

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: «РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СТВОРЕННЯ ТА
РЕДАГУВАННЯ МАКРОСІВ МОВОЮ C#»

Виконав: студент 5 курсу, групи ППЗ-52
спеціальності
121 Інженерії програмного забезпечення
(шифр і назва спеціальності)

_____ Княжук В. А.
(прізвище та ініціали)
Керівник _____ Гаманюк І. М.
(прізвище та ініціали)
Рецензент _____
(прізвище та ініціали)
Нормоконтроль _____
(прізвище та ініціали)

КИЇВ – 2021

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення _____

Ступінь вищої освіти - «Бакалавр» _____

Спеціальність - 121 «Інженерія програмного забезпечення» _____

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного
забезпечення

_____ О.В. Негоденко

“ _____ ” _____ 20 _____ року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Княжуку Владиславу Анатолійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: «РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СТВОРЕННЯ ТА РЕДАГУВАННЯ МАКРОСІВ МОВОЮ C#»

Керівник роботи с.в.к. Гаманюк

І. М.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “12” березня 2021 року №65.

2. Строк подання студентом роботи 01 “червня” 2021 р.

3. Вихідні дані до роботи:

- 3.1. Положення побудови макросів;

- 3.2. Методи побудови макросів;

- 3.3. Розробка моделі макросів;

- 3.4. Науково-технічна література.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

- 4.1. Загальні положення побудови макросів; _____

- 4.2. Аналіз технології і методів побудови макросів; _____

- 4.3. Побудова музичної рекомендаційної системи; _____

- 4.4. Висновки _____

5. Перелік графічного матеріалу.

- 5.1. Основні характеристики роботи;

- 5.2. Актуальність задачі;

- 5.3. Недоліки існуючих систем;

- 5.4. Реалізація алгоритмів для макросів;

- 5.5. Порівняння якості отриманих результатів;

- 5.6. Порівняння швидкості користування комп'ютером використовуючи макроси;

5.7. Архітектура рекомендаційної системи;

5.8. Висновки.

6. Дата видачі завдання 19.04.2021р.

КАЛЕНДАРНИЙ ПЛАН

№ з / п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	19.04.2021	Виконано
2	Дослідження положення побудови макросів	26.04.2021	Виконано
3	Аналіз методів побудови макросів	03.05.2021	Виконано
4	Розробка моделі макросів	10.05.2021	Виконано
5	Висновки, оформлення роботи	17.05.2021	Виконано
6	Розробка демонстраційних матеріалів	22.05.2021	Виконано
7	Здача роботи	01.06.2021	Виконано

Студент _____ Княжук В. А. _____
(підпис) (прізвище та ініціали)

Керівник роботи _____ Гаманюк І. М. _____
підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина магістерської роботи 59 стр., 22 рис., 6 табл., 20 джерел

СИСТЕМА РОБОТИ З МАКРОСАМИ, МАКРОКОМАНДА, АЛГОРИТМ,
СИСТЕМА, МЕТОД.

Об'єктом дослідження - є система для створення та написання макросів.

Мета роботи – є поліпшення та прискорення роботи користувачів під час роботи за персональним комп'ютером.

Предмет дослідження – є низька швидкість та ефективність використання персонального комп'ютера.

Методи дослідження. У науковій роботі було використані загальнонаукові та спеціальні методи дослідження, а саме: методи спостереження, порівняння для наочного для того щоб наочно бачити різницю між користувачами, методи розрахунку для отримання різниці в швидкості користування персональним комп'ютером, методи експерименту для перевірки істинності гіпотез, методи аналізу та узагальнення.

Зміст

Вступ	8
1 Загальна частина	9
1.1 Постановка задачі	9
1.2 Дослідження і аналіз об'єкту програмування	9
1.3 Використані програмні засоби	12
1.4 Вимоги до апаратного та програмного забезпечення	16
2 Практична частина	16
2.1 Створення та налагодження програми	16
2.2 Опис програми та її алгоритмів	18
2.3 Інструкція програміста	23
2.4 Інструкція оператора	23
3 Організаційно-економічна частина	30
3.1 Оцінка вартості програмного продукту	30
3.2 Розрахунок собівартості програмного продукту	38
3.3 Розрахунок ціни програмного продукту	40
3.4 Зведена таблиця показників	41
4 Безпека життєдіяльності	42
4.1 Вимоги до приміщень і організації робочих місць	42
4.2 Вимоги охорони праці при експлуатації комп'ютерних систем	51
Висновки	61
Список використаних джерел	62
Додаток А – Код програми	63

Вступ

Макрос — це набір вказівок та інструкцій, які виконуються як одна вказівка. Якщо створення чи оформлення документа передбачає повторення однакових дій з його об'єктами, їх виконання можна автоматизувати за допомогою макроса.

Призначення макрокоманд полягає або в автоматизації часто вживаних послідовностей дій чи команд, або у сильнішому абстрагуванні дій/команд.

Можна автоматизувати майже будь-яку рутинну або повторювану роботу. Можна виконати цю автоматизовану роботу миттєво, просто натиснувши клавішу на клавіатурі.

Макрос - це набір інструкцій, які повідомляють програмі або системі які дії слід виконати, щоб досягти певної мети.

Різниця, однак, полягає в тому, що макрос об'єднує всі ці інструкції в одному сценарії, який потім можна викликати за допомогою комбінації клавіш. З цієї точки зору макрос відрізняється від, скажімо, рецепта приготування хліба, але схожий на автоматичну хлібопекарню, завантаживши інгредієнти в яку, можна спекти хліб одним натисканням кнопки.

Серед найбільш відомих програмних продуктів такого виду є Hotkeycontrol, Key Manager, BotMek, AutoHotkey, Hot Keyboard тощо.

