в розробку і освоєння нових технологій і нових виробів, в закупівлю сучасного високотехнологічного обладнання. Щорічно в технічне переозброєння, розвиток виробничих потужностей та інфраструктури компанії вкладаються інвестиції в розмірі понад один мільйон доларів США. Завдяки цій політиці сьогодні у виробництві алмазного інструменту використовуються найпрогресивніші технології: DIAFIX – орієнтоване розташування алмазних зерен в алмазоносном шарі при виробництві сегментної продукції. Contact Welding – приварювання алмазних сегментів до корпусу фрез в автоматичному циклі. Laser Welding – промінь лазера проварює стик між сегментом і корпусом утворюючи надміцний монолітний шов. Hydrogen Inert Technology – спікання інструменту в захисно– відновних середовищах. Hydrogen Inert Technology Line – спікання інструменту в захисно– відновних середовищах на конвеєрній стрічці. Hot Forged Technology – спікання відбувається в металевих прес– формах при тиску і високих температура в автоматичному циклі. Double Hot Pressing – процес подвійного пресування при високих температурах забезпечує максимальну щільність алмазоносного шару [6].

Персонал Компанії становить понад 200 осіб, що володіють високим рівнем інженерно– технічного і професійної освіти, які об'єднані в єдину команду на основі прихильності Політиці Цінностям і Кодексу Корпоративної культури, націлених на реалізацію довгострокової стратегії технічного і соціального розвитку Компанії [6].

## **Структура підприємства**

Створимо структуру співробітників підприємства, а також розрахуємо кількість та визначимо види техніки для кожного відділу.

Структуру підприємства створено за допомогою програми пакету MS Office – Microsoft Visio (рис.2.4).

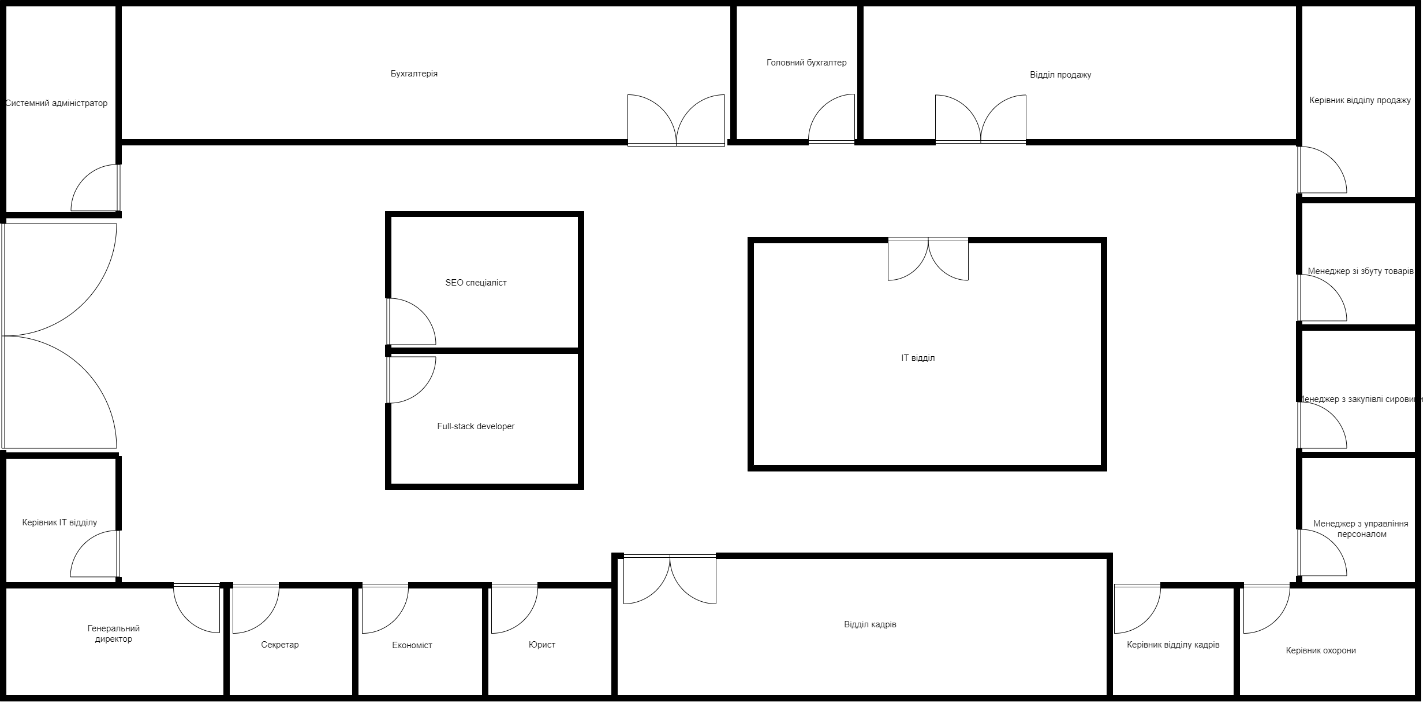
  
Рисунок 2.4 – Структура підприємства

Кількість та види необхідної техніки для кожного відділу показано в табл. 2.1.

Таблиця 2.1 – Кількість та види техніки

|  |  |  |
| --- | --- | --- |
| **Відділ** | **Техніка** | **Кількість** |
| Генеральний директор | Комп’ютер  Принтер | 1  1 |
| Секретар | Комп’ютер  Принтер | 1  1 |
| Економіст | Комп’ютер  Принтер | 1  1 |
| Юрист | Комп’ютер  Принтер | 1  1 |
| Головний бухгалтер | Комп’ютер  Принтер | 1  1 |
| Керівник відділу продажів | Комп’ютер  Принтер | 1  1 |
| Керівник відділу кадрів | Комп’ютер  Принтер | 1  1 |
| Керівник ІТ відділу | Комп’ютер | 1 |
| Менеджер із закупівлі сировини | Комп’ютер  Принтер | 1  1 |
| Менеджер зі збуту товарів | Комп’ютер  Принтер | 1  1 |
| Менеджер з управління персоналом | Комп’ютер  Принтер | 1  1 |
| Керівник охорони | Комп’ютер | 1 |
| Системний адміністратор | Сервер | 1 |
| SEO спеціаліст | Комп’ютер | 1 |
| Full– stack developer | Комп’ютер | 1 |
| Відділ продажу | Комп’ютер  Принтер | 8  2 |
| Відділ кадрів | Комп’ютер  Принтер | 8  1 |
| Бухгалтерія | Комп’ютер  Принтер | 9  2 |
| ІТ відділ | Комп’ютер | 11  1 |
| Всього | Принтер  Комп’ютер  Сервер | 15  50  1 |

На рис. 2.5 представлена схема будівлі підприємства.

  
Рисунок 2.5 – Схема будівлі підприємства

## **Вибір топології мережі**

Топологією КМ називають спосіб з’єднання в ній комп’ютерів (спосіб організації фізичних зв’язків). Іншими словами топологія – це конфігурація графа, вершинам якого відповідають комп’ютери мережі (іноді й інше обладнання, наприклад концентратори), а ребрам – фізичні зв’язки між ними [7].

Вибір топології істотно впливає на ряд характеристик мережі. Наприклад, наявність резервних зв’язків підвищує її надійність і робить можливим балансування завантаження окремих каналів. Простота приєднання нових вузлів, яка властива деяким топологіям, робить КМ легкорозширюваною. Економічні міркування часто приводять до вибору топологій, для яких характерна мінімальна сумарна довжина ЛЗ [7].

Повнозв’язна топологія відповідає мережі, в якій кожний комп’ютер мережі має зв’язки з усіма іншими її комп’ютерами. В загальному випадку це досить громіздкий і неефективний варіант КМ, оскільки потребує велику кількість комунікаційних портів для забезпечення такого зв’язку (кількість ЛЗ у такій КМ буде, n(n– 1)/2, де n – кількість вузлів у мережі). Повнозв’язана топологія застосовуються дуже рідко. Найчастіше вона використовується в багатомашинних комплексах або ГКМ при невеликій кількості комп’ютерів [7].

Всі інші варіанти топологій засновані на неповнозв’язаних структуррах, коли для обміну даними між двома комп’ютерами може бути потрібна проміжна передача даних через інші вузли мережі [7].

Коміркова топологія (mesh) виходить з повнозв’язної шляхом видалення деяких можливих зв’язків. У мережі з комірковою топологією зв’язуються лише ті комп’ютери, між якими відбувається інтенсивний обмін даними. Для обміну даними між комп’ютерами, не сполученими прямими ЛЗ, використовуються транзитні передачі через проміжні вузли. Коміркова топологія допускає з’єднання великої кількості комп’ютерів і характерна, як правило, для ГКМ [7].

Загальна шина донедавна була дуже поширеною топологією для ЛКМ. Тут комп’ютерипідєднуються до одного коаксіального кабелю за схемою «монтажного АБО». Інформація, що надсилається може розповсюджуватися в обидві сторони. Застосування загальної шини знижує вартість проводки, уніфікує підключення різних модулів, забезпечує можливість майже миттєвого широкомовного звернення до всіх станцій мережі [7].

Основними перевагами такої схеми є дешевизна і простота прокладки кабелю по приміщеннях. Основні недоліки – низька надійність (будь– який дефект кабелю або будь– якого з роз’ємів повністю паралізує всю КМ), невисока продуктивність (оскільки в кожний момент часу тільки один комп’ютер може надсилати дані у мережу). Тому пропускна спроможність каналу зв’язку тут завжди ділиться між усіма вузлами мережі [7].

В мережах з кільцевою конфігурацією дані передаються по кільцю від одного комп’ютера до іншого, як правило, в одному напрямку. Якщо комп’ютер розпізнає дані як “свої”, то він копіює їх у свій внутрішній буфер. У мережі з кільцевою топологією слід вживати спеціальних заходів, щоб у разі виходу з ладу або відключення будь– якої станції не розривався канал зв’язку між іншими станціями. Кільце є дуже зручною конфігурацію для організації зворотного зв’язку: дані, зробивши повний обіг, повертаються до вузла– джерела. Тому останній може контролювати процес доставки даних адресату. Часто ця властивість кільця використовується для тестування зв’язаності мережі і пошуку вузла, який працює некоректно. Для цього у мережу посилаються спеціальні тестові повідомлення [7].

В той час коли невеликі КМ, як правило, мають типову топологію зірка, кільце або загальна шина, для великих мереж характерна наявність довільних зв’язків між комп’ютерами. В таких мережах можна виділити окремі довільно зв’язані фрагменти (підмережі), що мають типову топологію, тому їх називають мережами зі змішаною топологією [7].

Ми обираємо топологію – зірка. У цьому випадку кожний комп’ютер підключається окремим кабелем до загального пристрою – концентратора, який знаходиться в центрі мережі і надсилає інформацію, що передається комп’ютером одному або всім іншим комп’ютерам мережі. Головна перевага цієї топології перед загальною шиною – набагато більша надійність (будь– які проблеми з кабелем стосуються лише того комп’ютера, до якого цей кабель приєднаний, і тільки несправність концентратора може вивести з ладу всю КМ). Крім того, концентратор може грати роль інтелектуального фільтра інформації, що надходить від вузлів у мережу, і при необхідності блокувати заборонені адміністратором передачі [7].

До недоліків топології типу зірка відноситься більш висока вартість мережевого обладнання (внаслідок необхідності придбання концентратора). Крім того, можливості по нарощуванню кількості вузлів у мережі обмежуються кількістю портів концентратора. Іноді доцільно будувати мережу з використанням кількох концентраторів, ієрархічно званих між собою у вигляді зірки. В наш час ієрархічна (розширена) зірка є найпоширенішим типом топології зв’язків як в ЛКМ, так і ГКМ [7].

## **Розміщення апаратного забезпечення**

Згідно з державними санітарними правилами та нормами «Влаштування і обладнання кабінетів комп'ютерної техніки» (ДСанШН 5.5.6.009– 98) площа приміщення на одне робоче місце повинна становити 6 м2 , а об'єм – не менше 20 м3 . Площа приміщень з ПК повинна розраховуватись не більш як 12 чоловік [8].

Не дозволяється розміщувати кабінети обчислювальної техніки у підвальних приміщеннях та на цокольних поверхах. Кабінети, обладнані комп'ютерною технікою, повинні розміщуватись в окремих приміщеннях з природним освітленням та організованим обміном повітря [8].