Вищеописані програмні продукти мають вузьке призначення, а об'єднання усіх функцій у одну програму є актуальною та корисною задачею.

В межах дипломної роботи необхідно розробити програмне забезпечення, задача якого полягає в тому, щоб об'єднати більшість корисних функцій взаємодії користувача з ядром операційної системи в одній програмі.

1 ЗАГАЛЬНА ЧАСТИНА

1.1 Постановка задачі

Основне завдання даного проекту полягає в створенні модуля взаємодії з користувачем системного програмного забезпечення для написання макросів. В програмі повинні бути передбачені такі основні функціональні можливості:

1. Завершення роботи процесів.
2. Перезапуск Windows.
3. Завершення роботи Windows.
4. Відкриття папок.
5. Відкриття файлів.
6. Відкриття сайтів.
7. Встановлення позиції миші.
8. Натискання клавіш миші та клавіатури.
9. Можливість встановлення затримки між діями.

Для виконання будь-якої дії потрібно написати відповідну команду та задати їй довільний ідентифікатор.

Модуль керування повинен використовувати клавіатуру.

Вихідні дані: Виконання команди макросу.

Вхідні дані: Натиснуті клавіші.

Користувачі: діти від 12-ти років.

1.2 Дослідження і аналіз об'єкту програмування

Програмне забезпечення (програ́мні засоби) – сукупність програм системи обробки інформації і програмних документів, необхідних для експлуатації цих програм.

Розрізняють системне програмне забезпечення (зокрема, операційна система, транслятори, редактори, графічний інтерфейс користувача); прикладне програмне забезпечення, що використовується для виконання конкретних завдань, наприклад,

статистичне програмне забезпечення; інструментальне програмне забезпечення (комп'ютерні програми, призначені для проектування, розробки, адміністрування і супроводження системного та прикладного програмного забезпечення).

Виконання програмного забезпечення комп'ютером полягає у маніпулюванні інформацією та керуванні апаратними компонентами комп'ютера. Наприклад, типовим для персональних комп'ютерів є відтворення інформації на екран та отримання її з клавіатури.

Програмне забезпечення (software) та апаратне забезпечення (hardware) — це два комплементарні компоненти комп'ютера, причому межа між ними нечітка: деякі фрагменти програмного забезпечення на практиці реалізуються суто апаратурою мікросхем комп'ютера, а програмне забезпечення, в свою чергу, здатне виконувати (емулювати) функції електронної апаратури. По суті, призначення програмного забезпечення полягає в керуванні як самим комп'ютером так і іншими програмами та маніпулюванні інформацією.

Комплекс програм, які забезпечують управління компонентами комп'ютерної системи, такими як процесор, оперативна пам'ять, пристрої введення-виведення, мережеве обладнання, виступаючи як «міжшаровий інтерфейс», з одного боку якого — апаратура, а з іншого — додатки користувача. На відміну від прикладного програмного забезпечення, системне не вирішує конкретні практичні завдання, а лише забезпечує роботу інших програм, надаючи їм сервісні функції, абстрагуючи деталі апаратної і мікропрограмної реалізації обчислювальної системи, керує апаратними ресурсами обчислювальної системи. Віднесення того чи іншого програмного забезпечення до системного є умовним, і залежить від угод, використовуваних у конкретному контексті. Як правило, до системного програмного забезпечення відносяться операційні системи, широкий клас сполучного програмного забезпечення.

Розглянемо докладніше аналоги програм для написання макросів.

Hotkeycontrol.

Розробник: Inchwes. Назва програми Hotkeycontrol вже говорить сама за себе - вона дозволить здійснювати настройку гарячих клавіш. Тут ви можете створити

необмежену кількість комбінацій і послідовностей будь-яких складнощів, звернувшись в окремий модуль софту під назвою «Масго». Саме тут відбувається автоматизація процесів шляхом додавання відповідних налаштувань. Якщо ж сама команда не є складною, її можна записати в якості гарячої клавіші. Комбінації створюються в інших модулях, відповідних напряму команд. Наприклад, управління яскравістю і гучністю проводиться в одній вкладці, а системні настройки та взаємодія з додатками - в інших. Розбереться зі складанням профілю навіть самий досвідчений користувач, оскільки інтерфейс реалізований максимально просто, проте відсутність російської мови відштовхує деяких користувачів.

Key Manager.

Розробник: ATNSOFT. Key Manager - це більш функціональний варіант програми Key Remapper, яка, крім перепризначення клавіш і кнопок миші, містить безліч інших корисних функцій.

Наприклад, Key Manager дозволяє перепризначувати комбінації клавіш і кнопок миші, послідовності натискань, довге натискання (натискання та утримання). Дозволяє представляти дії у вигляді спливаючих меню, які можна відкривати по гарячих клавішах; виконувати, записувати і редагувати макроси натискань (послідовності натискань); удруковувати або вставляти текст (в тому числі форматований), миттєво заповнювати форми сайтів; запускати програми і відкривати сайти (в параметрах адреси яких можна вказати поточний виділений текст); змінювати розкладку і регістр виділеного тексту; керувати живленням монітора, комп'ютера; регулювати гучність і встановлювати певний рівень; виконувати дії над вікнами (переміщення, установка конкретного розміру).

Макроси часто використовуються для таких цілей:

- прискорення процесу редагування або форматування;
- об'єднання кількох вказівок, наприклад, для вставлення таблиці з вказаними розмірами і межами та певною кількістю рядків і стовпців;
- спрощення доступу до параметрів у діалогових вікнах;
- автоматизація процесу опрацювання складних- послідовних дій у завданнях.