Покриття підлоги, стін, стелі мають бути матовими з коефіцієнтами відбиття відповідно 0,2– 0,3; 0,4– 0,5; 0,7– 0,8; робочого столу 0,4– 0,5; корпусу дисплея та клавіатури 0,3– 0,5; шаф і стелажів 0,4– 0,6 [8].

Поверхня підлоги повинна мати антистатичне покриття і бути зручною для вологого прибирання [8].

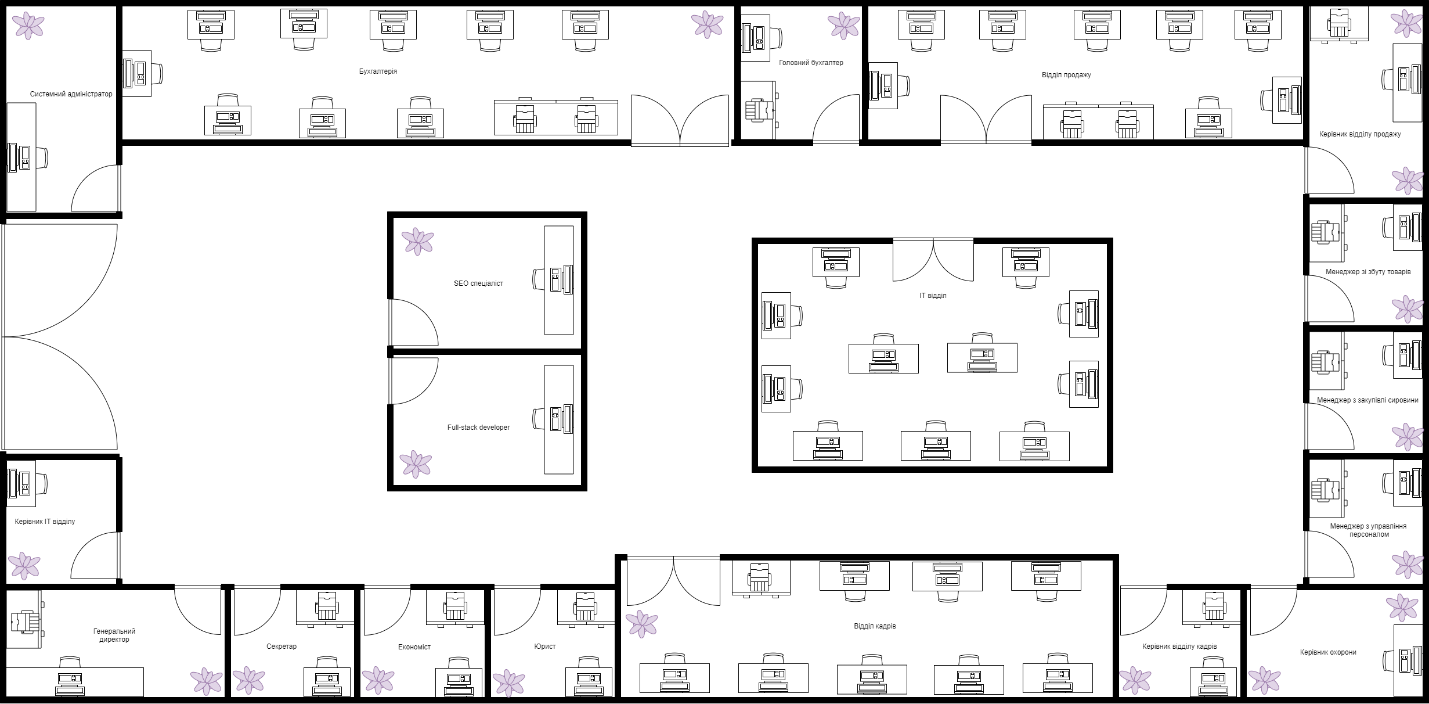
Забороняється застосовувати для оздоблення інтер'єру приміщень комп'ютерних класів полімерні матеріали (деревинно – стружкові плити, шпалери, що миються, рулонні синтетичні матеріали, шаруватий паперовий пластик тощо), які виділяють у повітря шкідливі хімічні речовини, що перевищують гранично допустимі норми [8].

При розташуванні елементів робочого місця користувача ПК слід враховувати робочу позу користувача, простір для розміщення користувача, можливість огляду елементів робочого місця, можливість ведення записів, розміщення документації і матеріалів, які використовуються користувачем [8].

Робочі місця з ПК повинні бути розташовані від стіни з вікнами на відстані не менш 1,5 м, від інших стін – на відстані 1 м, відстань між ними має становити не менш ніж 1,5 м [8].

Апаратне забезпечення розміщене згідно вимог та державних санітарних правил та норм.

Схема розміщення апаратного забезпечення на підприємстві представлена на рис. 2.6.

  
Рисунок 2.6 – Схема розміщення апаратного забезпечення

## **Облікові записи користувачів**

Обліковий запис користувача – це набір налаштувань і відомостей, які повідомляють Windows, які файли і папки ви можете отримати, що ви можете зробити на вашому комп'ютері, які ваші налаштування і які мережеві ресурси можна отримати, коли ви підключені до мережі [9].

Ідентифікація – процедура, в результаті виконання якої для суб'єкта ідентифікації виявляється його ідентифікатор, однозначно визначає цього суб'єкта в інформаційній системі [10].

Аутентифікація – процедура перевірки автентичності, наприклад перевірка справжності користувача шляхом порівняння введеного ним пароля з паролем, збереженим в базі даних [10].

Авторизація – надання певній особі або групі осіб прав на виконання певних дій [10].

Визначимо облікові записи для кожного користувача (табл. 2.2).

Таблиця 2.2 – Облікові записи користувачів

|  |  |  |
| --- | --- | --- |
| **Користувач** | **Логін** | **Пароль** |
| Генеральний директор | Gd\_1 | 1\_gd |
| Секретар | Sk\_2 | 2\_sk |
| Економіст | Ek\_3 | 3\_ek |
| Юрист | Lw\_4 | 4\_lw |
| Головний бухгалтер | Gb\_5 | 5\_gb |
| Керівник відділу продажів | Vp\_6 | 6\_vp |
| Керівник відділу кадрів | Vk\_7 | 7\_vk |
| Керівник ІТ відділу | It\_8 | 8\_it |
| Менеджер із закупівлі сировини | Zs\_9 | 9\_zs |
| Менеджер зі збуту товарів | Zt\_10 | 10\_zt |
| Менеджер з управління персоналом | Up\_11 | 11\_up |
| Керівник охорони | Ko\_12 | 12\_ko |
| Системний адміністратор | Sa\_13 | 13\_sa |

Продовження таблиці 2.2

|  |  |  |
| --- | --- | --- |
| SEO спеціаліст | Seo\_14 | 14\_seo |
| Full– stack developer | Fsd\_15 | 15\_fsd |
| Відділ продажу | Vp\_16  Vp\_17  Vp\_18  Vp\_19  Vp\_20  Vp\_21  Vp\_22  Vp\_23 | 16\_vp  17\_vp  18\_vp  19\_vp  20\_vp  21\_vp  22\_vp  23\_vp |
| Відділ кадрів | Vk\_24  Vk\_25  Vk\_26  Vk\_27  Vk\_28  Vk\_29  Vk\_30  Vk\_31 | 24\_vk  25\_vk  26\_vk  27\_vk  28\_vk  29\_vk  30\_vk  31\_vk |
| Бухгалтерія | Bh\_32  Bh\_33  Bh\_34  Bh\_35  Bh\_36  Bh\_37  Bh\_38  Bh\_39  Bh\_40 | 32\_bh  33\_bh  34\_bh  35\_bh  36\_bh  37\_bh  38\_bh  39\_bh  40\_bh |
| ІТ відділ | It\_41  It\_42  It\_43  It\_44  It\_45  It\_46  It\_47  It\_48  It\_49  It\_50  It\_51 | 41\_it  42\_it  43\_it  44\_it  45\_it  46\_it  47\_it  48\_it  49\_it  50\_it  51\_it |

## **Апаратне і програмне забезпечення та вартість мережі**

Характеристики апаратного забезпечення показано в табл. 2.3, а також вартість даного програмного забезпечення показано в табл. 2.4.

Таблиця 2.3 – Характеристики апаратного забезпечення

|  |  |
| --- | --- |
| **Найменування** | **Характеристики** |
| Системний блок Dell Vostro 3668+клавіатура+миша | Процесор: Чотириядерний Intel Core i7– 7700; Обсяг оперативної пам'яті: 8 ГБ; Відеокарта: Інтегрована, Intel HD Graphics 630; Чипсет материнської плати: Intel H110; Обсяг HDD: HDD: 1 ТБ (7200 об/хв); Порти: 2 x USB 3.0; 1 x аудіороз'єм; Кардридер; 4 x USB 2.0; 1 x LAN (RJ– 45); 1 x HDMI; 1 x VGA; Потужність БЖ: 240 Вт; Бездротові технології: Wi– Fi 802.11ac; Bluetooth 4.0; Охолодження: BOX; Тип пам'яті: DDR4 (1 х 8 ГБ) (2 слоти, макс. 16 ГБ); Оптичний привод: DVD+/– RW [11];] |
| Монітор 23.8" Acer Nitro VG240Ybmiix | Діагональ дисплея: 23.8"; Максимальна роздільна здатність дисплея: 1920 x 1080; Тип матриці: IPS; Інтерфейси: 2 x HDMI; VGA; Яскравість дисплея: 250 кд/м²; Вбудовані колонки: Є; Максимальна кількість кольорів: 16.7 млн. [11]; |
| Принтер HP LaserJet Pro M426fdw | Технологія друку: Лазерний друк (кольоровий); Максимальна роздільна здатність друку: 600x600 dpi; Тип пристрою: БФП; Кількість кольорів: 4; РК– монітор: 4.3" кольоровий сенсорний графічний дисплей; Інтерфейс: 1 x USB 2.0; 1 x мережевий порт Fast Ethernet 10/100Base– TX [11]; |
| Сервер HP Proliant DL380p G8 | HP Proliant DL380p G8; 2 x XEON 4 Core E5– 2609 2.4 GHz; 32GB DDR3; 8 слотів для дисків 2.5" SATA/SAS; Smart Array P420 / P420i [12]; |

Продовження таблиці 2.3

|  |  |
| --- | --- |
| Комплект бездротовий Esperanza Titanum Memphis TK108UA | Клавіатура: Тип клавіш клавіатури: мембранний; Бездротове під'єднання через USB– приймач; Миша: Кількість кнопок: 3; Тип датчика: оптичний; Бездротове під'єднання через USB– приймач [11]; |
| Жорсткий диск для сервера | Виробник: Seagate; Модель: 6TB; Артикул: ST6000NM0095; Інтерфейс: SAS;  Об'єм: 6TB; Швидкість обертання шпинделя: 7200 об/хв; Розмір буферу: 256MВ; Форм– фактор: 3,5" [13]; |

Таблиця 2.4 – Кількість та види апаратного забезпечення

|  |  |  |  |
| --- | --- | --- | --- |
| **Найменування** | **Кількість** | **Вартість за 1 од., грн** | **Вартість загальна, грн** |
| Системний блок | 50 | 11 035 | 551 750 |
| Монітор | 51 | 4 700 | 239 700 |
| Принтер | 15 | 13 569 | 203 535 |
| Сервер | 1 | 18 678 | 18 678 |
| Жорсткий диск для серверу | 8 | 6 175 | 49 400 |
| Комплект миша і клавіатура | 1 | 299 | 299 |
| Всього, грн | 1 063 362 | | |

Вартість мережевого обладнання для комп’ютерної мережі розрахована у таблиці 2.5.

Комутатор локальної мережі — пристрій, призначений для з'єднання декількох вузлів комп'ютерної мережі в межах одного сегмента [14].

Switch Zyxel GS1900 – 24E

Швидкість передачі даних: 1000 Мбит/с

Роз'єми: 24хRJ– 45 [15].

Вита пара – вид мережевого кабелю, з однією або декількома парами ізольованих провідників, скручених між собою (з невеликою кількістю витків на одиницю довжини) для зменшення взаємних наведень при передачі сигналу і покритих пластиковою оболонкою. Кабель приєднується до мережевих пристроїв за допомогою з'єднувача RJ– 45. Підтримує передачу даних на відстань біля 100 метрів [16].