Макроси зберігаються у шаблонах документів разом із стилями, комбінаціями клавіш, користувацькими панелями інструментів та меню тощо.

1.3 Використані програмні засоби

В якості операційної системи, для якої була реалізована програма, обрана система з серії Microsoft Windows. Головною причиною є те, що сім'я систем Microsoft на сьогоднішній день є найпоширенішою, і в той же час багатофункціональною системою.

Для написання програми була використана мова C#.

C# (вимовляється *Ci-шарп*) — об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Розроблена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під егідою Microsoft Research (при фірмі Microsoft).

Синтаксис C# близький до C++ і Java. Мова 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 C++, Delphi, Модула і Smalltalk — C#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад множинне спадкування класів (на відміну від C++).

C# є дуже близьким родичем мови програмування Java. Мова Java була створена компанією Sun Microsystems, коли глобальний розвиток інтернету поставив задачу розсосереджених обчислень. Взявши за основу популярну мову C++, Java виключила з неї потенційно небезпечні речі (типу вказівників без контролю виходу за межі). Для розсосереджених обчислень була створена концепція віртуальної машини та машинно-незалежного байт-коду, свого роду посередника між вихідним текстом програм і апаратними інструкціями комп'ютера чи іншого інтелектуального пристрою.

Java набула чималої популярності, і була ліцензована також і компанією Microsoft. Але з плином часу Sun почала винуватити Microsoft, що та при створенні свого клону Java робить її сумісною виключно з платформою Windows, чим суперечить самій концепції машинно-незалежного середовища виконання і порушує ліцензійну угоду. Microsoft відмовилася піти назустріч вимогам Sun, і тому з'ясування стосунків набуло статусу судового процесу. Суд визнав позицію Sun справедливою, і зобов'язав Microsoft відмовитися від позаліцензійного використання Java.

У цій ситуації в Microsoft вирішили, користуючись своєю вагою на ринку, створити свій власний аналог Java, мови, в якій корпорація стане повновладним господарем. Ця новостворена мова отримала назву C#. Вона успадкувала від Java концепції віртуальної машини (середовище .NET), байт-коду (MSIL) і більшої безпеки вихідного коду програм, плюс врахувала досвід використання програм на Java.

Нововведенням C# стала можливість легшої взаємодії, порівняно з мовами-попередниками, з кодом програм, написаних на інших мовах, що є важливим при створенні великих проектів. Якщо програми на різних мовах виконуються на платформі .NET, .NET бере на себе клопіт щодо сумісності програм (тобто типів даних, за кінцевим рахунком).

Станом на сьогодні C# визначено флагманською мовою корпорації Microsoft, бо вона найповніше використовує нові можливості .NET. Решта мов програмування, хоч і підтримуються, але визнані такими, що мають спадкові прогалини щодо використання .NET.

Microsoft Visual Studio - лінійка продуктів компанії Microsoft, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів. Дані продукти дозволяють розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, підтримуваних Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework і Silverlight.

Visual Studio включає в себе редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторінга коду. Вбудований відладчик може працювати як відладчик рівня вихідного коду, так і відладчик машинного рівня. Решта вбудовуються інструменти включають в себе редактор форм для спрощення створення графічного інтерфейсу додатку, веб-редактор, дизайнер класів і дизайнер схеми бази даних. Visual Studio дозволяє створювати і підключати сторонні додатки (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду (як, наприклад, Subversion і Visual SourceSafe), додавання нових наборів інструментів (наприклад, для редагування і візуального проектування коду на предметно-орієнтованих мовах програмування) або інструментів для інших аспектів процесу розробки програмного забезпечення (наприклад, клієнт Team Explorer для роботи з Team Foundation Server).

Компоненти:

Visual Studio включає один або кілька компонентів з наступних:

Visual Basic .NET, а до його появи - Visual Basic

Visual C ++

Visual C # (включений починаючи з Visual Studio .NET)

Visual F # (включений починаючи з Visual Studio 2010)

Багато варіантів поставки також включають:

Microsoft SQL Server або Microsoft SQL Server Express

У минулому до складу Visual Studio також входили продукти:

Visual InterDev

Visual J ++

Visual J #

Visual FoxPro

Visual Source Safe - файл-серверна система управління версіями

Бібліотеки:

1) System - простір імен System містить фундаментальні і базові класи, що визначають часто використовувані типи значень і посилальних даних, події і обробники подій, інтерфейси, атрибути і виключення обробки.

2) System.Drawing - простір імен System.Drawing забезпечує доступ до базових функціональними можливостями графічного інтерфейсу GDI +. Простору імен System.Drawing.Drawing2D, System.Drawing.Imaging і System.Drawing.Text забезпечують додаткові функціональні можливості.

3) System.Text - простір імен System.Text містить класи, які представляють кодування ASCII і Юнікода; абстрактні базові класи для перетворення блоків знаків в блоки байтів і назад; допоміжний клас, який обробляє і форматує об'єкти String, не створюючи проміжні екземпляри String.

4) System.Windows.Forms - простір імен містить класи для створення додатків Windows, що користуються перевагами повного призначеного для користувача інтерфейсу, що надаються в операційній системі Microsoft Windows.

5) System.Diagnostics - простір імен надає класи, що дозволяють взаємодіяти з системними процесами, журналами подій і лічильники продуктивності.

6) System.IO - простір імен System.IO містить типи, що дозволяють здійснювати читання і запис у файли і потоки даних, а також типи для базової підтримки файлів і папок.

7) System.Runtime.InteropServices - простір імен надає широкий спектр членів, що підтримують COM-взаємодія і служби виклику платформи. Якщо ви не знайомі з цими службами, див. Взаємодію з некерованим кодом.

1.4 Вимоги до апаратного та програмного забезпечення

Вимоги до апаратного забезпечення:

- Оперативна пам'ять: 512 Мб.
- Процесор: Intel Pentium x86 і вище, частота від 300 MHz.
- Touchscreen або миша
- Відеокарта: інтегрована.

Вимоги до програмного забезпечення:

- Операційна система: Windows XP, 7, 8, 10
- .Net Framework 4.5.1

2 ПРАКТИЧНА ЧАСТИНА

2.1 Створення та налагодження програми

Системне програмне забезпечення для написання макросів складається з набору підпрограм, які створені за стандартом Windows Form Application у середовищі програмування Microsoft Visual Studio 2019 мовою програмування C#.

Розроблений програмний комплекс використовує такий набір бібліотек:

1. System.IO

Простір імен System.IO містить типи, що дозволяють здійснювати читання і запис у файли і потоки даних, а також типи для базової підтримки файлів і папок.

Здебільшого використовується для читання потокових даних, для роботи з текстовими документами та для створення файлів.

2. System.Diagnostics

Простір імен System.Diagnostics містить типи, що дозволяють здійснювати взаємодію з системними процесами, журналами подій і лічильниками продуктивності. Дочірні простори імен містять типи для взаємодії із засобами аналізу коду, підтримки контрактів, розширення підтримки відстеження і інструментування додатків під час розробки, ведення журналів подій за допомогою засобу відстеження подій для Windows (ETW), читання і записи в журнали подій і

збору даних про продуктивність , а також для читання і запису інформації про символи налагодження.

Використовується для відкриття ссилки, файлів, папок, завершення та перезапуску роботи комп'ютера, для переведення комп'ютеру у режим сну та гібернації.

3. System.Text

Містить класи, які представляють кодування ASCII і Юнікода; абстрактні базові класи для перетворення блоків знаків в блоки байтів і назад; допоміжний клас, який обробляє і форматує об'єкти String, не створюючи проміжні екземпляри String. Класи кодування в основному призначені для перетворення між різними кодуваннями або кодовими сторінками і кодуванням Юнікод. Encoding.Unicode (UTF-16) використовується для внутрішніх цілей .NET, а Encoding.UTF8 кодування часто використовується для зберігання символьних даних, щоб забезпечити переносимість між комп'ютерами і культурами.

4. System.Runtime.InteropServices

Надає різноманітні члени, які підтримують COM-взаємодію і служби виклику платформи. Члени цього простору імен надають кілька категорій функціональних можливостей. Атрибути керують поведінкою маршала, наприклад, як упорядковувати структури або представляти рядка.

2.2 Опис програми та її алгоритмів

Для більшого розуміння роботи системи необхідно розглянути функції системи докладніше.

Діаграма послідовності на рисунку 2.1 показує як працює система роботи з макросами.

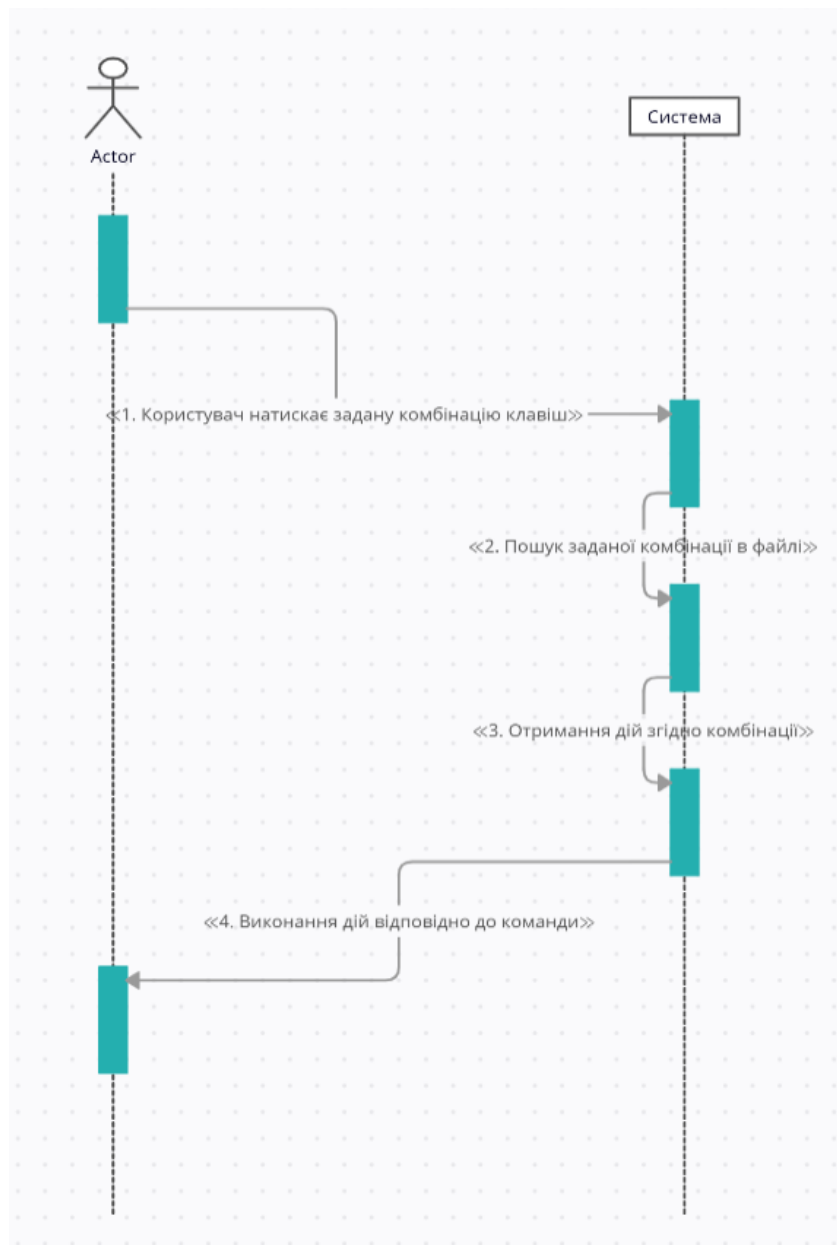


Рис. 2.1 – Діаграма послідовності роботи макросів.