RJ – 45 – фізичний інтерфейс, який загалом використовується для з'єднання комп'ютерних мереж за допомогою звитої пари через мережевий комутатор, або при створенні мережі з двох комп'ютерів до один одного через мережеву карту [17].

Таблиця 2.5 – Вартість мережевого обладнання

|  |  |  |  |
| --- | --- | --- | --- |
| **Найменування** | **Кількість** | **Вартість за 1 од., грн** | **Вартість загальна, грн** |
| Комутатор локальної мережі | 3 | 4 955 | 14 865 |
| Кабель | 700 м | 12 | 8 400 |
| Конектор, роз'єм RJ– 45 | 300 | 2,50 | 750 |
| Всього, грн | 24 006 | | |

Оберемо та розрахуємо вартість програмного забезпечення для комп’ютерної мережі (табл. 2.6).

Таблиця 2.6 – Вартість програмного забезпечення

|  |  |  |  |
| --- | --- | --- | --- |
| **Найменування** | **Кількість** | **Вартість за 1 од., грн** | **Вартість загальна, грн** |
| Microsoft Windows 10 Enterprise (корпоративна) | 1 | 7 800 | 7 800 |
| Microsoft Office 2016 (коробкова версія) | 50 | 7 000 | 350 000 |
| Microsoft Windows Server 2019 Standart | 1 | 25 899 | 25 899 |
| 1С Бухгалтерія для України | 10 | 6 690 | 66 900 |
| JetBrains PhpStorm комерційна | 1 | 5 453 | 5 453 |
| Всього, грн | 456 052 | | |

Підрахуємо кінцеву вартість розробки комп’ютерної мережі для підприємства (табл. 2.7).

Таблиця 2.7 – Вартість розробки мережі

|  |  |
| --- | --- |
| **Складова** | **Вартість, грн** |
| Програмне забезпечення | 456 052 |
| Апаратне забезпечення | 1 063 362 |
| Мережеве обладнання | 24 006 |
| Всього | 1 543 420 |

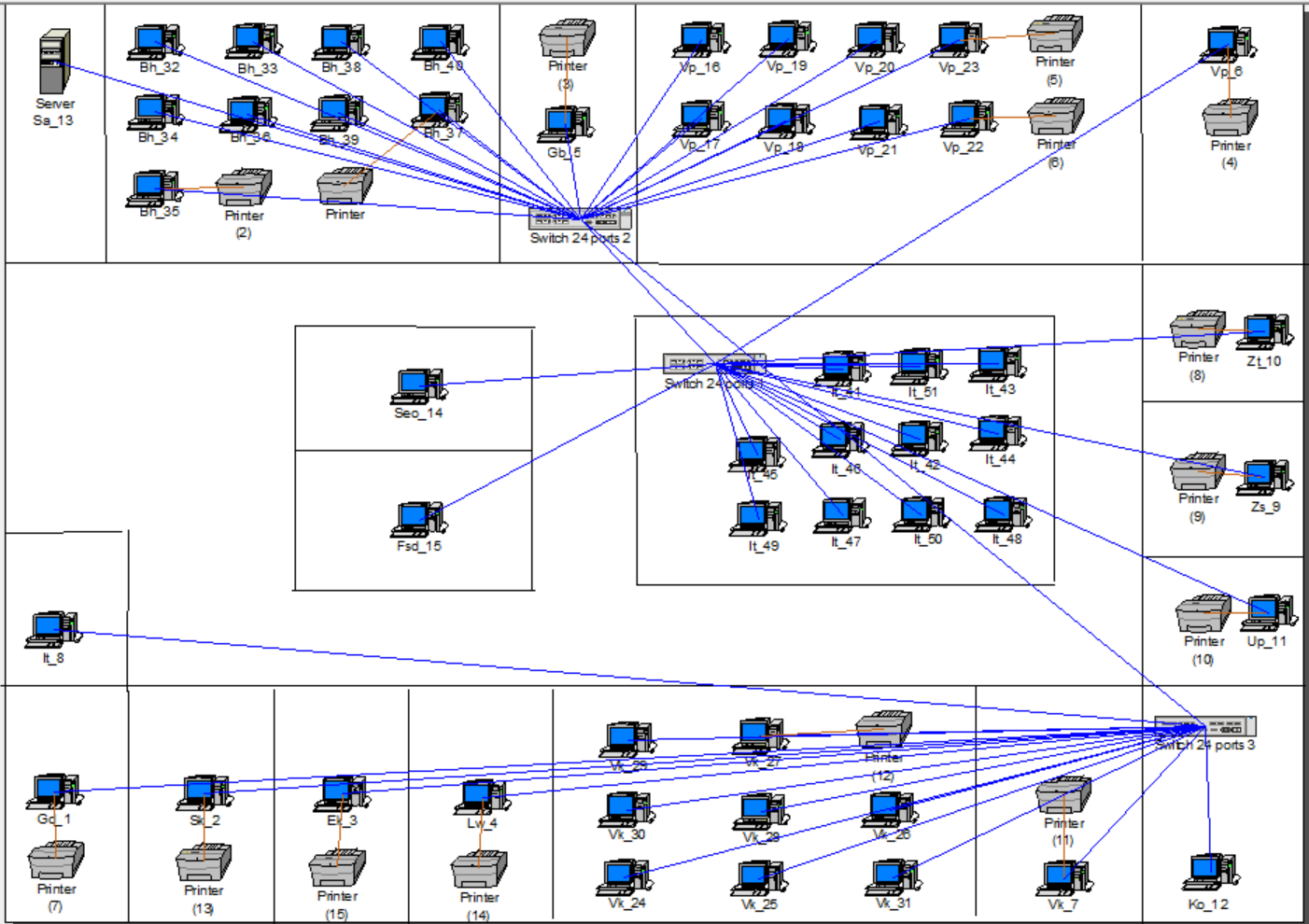
## **Перевірка коректності роботи комп’ютерної мережі**

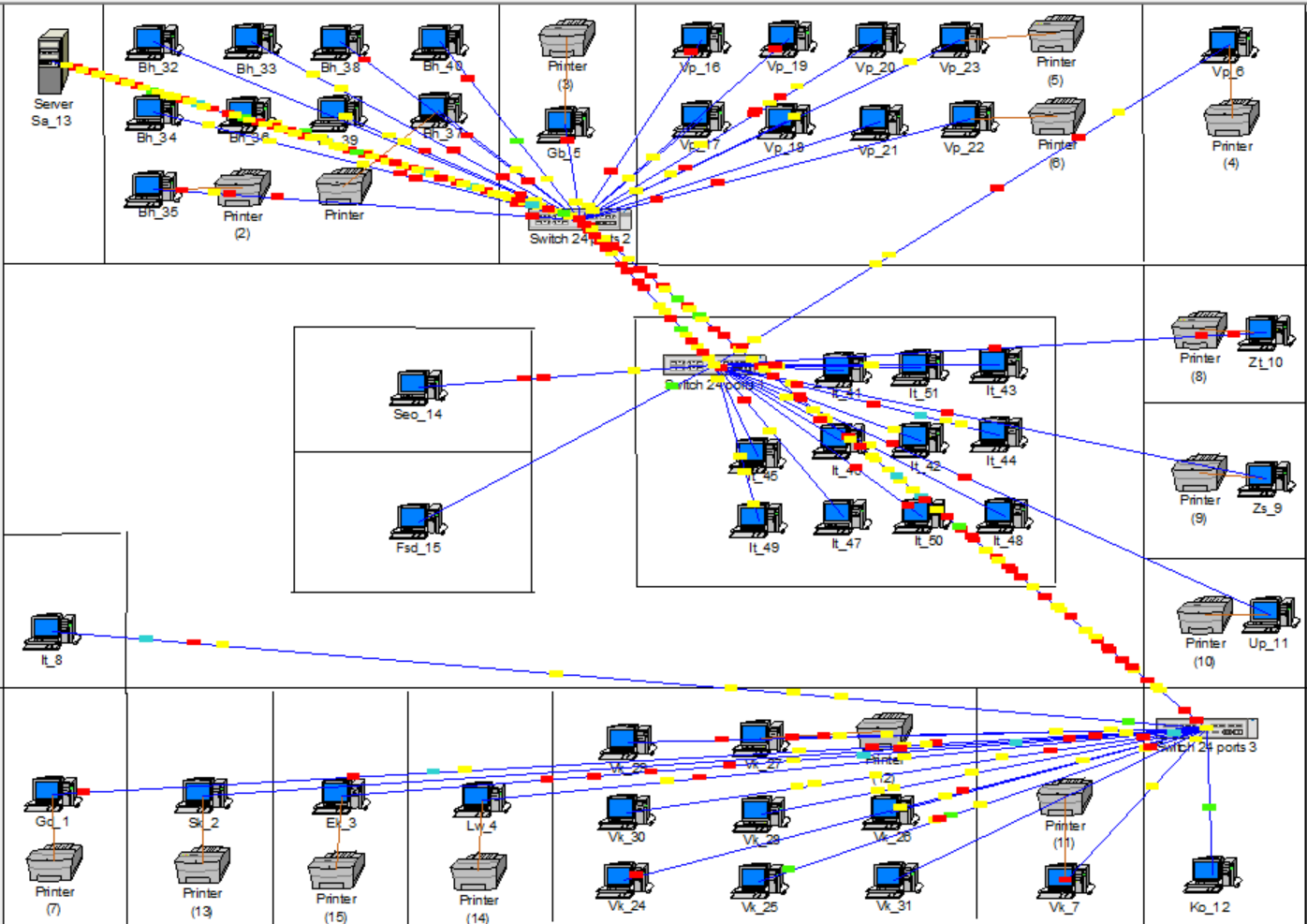
Виконаємо перевірку комп’ютерної мережі у програмі NetCracker.

NetCracker – це система яка представляє собою CASE – ресурси автоматизованого проектування, моделювання та аналізу комп'ютерних мереж. Дозволяє провести дослідження, результати яких можуть бути використані для обгрунтування вибору типу мережі, середовищ передачі, мережевих складових обладнання та програмно– математичного забезпечення [18].

Програмні ресурси NetCracker дають можливість здійснити збір відповідних даних про існуючу мережу без зупинки її роботи, створити проект цієї мережі та виконати необхідні дослідження для визначення граничних характеристик, можливості розширення, зміни топології та модифікації мережевого обладнання з ціллю подальшого її вдосконалення та розвитку [18].

1. Розташовуємо комп'ютери та сервери згідно з планом (рис. 2.7).
2. У кабінетах – головного бухгалтера, керівника охорони та ІТ відділі розташовуємо коммутатори.
3. Встановлюємо на всі ПК мережеві карти – Fast Ethernet adapters.
4. Ставимо драйвера на сервер: File Server, SQL Server, FTP Server та HTTP Server.
5. Налаштовуємо трафік між комп’ютером та сервером: File, FTP, HTTP та SQL.
6. Запускаємо роботу локальної мережі, перевіряємо правильність роботи. (рис. 2.8)

  
Рисунок 2.7 – Схема розташування апаратного забезпечення

  
Рисунок 2.8 – Перевірка роботи комп’ютерної мережі

# **POЗДIЛ 3 ПРАКТИЧНА ЧАСТИНА РОЗРОБКА ІНФОРМАЦІЙНОГО ЗАХИСТУ**

## **Криптологічний захист інформації**

Криптологія – наука про захист інформації, шляхом її перетворення. Криптологія поєднує два напрямки – криптографію й криптоаналіз. Слід зазначити, що ці дві науки – парна категорія. Розвиток однієї з них є поштовхом для іншої. Розглядати окремо криптографію від криптоаналізу, значить порушити основи філософії й один з її законів єдності й боротьби протилежностей [19].

Криптографія займається пошуком і дослідженням методів перетворення інформації з метою приховання її змісту. Основні напрямки використання криптографічних методів – передача конфіденційної інформації з каналів зв'язку, установлення дійсності переданих повідомлень, зберігання інформації (документів, баз даних) на носіях у зашифрованому виді [19].

Криптосистеми підрозділяються на симетричні й асиметричні (або з відкритим ключем) [19].

У симетричних криптосистемах для шифрування й для розшифровування використовується один і той самий ключ [19].

У системах з відкритим ключем використовуються два ключі– відкритий і закритий (секретний), які математично зв'язано один з одним. Інформація шифрується за допомогою відкритого ключа, який доступний усім бажаючим, а розшифровується за допомогою закритого ключа, відомого тільки одержувачеві повідомлення [19].

Терміни «розподіл ключів» і «керування ключами» ставляться до процесів системи обробки інформації, змістом яких є вироблення й розподіл ключів між користувачами [19].

Електронним цифровим підписом називається його криптографічне перетворення, що приєднується до тексту, яке дозволяє при одержанні тексту іншим користувачем перевірити авторство й дійсність повідомлення [19].

Криптологічною стійкістю називається характеристика шифру, що визначає його стійкість до розшифровування без знання ключа (тобто криптоаналізу). Є кілька показників криптостійкости, серед яких:

* кількість усіх можливих ключів;
* середній час, необхідний для успішної криптоаналитичної атаки того або іншого виду.

Ефективність шифрування з метою захисту інформації залежить від збереження таємниці ключа й криптостійкости шифру [19].

## **Вибір та обґрунтування використання алгоритму шифрування XOR**

Шифр XOR – це алгоритм шифрування даних з використанням виключної диз’юнкції [20].

Набув широкого застосування у комп'ютерних мережах [90– х років](https://uk.wikipedia.org/wiki/1990-%D1%82%D1%96) у зв'язку зі простотою реалізації. Застосовувався для шифрування документів [Microsoft Word](https://uk.wikipedia.org/wiki/Microsoft_Word) в середовищі [Windows](https://uk.wikipedia.org/wiki/Windows_95) [21].

Як логічну операцію, XOR також називають подвійним додаванням модуля. В операції XOR вихід є дійсним, коли входи відрізняються. Іншими словами, операція XOR означає "або одну, але не обидві, або жодну" [22].

Алгоритм XOR шифрування полягає в "накладенні" послідовності випадкових чисел на текст, який необхідно зашифрувати. Послідовність випадкових чисел називається гамма– послідовність, і використовується для шифрування і розшифрування даних [22].

Формула для отримання закодованого тексту:

Ключ шифрування можна отримати двома способами:

* Повторювати ключове слово поки довжина гами не дорівнюватиме довжині повідомлення;
* Згенерувати послідовність псевдовипадкових чисел, що дорівнює по довжині тексту повідомлення [22].

Для більшої криптостійкості, можна генерувати гамма– послідовність з використанням псевдовипадкових чисел, використовуючи цілочисельний ключ [20].

Коли шифр XOR має випадковий ключ, довжиною якого є повідомлення, зламати його неможливо. Іншими словами, він пропонує найвищий рівень безпеки [22].

## **Вибір та обґрунтування використання мови програмування C# для розроблення програмного продукту**

C # (вимовляється Сі– Шарп) – мову програмування, що поєднує об'єктно– орієнтовані і контекстно– орієнтовані концепції. Розроблено в 1998– 2001 роках групою інженерів під керівництвом Андерсa Хейлсбергa в компанії Microsoft як основна мова розробки додатків для платформи Microsoft .NET. Компілятор з C # входить в стандартну установку самої .NET, тому програми на ньому можна створювати і компілювати навіть без інструментальних засобів на кшталт Visual Studio [23].

Мова програмування C# переважно використовується для створення корпоративного програмного забезпечення, фінансових проектів, наприклад для банків і бірж, зокрема мобільних додатків, хмарних сервісів [24].

C# у порівнянні з Java легше взаємодіє, з кодом програм, написаних на інших мовах. І саме на C# часто пишуться розширення для інших мов програмування, які використовуються у якості прошарку між бібліотекою C# і мовою, можливості якої під конкретні цілі планується розширювати [24].

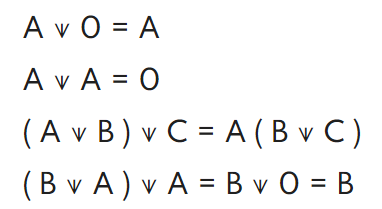
Що стосується порівняння мов програмування, слід відзначити, мова програмування C# (C Sharp) високорівнева, це означає, що вона дещо схожа на англійську. Мова програмування C# має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції– члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Перейнявши багато що від своїх попередників — мов С++, Delphi, Модула і Smalltalk — у С#, опираючись на практику їхнього використання, навмисне виключили деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем у вище перелічених мовах програмування [24].

С# керується чіткими строгими правилами використання пріоритетів, а значить абстрагує програмістів початківців робити дурні помилки і навчатися її, смаживши свій мозок [24].

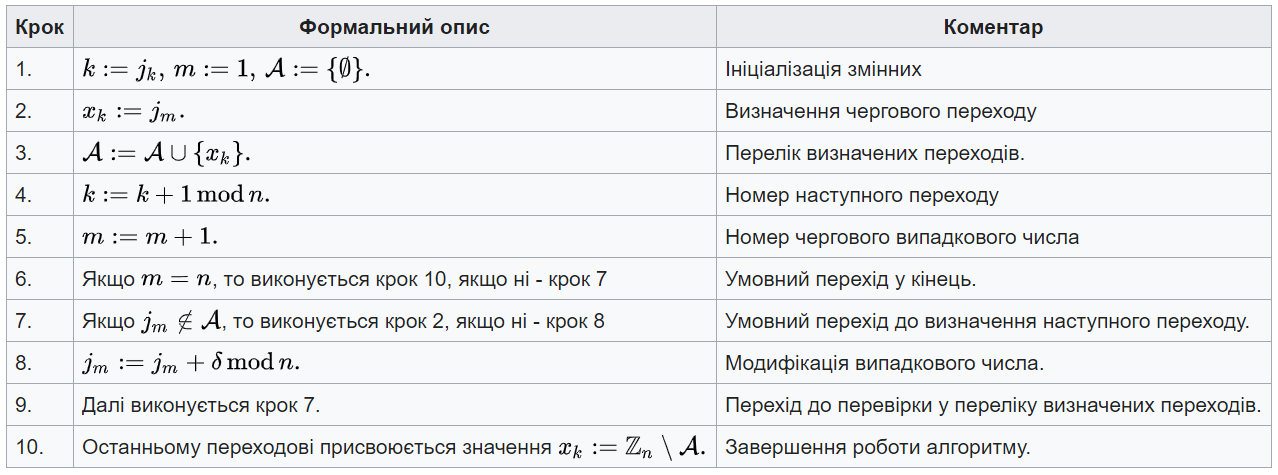
Синтаксис досить мінімалістичний C#. З ручним управлінням пам’яттю [24].

## **Програмна реалізація**

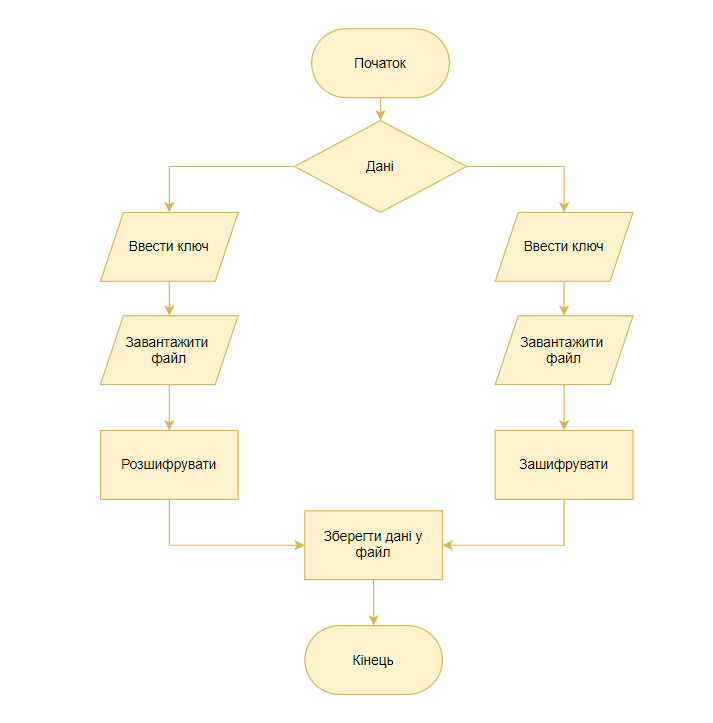
* + 1. **Принцип роботи шифрування XOR**. Шифр XOR використовує логічну операцію XOR для шифрування даних. Спочатку генерується випадковий ключ. Потім операція XOR виконується за допомогою ключа, щоб створити зашифровані дані. Для дешифрування слід використовувати той самий ключ, а операцію XOR потрібно запустити знову. Операція XOR використовує той самий ключ як для шифрування, так і для дешифрування. Ось чому воно відоме як симетричне шифрування. Приклад роботи операції XOR представлено на рис. 3.1 (⩛ позначає операцію XOR) [22].

  
Рисунок 3.1 – Приклад роботи операції XOR

* + 1. **Алгоритм роботи шифру XOR.** Для формального опису алгоритму використовується оператор Бекуса (рис. 3.2) присвоєння значення деякої змінної :=, множина сформованих переходів позначена через A [21].

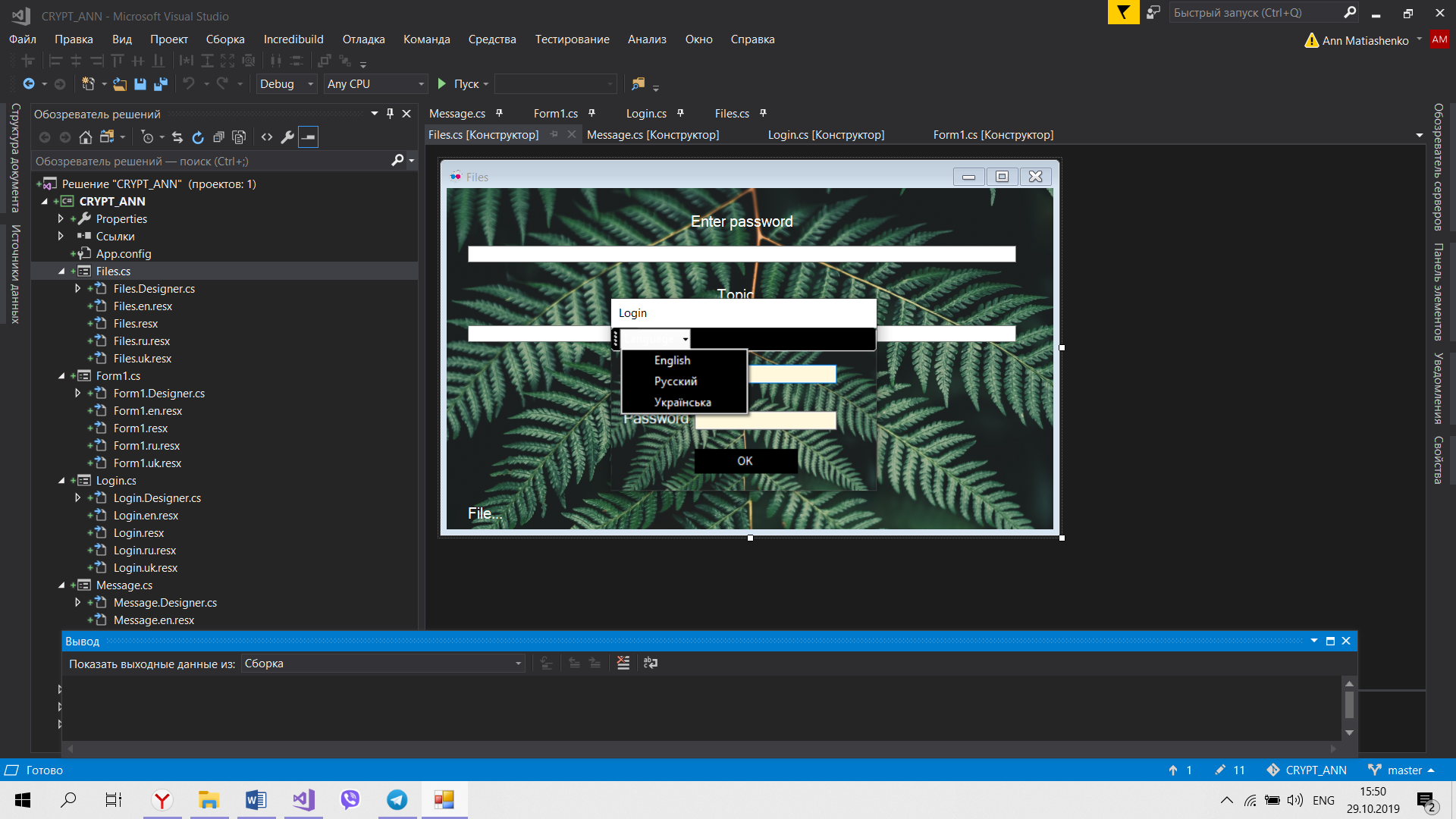
  
Рисунок 3.2 – Формальний опис роботи алгоритму шифрування XOR

* + 1. **Блок – схема.** Алгоритм роботи програми представлено за допомогою блок – схеми на рис. 3.3.