На рисунку 2.2 показана UML-Діаграма класів

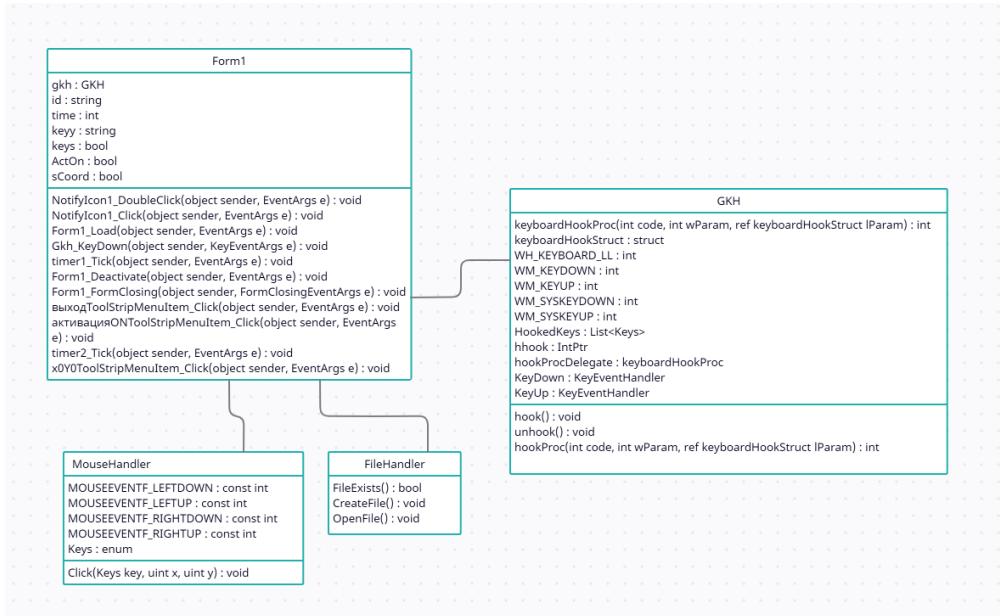


Рис. 2.2 – UML-Діаграма класів

На рисунку 2.3 показана UML-Діаграма, на якій зображені можливості користувача.

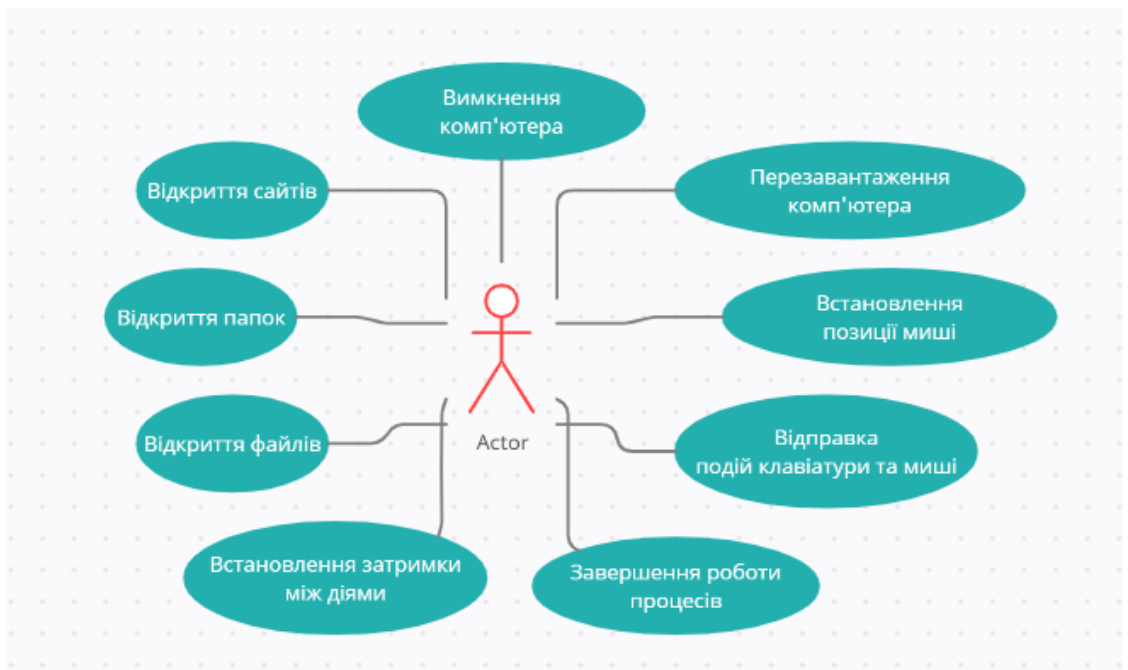


Рис. 2.3 – UML-Діаграма.

Як видно на рисунку 2.3 користувач може:

1. Відкривати папки, сайти та файли.
2. Вимикати та перезавантажувати комп'ютер.
3. Встановлювати позицію миші на екрані.
4. Відправляти події клавіатури та миші.
5. Встановлювати затримку між командами макросу.
6. Завершувати роботу процесів системи.