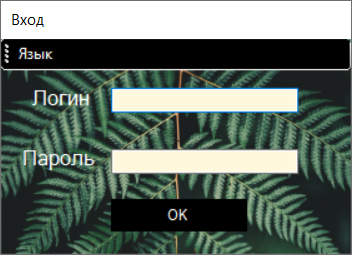
  
Рисунок 3.3 – Блок – схема алгоритму роботи програми

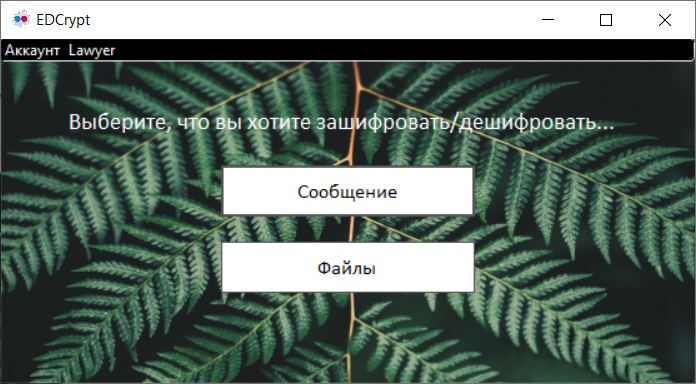
* + 1. **Опис та демонстрація роботи програмної частини.** Для криптологічного захисту інформації підприємства створено програмне забеспечення, що виконує такі функції:
* Вхід за обліковим записом;
* Неможливість користування системою без входу за обліковим записом;
* Повідомлення про введення некоректних даних про користувача;
* Шифрування повідомлень;
* Розшифрування повідомлень;
* Можливість вибору отримувача повідомлення;
* Шифрування текстових файлів;
* Розшифрування тектових файлів;
* Розсилання тектових файлів вибраним отримувачам;
* Три мови інтерфейсу: Англійська, Російська, Українська.

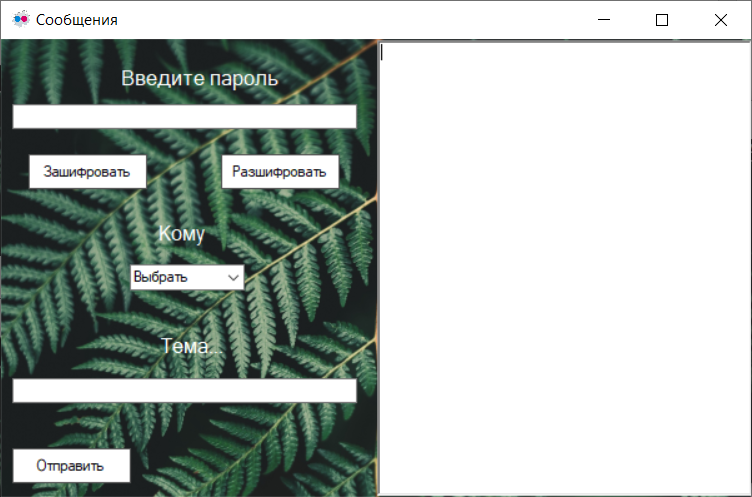
Далі розглянемо принцип роботи за етапами. Отже, після запуску програми ми бачимо вікно входу до системи, якщо не ввести пароль та логін програма не дозволить продовжити роботу. Також на початку роботи з програмою можна вибрати мову інтерфейсу – англійська, російська або українська (рис. 3.4).

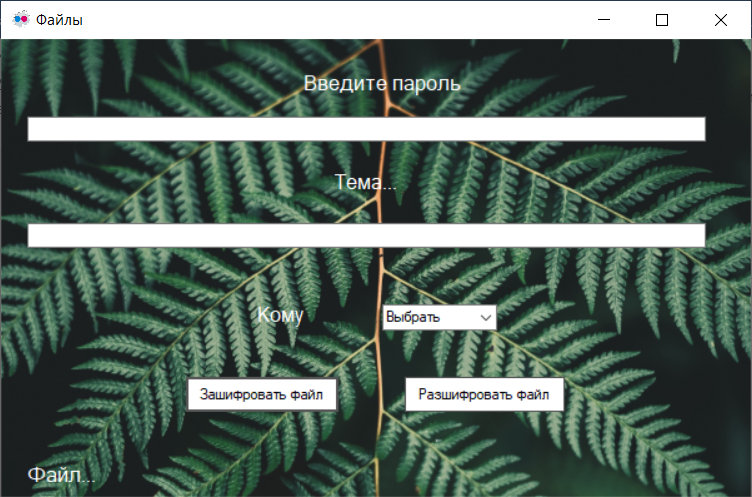
  
Рисунок 3.4 – Вибір мови інтерфейсу.

На рис. 3.5 – 3.8 показано інтерфейс російської версії.

  
Рисунок 3.5 – Вікно входу російського інтерфейсу.

  
Рисунок 3.6 – Вікно меню російського інтерфейсу.

  
Рисунок 3.7 – Вікно повідомлення російського інтерфейсу.

  
Рисунок 3.8 – Вікно файли російського інтерфейсу.

На рис. 3.9 – 3.12 показано інтерфейс російської версії.

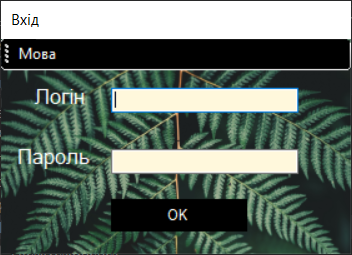
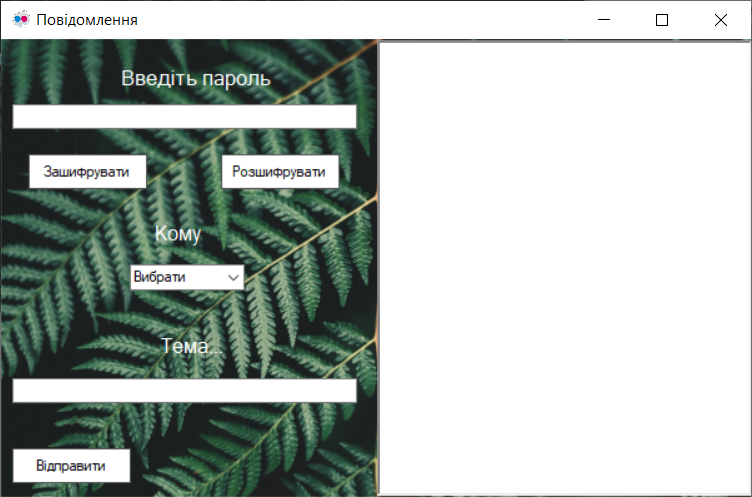
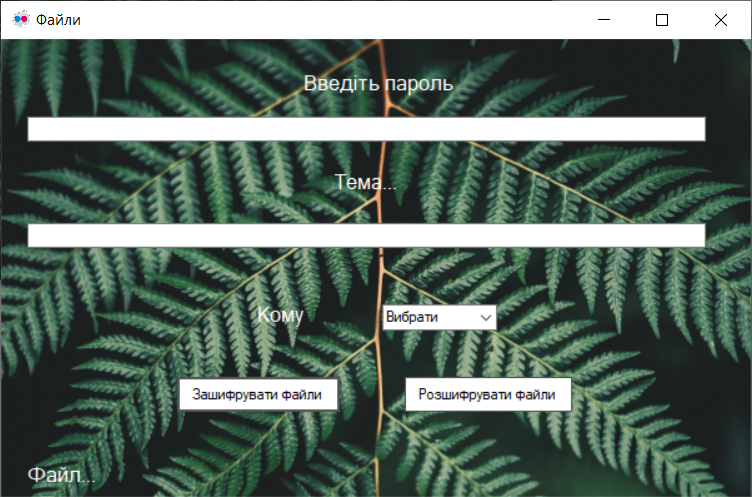
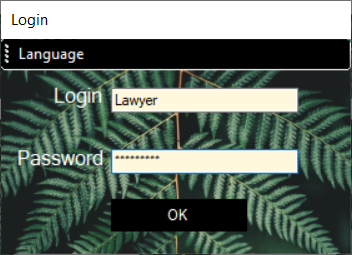
  
Рисунок 3.9 – Вікно входу українського інтерфейсу.

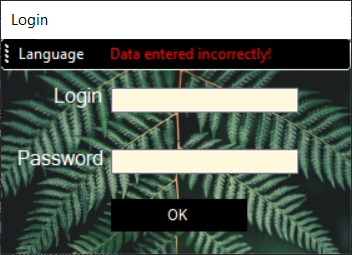
  
Рисунок 3.10 – Вікно меню українського інтерфейсу.

  
Рисунок 3.11 – Вікно повідомлення українського інтерфейсу.

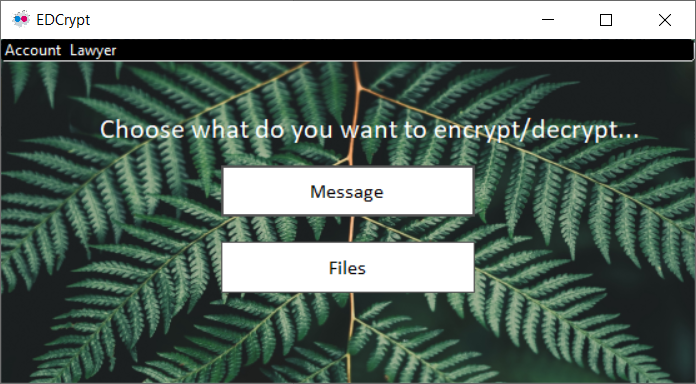
  
Рисунок 3.12 – Вікно файли українського інтерфейсу.

Приклад роботи програми буде показано з англійською версією інтерфейсу. Спершу входимо в систему (рис. 3.13). Якщо логін та/або пароль введено не вірно то програма повідомляє про це та видаляє введені дані про користувача (рис. 3.14).

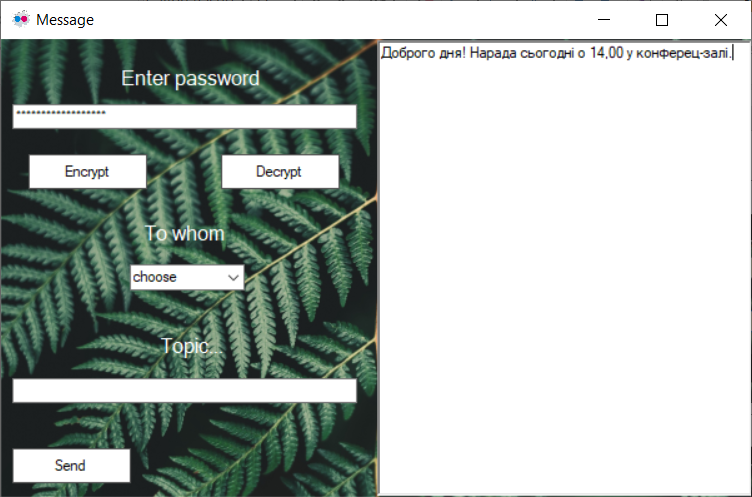
  
Рисунок 3.13 – Вхід до системи.

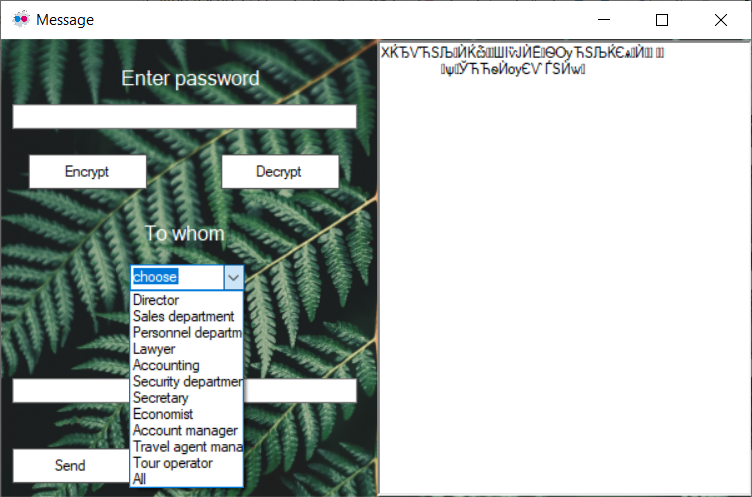
  
Рисунок 3.14 – Відмова у вході в систему.

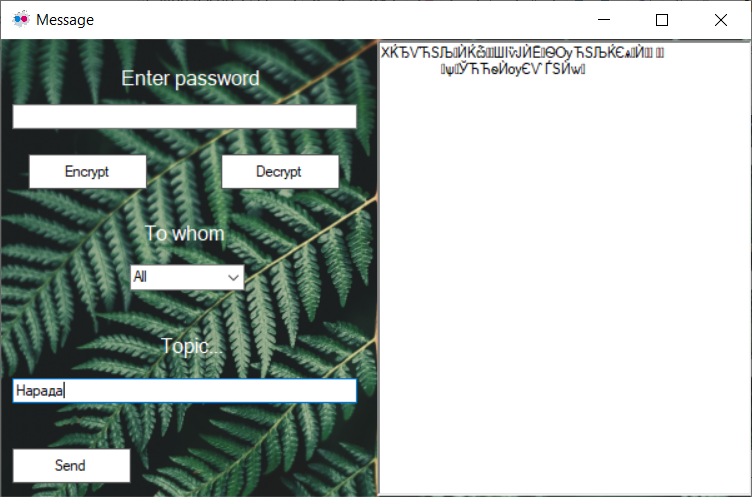
Після входу в систему, у головному меню бачимо свій логін, обліковий запис з якого ми будемо виконувати певні дії. Обираємо, що хочемо виконати надіслати повідомлення чи файли (рис. 3.15).

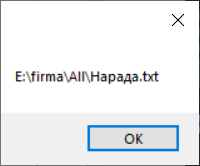
  
Рисунок 3.15 – Головне меню.

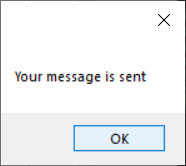
Щоб надіслати повідомлення потрібно лише обрати кому треба надіслати повідомлення, ввести пароль для шифрування даних, ввести сам текст повідомлення та тему повідомлення. Аналогічно для розшифрування даних, треба лише ввести зашифрований текст і відправити потрібному користувачу, наприклад собі (рис. 3.16 – 3.23).

  
Рисунок 3.16 – Надсилання повідомлення.

  
Рисунок 3.17 – Вибір отримувача.

  
Рисунок 3.18 – Відправка зашифрованого повідомлення.

  
Рисунок 3.19 – Шлях надісланого повідомлення

  
Рисунок 3.20 – Повідомлення про відправку.

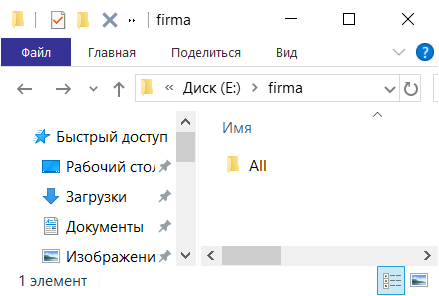
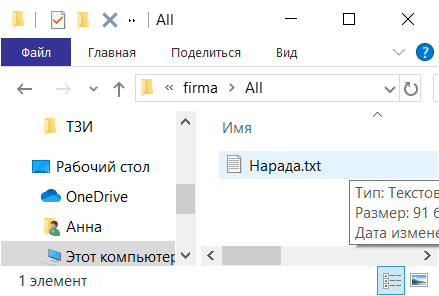
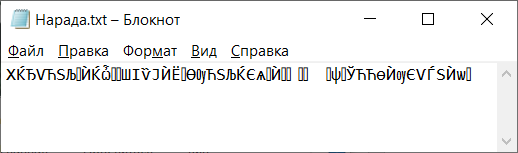
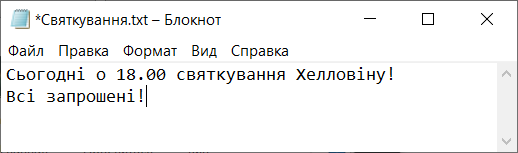


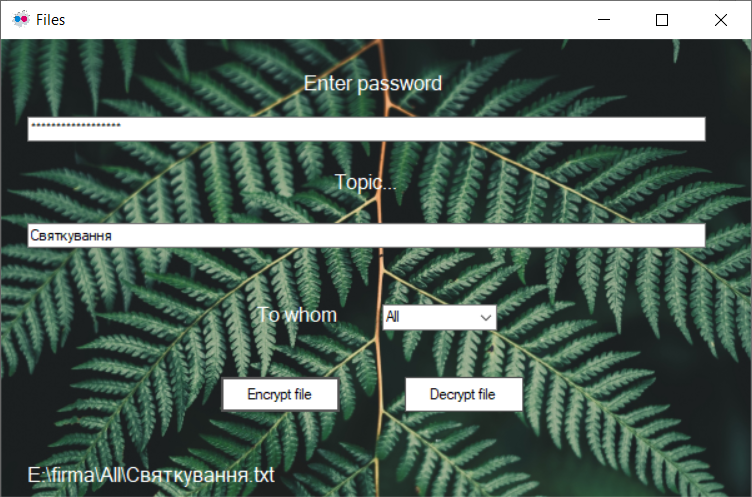
Рисунок 3.21 – Папка створена програмою.

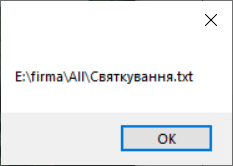
  
Рисунок 3.22 – Надіслане повідомлення.

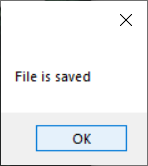
  
Рисунок 3.23 – Вміст надісланого повідомлення.

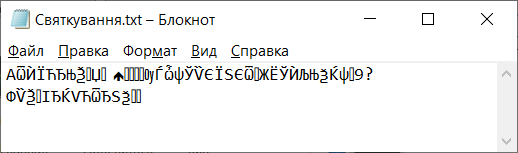
Для шифрування текстових файлів, потрібно ввести пароль для шифрування, обрати шлях для збереження файлу, ввести тему або назву файлу та натиснути на потрібну кнопку, потім завантажити файл, він зашифровується або розшифровується автоматично та зберігається в залежності від вибраного шляху (рис. 3.24 – 3.29).

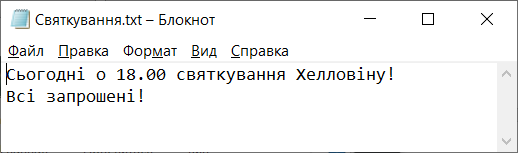
  
Рисунок 3.24 – Вміст файлу до шифрування.

  
Рисунок 3.25 – Завантаження та шифрування файлу.

  
Рисунок 3.26 – Шлях збереженого файлу.

  
Рисунок 3.27 – Повідомлення про збереження файлу.

  
Рисунок 3.28 – Вміст файлу після шифрування.

  
Рисунок 3.29 – Вміст файлу після розшифрування.

Якщо, довжина ключа близька до довжини повідомлення, це забезпечить максимальну криптостійкість. Приклад лістингу коду представлено в додатку А.

# **ВИCНOВКИ**

В ході виконання кваліфікаційної роботи бакалавра було проведено аналіз потреб підприємства та розроблено проект локальної мережі для будівлі підприємства Ді-Стар. Обране необхідне апаратне забезпечення відповідно до потреб співробітників підприємства, робочі станції різного типу продуктивності, а також файлові сервери, веб – сервери.

При побудові мережі було обрано і реалізовано топологію зірка, як найбільш надійну та оптимально обґрунтовану, було підібране активне обладнання мережі для забезпечення її працездатності та оптимального рівня ефективності.

Визначено вартість запропонованого обладнання, а також підраховано всі витрати на організацію комп’ютерної мережі на даному підприємстві, також було підраховано вартість придбання та експлуатації програмного забезпечення запропонованого для використання на даному підприємстві. В загальному вартість складових комп’ютерної мережі становить:

* Програмне забезпечення – 456 052 грн;
* Апаратне забезпечення – 1 063 362 грн;
* Мережеве обладнання – 24 006;
* Всього – 1 543 420 грн.

Запропоновано методи, що до інженерно – технічного організаційного та криптографічного захисту інформації. Далі в роботі було реалізовано рекомендації що до криптографічного захисту, а саме, розроблено програмний додаток для шифрування та дешифрування даних з використанням алгоритму шифрування XOR. Розробка програмного додатку здійснювалась за допомогою мови програмування С#. Програмний додаток повністю відповідає всім вимогам, поставленим на етапі постановки задачі, та готовий до експлуатації.

Результати роботи апробовані в науковій статті – «Data encryption using XDOR cipher», наукового журналу «Системи управління, навігації та зв’язку», випуск 1(63).

# **CПИCOК ВИКOPИCТAНИХ ДЖEPEЛ**

1. Класифікація комп’ютерних мереж [Електронний ресурс] – Режим доступу. :[https://studopedia.com.ua/1\_48309\_klasifikatsiya– kompyuternih– merezh.html](https://studopedia.com.ua/1_48309_klasifikatsiya-kompyuternih-merezh.html)
2. Призначення мережі [Електронний ресурс] – Режим доступу. :[https://studopedia.com.ua/1\_48310\_priznachennya– merezhi.html](https://studopedia.com.ua/1_48310_priznachennya-merezhi.html)
3. Основні програмні і апаратні компоненти обчислювальної мережі [Електронний ресурс] – Режим доступу. :http://ua.textreferat.com/referat– 7867– 1.html
4. Комп'ютерні мережі: види та хактеристики [Електронний ресурс] – Режим доступу. :https://sites.google.com/site/kompvidhar/2
5. Методичні вказівки [Електронний ресурс] – Режим доступу. :[https://uchika.in.ua/metodichni– vkazivki– ta– individualeni– zavdannya– do– vivchennya– d– v3.html?page=2](https://uchika.in.ua/metodichni-vkazivki-ta-individualeni-zavdannya-do-vivchennya-d-v3.html?page=2)
6. Ді-стар про нас [Електронний ресурс] – Режим доступу. :<https://www.distar.ua/ru/about/>
7. Топологія «Зірка» [Електронний ресурс] – Режим доступу. :https://studfile.net/preview/5263810/page:2/
8. Вимоги до приміщення [Електронний ресурс] – Режим доступу. :[https://studopedia.com.ua/1\_165969\_vimogi– do– primishchen– ta– roztashuvannya– robochih– mists– z– pk.html](https://studopedia.com.ua/1_165969_vimogi-do-primishchen-ta-roztashuvannya-robochih-mists-z-pk.html)
9. Користувачі, аккаунти, групи, дозволи [Електронний ресурс] – Режим доступу. :[https://uk.digitalentertainmentnews.com/user– accounts– groups– permissions– their– role– sharing– 994082](https://uk.digitalentertainmentnews.com/user-accounts-groups-permissions-their-role-sharing-994082)
10. Ідентифікація, авторизація, аутентифікація [Електронний ресурс] – Режим доступу. :https://www.kaspersky.ru/blog/identification– authentication– authorization– difference/29123/
11. Інтернет-магазин Розетка [Електронний ресурс] – Режим доступу. :<https://rozetka.com.ua/ua/>
12. Інтернет- магазин [Електронний ресурс] – Режим доступу. :<https://ekt.com.ua/>
13. Інтернет-магазин серверів [Електронний ресурс] – Режим доступу. :[https://server– shop.ua/](https://server-shop.ua/)
14. Комутатор для мережі [Електронний ресурс] – Режим доступу. :https://www.vostok.dp.ua/ukr/infa1/office/kommutator\_dlya\_seti/
15. Комутатор для мережі [Електронний ресурс] – Режим доступу. :https://hotline.ua/computer– kommutatory/zyxel– gs1900– 24e– eu0101f/?gclid=EAIaIQobChMIy66pqvaH8QIVSOuyCh0X5gS0EAAYAyAAEgIoPPD\_BwE
16. Вита пара [Електронний ресурс] – Режим доступу. :https://secur.ua/ua/articles/ua\_vita– para– scho– ce– za– kabel– i– jakij– krasche– vibrati.html
17. RJ45 [Електронний ресурс] – Режим доступу. :https://artsandculture.google.com/entity/g122qxxcy?hl=uk
18. NetCracker Professional 3.1 portable [Електронний ресурс] – Режим доступу. :https://www.twirpx.com/file/1314986/
19. Криптологія [Електронний ресурс] – Режим доступу. :http://lib.mdpu.org.ua/e– book/kruptologiya/lect1.html
20. Алгоритм шифру [Електронний ресурс] – Режим доступу. :https://programm.top/c– sharp/algorithm/encryption/xor– cipher/
21. Генадій Гулак. Швидкий алгоритм генерації підстановок багатоалфавітної заміни / Володимир Бурячок, Павло Складанний.
22. Як працює xor [Електронний ресурс] – Режим доступу. :[https://www.logsign.com/blog/how– does– xor– cipher– work/](https://www.logsign.com/blog/how-does-xor-cipher-work/)
23. Мова програмування C# [Електронний ресурс] – Режим доступу. :http://progopedia.ru/language/csharp/
24. Чому варто вивчати C# [Електронний ресурс] – Режим доступу. :https://www.quality– assurance– group.com/chomu– varto– vyvchaty– movu– programuvannya– c– c– sharp/
25. Golovko G. V. Information systems use at Poltava national technical Yuri Kondratyuk University / G. V. Golovko, K. M. Nikiforova // Control, navigation and communication systems. – 2018. – vol. 3. – pp. 103– 105. – URL: http://nbuv.gov.ua/UJRN/suntz\_2018\_3\_22
26. Golovko G. V. Data encryption using XOR cipher / G. V. Golovko, A. V. Matiashenko // Control, navigation and communication systems. – 2021. – vol. 1. – pp. 81– 83. – URL: http://journals.nupp.edu.ua/sunz/issue/view/74