Для прикладу були створені блок-схеми для деяких команд які представлені на рисунках 2.4, 2.5, 2.6.

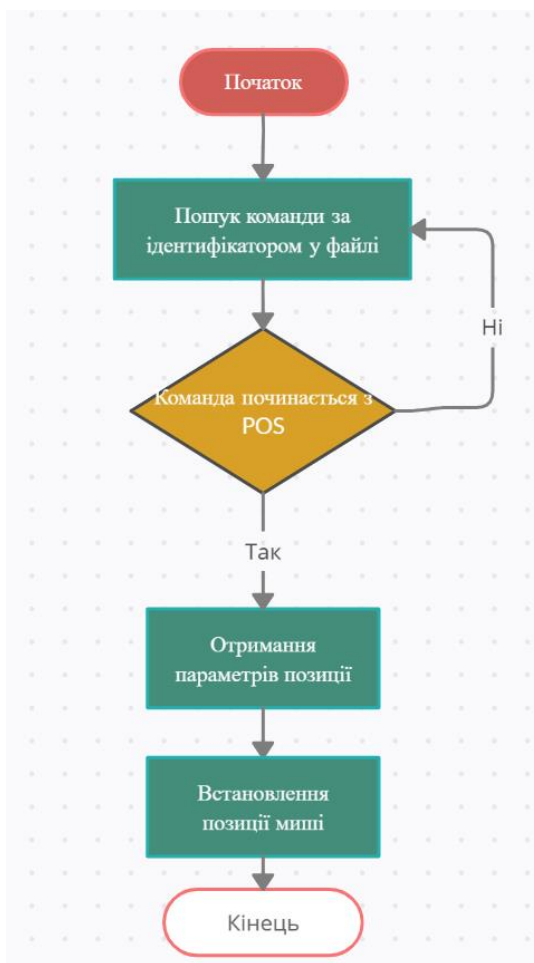


Рис. 2.4 – Діаграма алгоритму для встановлення позиції миші

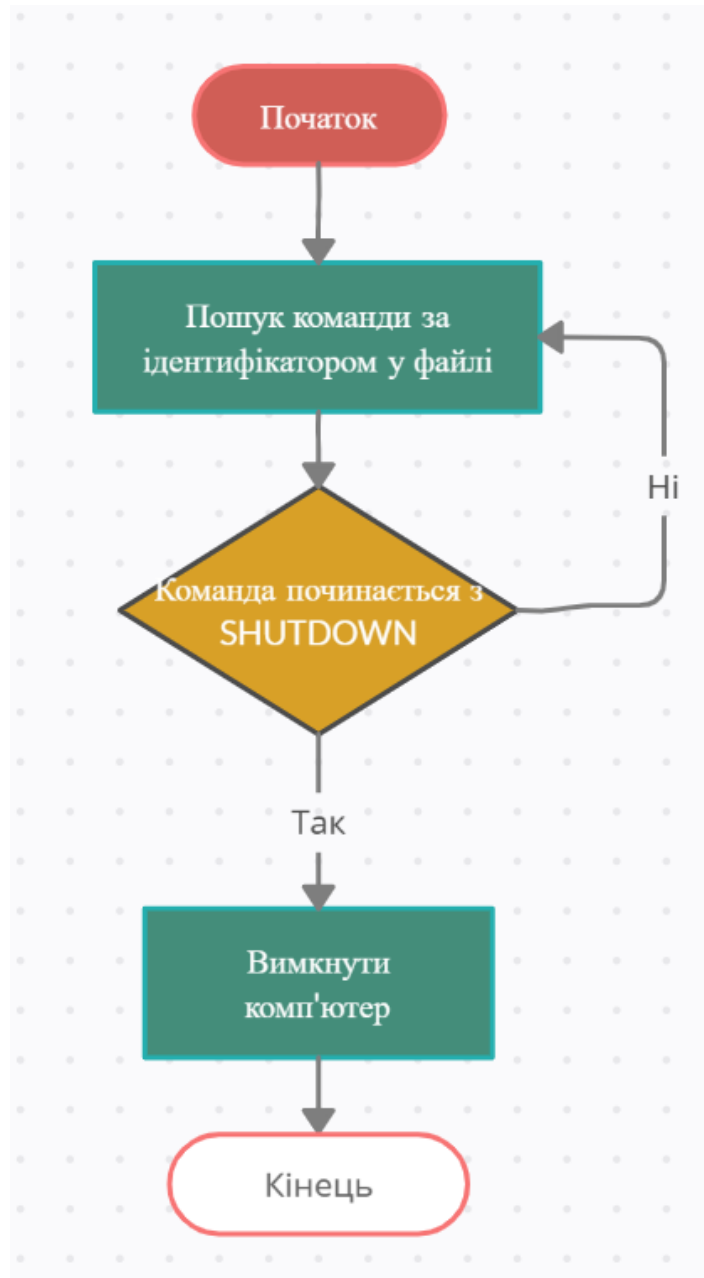


Рис. 2.5 – Діаграма алгоритму вимкнення комп'ютера

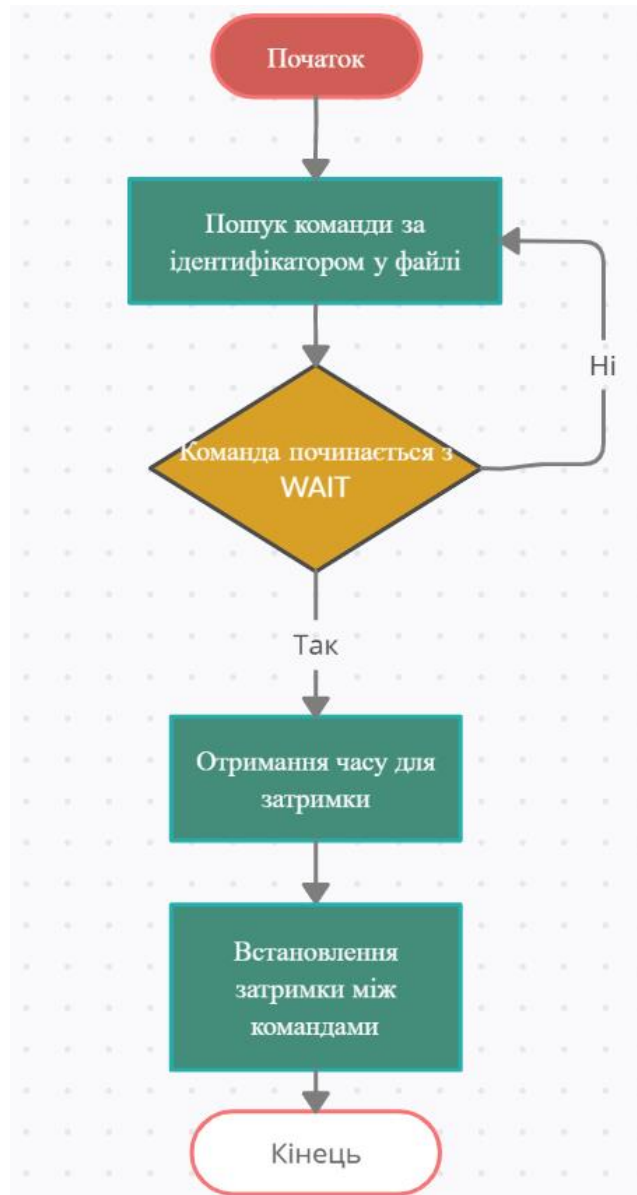


Рис. 2.6 – Діаграма алгоритму для встановлення затримки між командами

2.3 Інструкція програміста

Проект являє собою програмне забезпечення для будь-якого пристрою під керуванням ОС Windows.

Автор: Княжук В.А.

Мова програмування: C#.

Середовище програмування: Microsoft Visual Studio 2019.

Вимоги до апаратного забезпечення:

- Оперативна пам'ять: 512 Мб.
- Процесор: Intel Pentium x86 і вище, частота від 1.0 GHz.
- Touchscreen або миша.
- Клавіатура.

Вимоги до програмного забезпечення:

- Операційна система: Windows 7 та вище.
- .Net Framework 4.5.1.

2.4 Інструкція оператора

Для створення макросу користувач повинен відкрити текстовий документ id.txt, або двічі натиснути на іконку програми на панелі задач.



Рис. 2.7 – Іконка програми на панелі задач

Після відкриття текстового документу користувач повинен задати ідентифікатор для макросу, після чого встановити символ “>” та почати писати команди для макросу. Після закінчення вводу команд та збереження текстового файлу користувач може користуватися макросом.

Для коректної роботи програми на ПК користувача повинен бути встановлений .Net Framework 4.5.1 або вище. Мінімальний набір файлів для коректної роботи проекту: файл програми (IdOpener.exe)