# **ДOДAТOК A ПРИКЛАД ЛІСТИНГУ ПРОГРАМНОГО КОДУ**

**Код файлу XORCipher.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CRYPT\_ANN

{

class XORCipher

{ //генератор повторений пароля

private string GetRepeatKey(string s, int n)

{

var r = s;

while (r.Length < n)

{

r += r;

}

return r.Substring(0, n);

}

//метод шифрования/дешифровки

private string Cipher(string text, string secretKey)

{

var currentKey = GetRepeatKey(secretKey, text.Length);

var res = string.Empty;

for (var i = 0; i < text.Length; i++)

{

res += ((char)(text[i] ^ currentKey[i])).ToString();

}

return res;

}

//шифрование текста

public string Encrypt(string plainText, string password)

=> Cipher(plainText, password);

//расшифровка текста

public string Decrypt(string encryptedText, string password)

=> Cipher(encryptedText, password);

}

}

**Код файлу Files.cs**

namespace CRYPT\_ANN

{

partial class Files

{

/// <summary>

/// Required designer variable.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support – do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(Files));

this.button1 = new System.Windows.Forms.Button();

this.button2 = new System.Windows.Forms.Button();

this.saveFileDialog1 = new System.Windows.Forms.SaveFileDialog();

this.openFileDialog1 = new System.Windows.Forms.OpenFileDialog();

this.label1 = new System.Windows.Forms.Label();

this.textBox1 = new System.Windows.Forms.TextBox();

this.comboBox1 = new System.Windows.Forms.ComboBox();

this.textBox2 = new System.Windows.Forms.TextBox();

this.label2 = new System.Windows.Forms.Label();

this.label3 = new System.Windows.Forms.Label();

this.label4 = new System.Windows.Forms.Label();

this.SuspendLayout();

//

// button1

//

this.button1.BackColor = System.Drawing.Color.White;

resources.ApplyResources(this.button1, "button1");

this.button1.ForeColor = System.Drawing.SystemColors.ControlText;

this.button1.Name = "button1";

this.button1.UseVisualStyleBackColor = false;

this.button1.Click += new System.EventHandler(this.button1\_Click);

//

// button2

//

this.button2.BackColor = System.Drawing.Color.White;

resources.ApplyResources(this.button2, "button2");

this.button2.ForeColor = System.Drawing.SystemColors.ControlText;

this.button2.Name = "button2";

this.button2.UseVisualStyleBackColor = false;

this.button2.Click += new System.EventHandler(this.button2\_Click);

//

// openFileDialog1

//

this.openFileDialog1.FileName = "openFileDialog1";

//

// label1

//

resources.ApplyResources(this.label1, "label1");

this.label1.BackColor = System.Drawing.Color.Transparent;

this.label1.ForeColor = System.Drawing.Color.White;

this.label1.Name = "label1";

//

// textBox1

//

resources.ApplyResources(this.textBox1, "textBox1");

this.textBox1.Name = "textBox1";

//

// comboBox1

//

this.comboBox1.FormattingEnabled = true;

this.comboBox1.Items.AddRange(new object[] {

resources.GetString("comboBox1.Items"),

resources.GetString("comboBox1.Items1"),

resources.GetString("comboBox1.Items2"),

resources.GetString("comboBox1.Items3"),

resources.GetString("comboBox1.Items4"),

resources.GetString("comboBox1.Items5"),

resources.GetString("comboBox1.Items6"),

resources.GetString("comboBox1.Items7"),

resources.GetString("comboBox1.Items8"),

resources.GetString("comboBox1.Items9"),

resources.GetString("comboBox1.Items10"),

resources.GetString("comboBox1.Items11")});

resources.ApplyResources(this.comboBox1, "comboBox1");

this.comboBox1.Name = "comboBox1";

//

// textBox2

//

resources.ApplyResources(this.textBox2, "textBox2");

this.textBox2.Name = "textBox2";

//

// label2

//

resources.ApplyResources(this.label2, "label2");

this.label2.BackColor = System.Drawing.Color.Transparent;

this.label2.ForeColor = System.Drawing.Color.White;

this.label2.Name = "label2";

//

// label3

//

resources.ApplyResources(this.label3, "label3");

this.label3.BackColor = System.Drawing.Color.Transparent;

this.label3.ForeColor = System.Drawing.Color.White;

this.label3.Name = "label3";

//

// label4

//

resources.ApplyResources(this.label4, "label4");

this.label4.BackColor = System.Drawing.Color.Transparent;

this.label4.ForeColor = System.Drawing.Color.White;

this.label4.Name = "label4";

//

// Files

//

resources.ApplyResources(this, "$this");

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.Controls.Add(this.label4);

this.Controls.Add(this.label3);

this.Controls.Add(this.label2);

this.Controls.Add(this.textBox2);

this.Controls.Add(this.comboBox1);

this.Controls.Add(this.textBox1);

this.Controls.Add(this.label1);

this.Controls.Add(this.button2);

this.Controls.Add(this.button1);

this.Name = "Files";

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.Button button1;

private System.Windows.Forms.Button button2;

private System.Windows.Forms.SaveFileDialog saveFileDialog1;

private System.Windows.Forms.OpenFileDialog openFileDialog1;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.TextBox textBox1;

private System.Windows.Forms.ComboBox comboBox1;

private System.Windows.Forms.TextBox textBox2;

private System.Windows.Forms.Label label2;

private System.Windows.Forms.Label label3;

private System.Windows.Forms.Label label4;

}

}

**Код файлу Form1.cs**

namespace CRYPT\_ANN

{

partial class Form1

{

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

/// <param name="disposing">истинно, если управляемый ресурс должен быть удален; иначе ложно.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Код, автоматически созданный конструктором форм Windows

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

private void InitializeComponent()

{

System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(Form1));

this.button1 = new System.Windows.Forms.Button();

this.button2 = new System.Windows.Forms.Button();

this.label1 = new System.Windows.Forms.Label();

this.toolStrip1 = new System.Windows.Forms.ToolStrip();

this.toolStripLabel2 = new System.Windows.Forms.ToolStripLabel();

this.toolStripLabel1 = new System.Windows.Forms.ToolStripLabel();

this.toolStrip1.SuspendLayout();

this.SuspendLayout();

//

// button1

//

resources.ApplyResources(this.button1, "button1");

this.button1.Name = "button1";

this.button1.UseVisualStyleBackColor = true;

this.button1.Click += new System.EventHandler(this.button1\_Click);

this.button1.MouseLeave += new System.EventHandler(this.button1\_MouseLeave);

this.button1.MouseHover += new System.EventHandler(this.button1\_MouseHover);

//

// button2

//

resources.ApplyResources(this.button2, "button2");

this.button2.Name = "button2";

this.button2.UseVisualStyleBackColor = true;

this.button2.Click += new System.EventHandler(this.button2\_Click);

this.button2.MouseLeave += new System.EventHandler(this.button2\_MouseLeave);

this.button2.MouseHover += new System.EventHandler(this.button2\_MouseHover);

//

// label1

//

resources.ApplyResources(this.label1, "label1");

this.label1.BackColor = System.Drawing.Color.Transparent;

this.label1.ForeColor = System.Drawing.Color.White;

this.label1.Name = "label1";

//

// toolStrip1

this.toolStrip1.BackColor = System.Drawing.Color.Black;

this.toolStrip1.ImageScalingSize = new System.Drawing.Size(20, 20);

this.toolStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.toolStripLabel2,

this.toolStripLabel1});

this.toolStrip1.LayoutStyle = System.Windows.Forms.ToolStripLayoutStyle.Flow;

resources.ApplyResources(this.toolStrip1, "toolStrip1");

this.toolStrip1.Name = "toolStrip1";

//

// toolStripLabel2

//

this.toolStripLabel2.ForeColor = System.Drawing.Color.White;

this.toolStripLabel2.Name = "toolStripLabel2";

resources.ApplyResources(this.toolStripLabel2, "toolStripLabel2");

//

// toolStripLabel1

//

this.toolStripLabel1.ForeColor = System.Drawing.Color.White;

this.toolStripLabel1.LinkColor = System.Drawing.Color.Green;

this.toolStripLabel1.Name = "toolStripLabel1";

resources.ApplyResources(this.toolStripLabel1, "toolStripLabel1");

//

// Form1

//

resources.ApplyResources(this, "$this");

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.BackColor = System.Drawing.SystemColors.ButtonHighlight;

this.Controls.Add(this.toolStrip1);

this.Controls.Add(this.label1);

this.Controls.Add(this.button2);

this.Controls.Add(this.button1);

this.HelpButton = true;

this.Name = "Form1";

this.Load += new System.EventHandler(this.Form1\_Load);

this.toolStrip1.ResumeLayout(false);

this.toolStrip1.PerformLayout();

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.Button button1;

private System.Windows.Forms.Button button2;

private System.Windows.Forms.Label label1;

public System.Windows.Forms.ToolStrip toolStrip1;

public System.Windows.Forms.ToolStripLabel toolStripLabel1;

private System.Windows.Forms.ToolStripLabel toolStripLabel2;

}

}

**Код файлу Login.cs**

namespace CRYPT\_ANN

{

partial class Login

{

/// <summary>

/// Required designer variable.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support – do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(Login));

this.textBox1 = new System.Windows.Forms.TextBox();

this.textBox2 = new System.Windows.Forms.TextBox();

this.label1 = new System.Windows.Forms.Label();

this.button1 = new System.Windows.Forms.Button();

this.label2 = new System.Windows.Forms.Label();

this.toolStrip1 = new System.Windows.Forms.ToolStrip();

this.toolStripButton1 = new System.Windows.Forms.ToolStripSplitButton();

this.englishToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.русскийToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.українськаToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();

this.toolStripLabel1 = new System.Windows.Forms.ToolStripLabel();

this.button2 = new System.Windows.Forms.Button();

this.toolStrip1.SuspendLayout();

this.SuspendLayout();

//

// textBox1

//

this.textBox1.BackColor = System.Drawing.Color.Cornsilk;

this.textBox1.ForeColor = System.Drawing.Color.Black;

resources.ApplyResources(this.textBox1, "textBox1");

this.textBox1.Name = "textBox1";