Для запуску програми потрібно перейти до папки з її файлами та натиснути ЛКМ на файлі **IdOpener.exe**.

Після цього з'явиться іконка на панелі задач як показано на рисунку 2.7.

Якщо натиснути правою кнопкою миші на іконку програми, з'явиться меню де користувач може побачити поточну позицію миші, увімкнути та вимкнути перехоплення подій клавіатури та вийти з програми. Представлено на рисунку 2.8

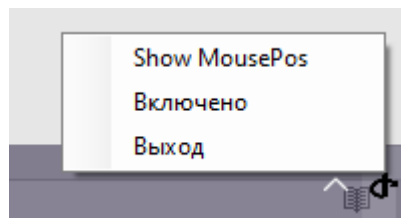


Рис. 2.8 – Меню

Для відображення поточної позиції миші потрібно натиснути на пункт меню “Show MousePos”. Представлено на рисунку 2.9

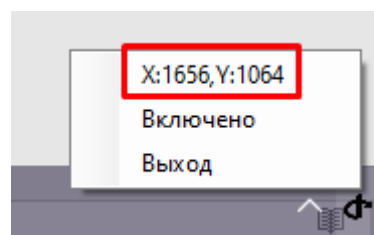


Рис. 2.9 – Позиція миші

Розглянемо докладніше алгоритм написання деяких макросів.

Макрос для відкриття папки та сайту одночасно.

Відкриваємо текстовий документ та задаємо будь-який ідентифікатор. Представлено на рисунку 2.10.

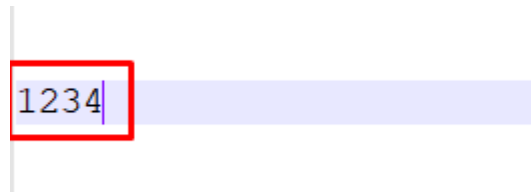


Рис. 2.10 – Ідентифікатор макросу

Після написання ідентифікатору користувач повинен встановити символ “>”. Предствлено на рисунку 2.11



Рис. 2.11 – Встановлення символу “>”

Тепер користувач може писати функцію яку буде виконувати програма після введення алгоритму. В нашому випадку потрібно відкрити папку D:\Диплом та сайт google.com. Представлено на рисунку 2.12



Рис. 2.12 – Відкриття папки D:\Диплом

Для одночасного відкриття папки та сайту потрібно використовувати символ “|”. Представлено на рисунку 2.13

```
1234>D: \Диплом | www.google.com
```

Рис. 2.13 – Відкриття диску D та сайту google.com

Після написання функції потрібно зберегти текстовий документ та натиснути задані клавіші на клавіатурі. Представлено на рисунках 2.14 та 2.15