//

// textBox2

//

this.textBox2.BackColor = System.Drawing.Color.Cornsilk;

this.textBox2.ForeColor = System.Drawing.Color.Black;

resources.ApplyResources(this.textBox2, "textBox2");

this.textBox2.Name = "textBox2";

//

// label1

//

resources.ApplyResources(this.label1, "label1");

this.label1.BackColor = System.Drawing.Color.Transparent;

this.label1.FlatStyle = System.Windows.Forms.FlatStyle.Popup;

this.label1.ForeColor = System.Drawing.Color.White;

this.label1.Name = "label1";

//

// button1

//

this.button1.BackColor = System.Drawing.Color.Black;

resources.ApplyResources(this.button1, "button1");

this.button1.ForeColor = System.Drawing.Color.White;

this.button1.Name = "button1";

this.button1.UseVisualStyleBackColor = false;

this.button1.Click += new System.EventHandler(this.button1\_Click);

//

// label2

//

resources.ApplyResources(this.label2, "label2");

this.label2.BackColor = System.Drawing.Color.Transparent;

this.label2.FlatStyle = System.Windows.Forms.FlatStyle.Popup;

this.label2.ForeColor = System.Drawing.Color.White;

this.label2.Name = "label2";

//

// toolStrip1

//

this.toolStrip1.BackColor = System.Drawing.SystemColors.ActiveCaptionText;

this.toolStrip1.ImageScalingSize = new System.Drawing.Size(20, 20);

this.toolStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.toolStripButton1,

this.toolStripLabel1});

this.toolStrip1.LayoutStyle = System.Windows.Forms.ToolStripLayoutStyle.Flow;

resources.ApplyResources(this.toolStrip1, "toolStrip1");

this.toolStrip1.Name = "toolStrip1";

//

// toolStripButton1

//

this.toolStripButton1.BackColor = System.Drawing.Color.Black;

this.toolStripButton1.DisplayStyle = System.Windows.Forms.ToolStripItemDisplayStyle.Text;

this.toolStripButton1.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {

this.englishToolStripMenuItem,

this.русскийToolStripMenuItem,

this.українськаToolStripMenuItem});

this.toolStripButton1.ForeColor = System.Drawing.Color.White;

resources.ApplyResources(this.toolStripButton1, "toolStripButton1");

this.toolStripButton1.Name = "toolStripButton1";

//

// englishToolStripMenuItem

//

this.englishToolStripMenuItem.BackColor = System.Drawing.SystemColors.ActiveCaptionText;

this.englishToolStripMenuItem.ForeColor = System.Drawing.Color.White;

this.englishToolStripMenuItem.Name = "englishToolStripMenuItem";

resources.ApplyResources(this.englishToolStripMenuItem, "englishToolStripMenuItem");

this.englishToolStripMenuItem.Click += new System.EventHandler(this.englishToolStripMenuItem\_Click);

//

// русскийToolStripMenuItem

//

this.русскийToolStripMenuItem.BackColor = System.Drawing.SystemColors.ActiveCaptionText;

this.русскийToolStripMenuItem.ForeColor = System.Drawing.Color.White;

this.русскийToolStripMenuItem.Name = "русскийToolStripMenuItem";

resources.ApplyResources(this.русскийToolStripMenuItem, "русскийToolStripMenuItem");

this.русскийToolStripMenuItem.Click += new System.EventHandler(this.русскийToolStripMenuItem\_Click);

//

// українськаToolStripMenuItem

//

this.українськаToolStripMenuItem.BackColor = System.Drawing.SystemColors.ActiveCaptionText;

this.українськаToolStripMenuItem.ForeColor = System.Drawing.Color.White;

this.українськаToolStripMenuItem.Name = "українськаToolStripMenuItem";

resources.ApplyResources(this.українськаToolStripMenuItem, "українськаToolStripMenuItem");

this.українськаToolStripMenuItem.Click += new System.EventHandler(this.українськаToolStripMenuItem\_Click);

//

// toolStripLabel1

//

this.toolStripLabel1.ForeColor = System.Drawing.Color.Red;

this.toolStripLabel1.LinkColor = System.Drawing.Color.DarkGreen;

this.toolStripLabel1.Name = "toolStripLabel1";

resources.ApplyResources(this.toolStripLabel1, "toolStripLabel1");

//

// button2

//

this.button2.BackColor = System.Drawing.Color.DarkRed;

resources.ApplyResources(this.button2, "button2");

this.button2.ForeColor = System.Drawing.Color.White;

this.button2.Name = "button2";

this.button2.UseVisualStyleBackColor = false;

this.button2.Click += new System.EventHandler(this.button2\_Click);

//

// Login

//

resources.ApplyResources(this, "$this");

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.Controls.Add(this.button2);

this.Controls.Add(this.toolStrip1);

this.Controls.Add(this.label2);

this.Controls.Add(this.button1);

this.Controls.Add(this.label1);

this.Controls.Add(this.textBox2);

this.Controls.Add(this.textBox1);

this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedToolWindow;

this.Name = "Login";

this.Load += new System.EventHandler(this.Login\_Load);

this.toolStrip1.ResumeLayout(false);

this.toolStrip1.PerformLayout();

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.TextBox textBox2;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.Button button1;

private System.Windows.Forms.Label label2;

public System.Windows.Forms.TextBox textBox1;

private System.Windows.Forms.ToolStrip toolStrip1;

private System.Windows.Forms.ToolStripSplitButton toolStripButton1;

private System.Windows.Forms.ToolStripMenuItem englishToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem русскийToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem українськаToolStripMenuItem;

private System.Windows.Forms.ToolStripLabel toolStripLabel1;

private System.Windows.Forms.Button button2;

}

}

**Код файлу Message.cs**

namespace CRYPT\_ANN

{

partial class Message

{

/// <summary>

/// Required designer variable.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support – do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(Message));

this.richTextBox1 = new System.Windows.Forms.RichTextBox();

this.button1 = new System.Windows.Forms.Button();

this.button2 = new System.Windows.Forms.Button();

this.textBox1 = new System.Windows.Forms.TextBox();

this.label1 = new System.Windows.Forms.Label();

this.saveFileDialog1 = new System.Windows.Forms.SaveFileDialog();

this.comboBox1 = new System.Windows.Forms.ComboBox();

this.label2 = new System.Windows.Forms.Label();

this.textBox2 = new System.Windows.Forms.TextBox();

this.label3 = new System.Windows.Forms.Label();

this.button3 = new System.Windows.Forms.Button();

this.SuspendLayout();

//

// richTextBox1

//

resources.ApplyResources(this.richTextBox1, "richTextBox1");

this.richTextBox1.Name = "richTextBox1";

this.richTextBox1.MouseClick += new System.Windows.Forms.MouseEventHandler(this.richTextBox1\_MouseClick);

//

// button1

//

this.button1.BackColor = System.Drawing.Color.White;

resources.ApplyResources(this.button1, "button1");

this.button1.ForeColor = System.Drawing.SystemColors.ControlText;

this.button1.Name = "button1";

this.button1.UseVisualStyleBackColor = false;

this.button1.Click += new System.EventHandler(this.button1\_Click);

//

// button2

//

this.button2.BackColor = System.Drawing.Color.White;

resources.ApplyResources(this.button2, "button2");

this.button2.ForeColor = System.Drawing.SystemColors.ControlText;

this.button2.Name = "button2";

this.button2.UseVisualStyleBackColor = false;

this.button2.Click += new System.EventHandler(this.button2\_Click);

//

// textBox1

//

resources.ApplyResources(this.textBox1, "textBox1");

this.textBox1.Name = "textBox1";

//

// label1

//

resources.ApplyResources(this.label1, "label1");

this.label1.BackColor = System.Drawing.Color.Transparent;

this.label1.ForeColor = System.Drawing.Color.White;

this.label1.Name = "label1";

//

// comboBox1

//

this.comboBox1.FormattingEnabled = true;

this.comboBox1.Items.AddRange(new object[] {

resources.GetString("comboBox1.Items"),

resources.GetString("comboBox1.Items1"),

resources.GetString("comboBox1.Items2"),

resources.GetString("comboBox1.Items3"),

resources.GetString("comboBox1.Items4"),

resources.GetString("comboBox1.Items5"),

resources.GetString("comboBox1.Items6"),

resources.GetString("comboBox1.Items7"),

resources.GetString("comboBox1.Items8"),

resources.GetString("comboBox1.Items9"),

resources.GetString("comboBox1.Items10"),

resources.GetString("comboBox1.Items11")});

resources.ApplyResources(this.comboBox1, "comboBox1");

this.comboBox1.Name = "comboBox1";

this.comboBox1.SelectedIndexChanged += new System.EventHandler(this.comboBox1\_SelectedIndexChanged);

//

// label2

//

resources.ApplyResources(this.label2, "label2");

this.label2.BackColor = System.Drawing.Color.Transparent;

this.label2.ForeColor = System.Drawing.Color.White;

this.label2.Name = "label2";

//

// textBox2

//

resources.ApplyResources(this.textBox2, "textBox2");

this.textBox2.Name = "textBox2";

//

// label3

//

resources.ApplyResources(this.label3, "label3");

this.label3.BackColor = System.Drawing.Color.Transparent;

this.label3.ForeColor = System.Drawing.Color.White;

this.label3.Name = "label3";

//

// button3

//

this.button3.BackColor = System.Drawing.Color.White;

resources.ApplyResources(this.button3, "button3");

this.button3.ForeColor = System.Drawing.SystemColors.ControlText;

this.button3.Name = "button3";

this.button3.UseVisualStyleBackColor = false;

this.button3.Click += new System.EventHandler(this.button3\_Click);

//

// Message

//

resources.ApplyResources(this, "$this");

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.Controls.Add(this.label3);

this.Controls.Add(this.textBox2);

this.Controls.Add(this.label2);

this.Controls.Add(this.comboBox1);

this.Controls.Add(this.label1);

this.Controls.Add(this.textBox1);

this.Controls.Add(this.button3);

this.Controls.Add(this.button2);

this.Controls.Add(this.button1);

this.Controls.Add(this.richTextBox1);

this.Name = "Message";

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.RichTextBox richTextBox1;

private System.Windows.Forms.Button button1;

private System.Windows.Forms.Button button2;

private System.Windows.Forms.TextBox textBox1;

private System.Windows.Forms.Label label1;

private System.Windows.Forms.SaveFileDialog saveFileDialog1;

private System.Windows.Forms.ComboBox comboBox1;

private System.Windows.Forms.Label label2;

private System.Windows.Forms.TextBox textBox2;

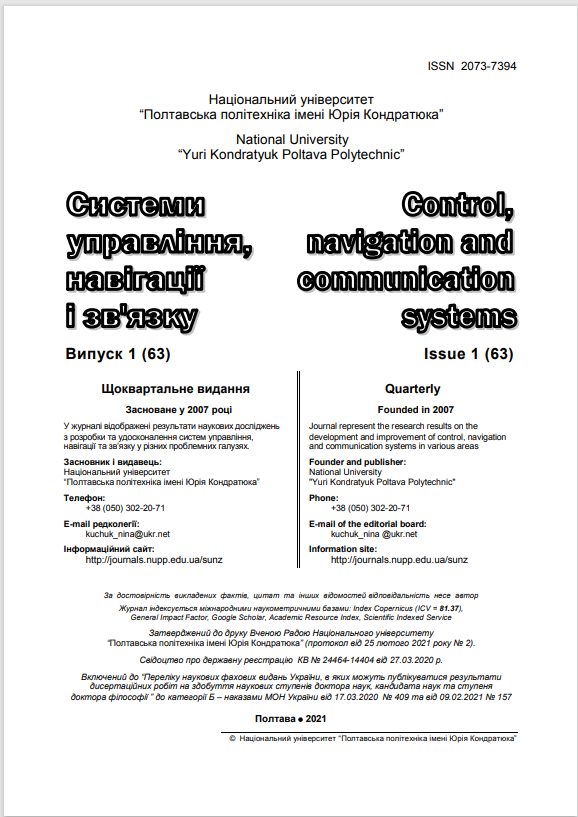
private System.Windows.Forms.Label label3;

private System.Windows.Forms.Button button3;

}

}

# **ДOДAТOК В НАУКОВИЙ ЖУРНАЛ**



# **ДOДAТOК С ЗМІСТ НАУКОВОГО ЖУРНАЛУ**



# **ДOДAТOК D ФРАГМЕНТ СТАТТІ**