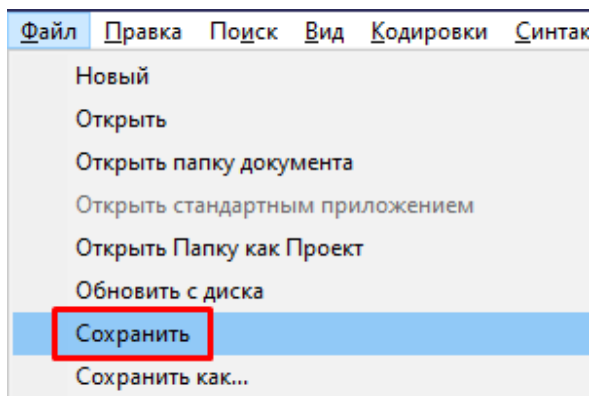


Рис. 2.14 – Збереження документу

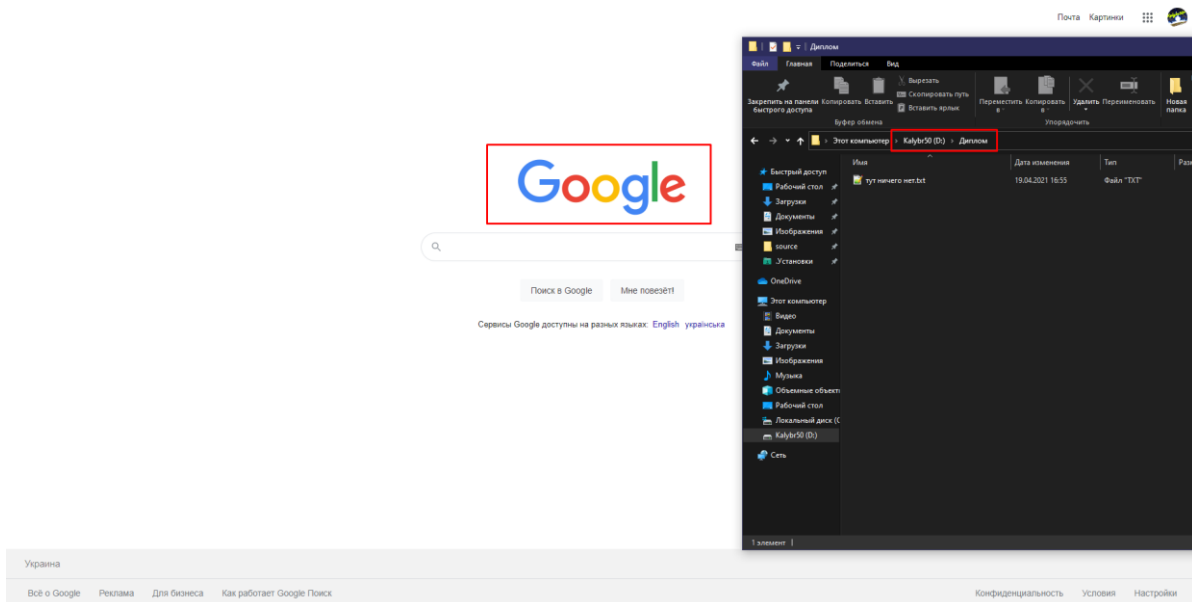


Рис. 2.15 – Відкриття папки та сайту

Макрос для встановлення позиції миші, введення тексту та відправки подій клавіш миші

Відкриваємо текстовий документ та задаємо будь-який ідентифікатор. Представлено на рисунку 2.16.


A screenshot of a text input field with a light blue background. The number "5678" is entered in the field and is highlighted with a red rectangular border.

Рис. 2.16 – Ідентифікатор макросу

Після написання ідентифікатору користувач повинен встановити символ “>”. Предствлено на рисунку 2.17


A screenshot of a text input field with a light blue background. The text "5678>" is entered in the field and is highlighted with a red rectangular border.

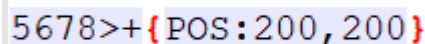
Рис. 2.17 – Встановлення символу “>”

Перед тим як писати макрос користувач повинен встановити символ “+” для використання розширених функцій. Предствлено на рисунку 2.18

A screenshot of a text input field with a light blue background. The text "5678>+" is entered in the field and is highlighted with a red rectangular border.

Рис. 2.18 - Встановлення символу “+”

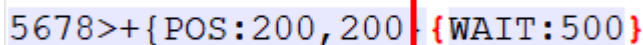
Усі розширені команди потрібно писати у фігурних скобках. Для встановлення позиції миші потрібно написати {POS:X,Y}, замість X та Y потрібно написати координати точки на екрані, куди потрібно перемістити курсор. Представлено на рисунку 2.19



```
5678>+{POS:200,200}
```

Рис. 2.19 – Встановлення позиції миші

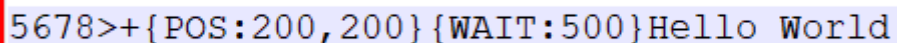
Для того, щоб створити затримку між командами, потрібно написати {WAIT:мс}, де “мс” – це час в мілісекундах. Представлено на рисунку 2.20



```
5678>+{POS:200,200}{WAIT:500}
```

Рис. 2.20 – Встановлення затримки між командами

Для написання тексту не потрібно використовувати фігурні скобки. Представлено на рисунку 2.21



```
5678>+{POS:200,200}{WAIT:500>Hello World
```

Рис. 2.21 – Введення тексту

Після написання функції потрібно зберегти текстовий документ та натиснути задані клавіші на клавіатурі. Представлено на рисунку 2.22

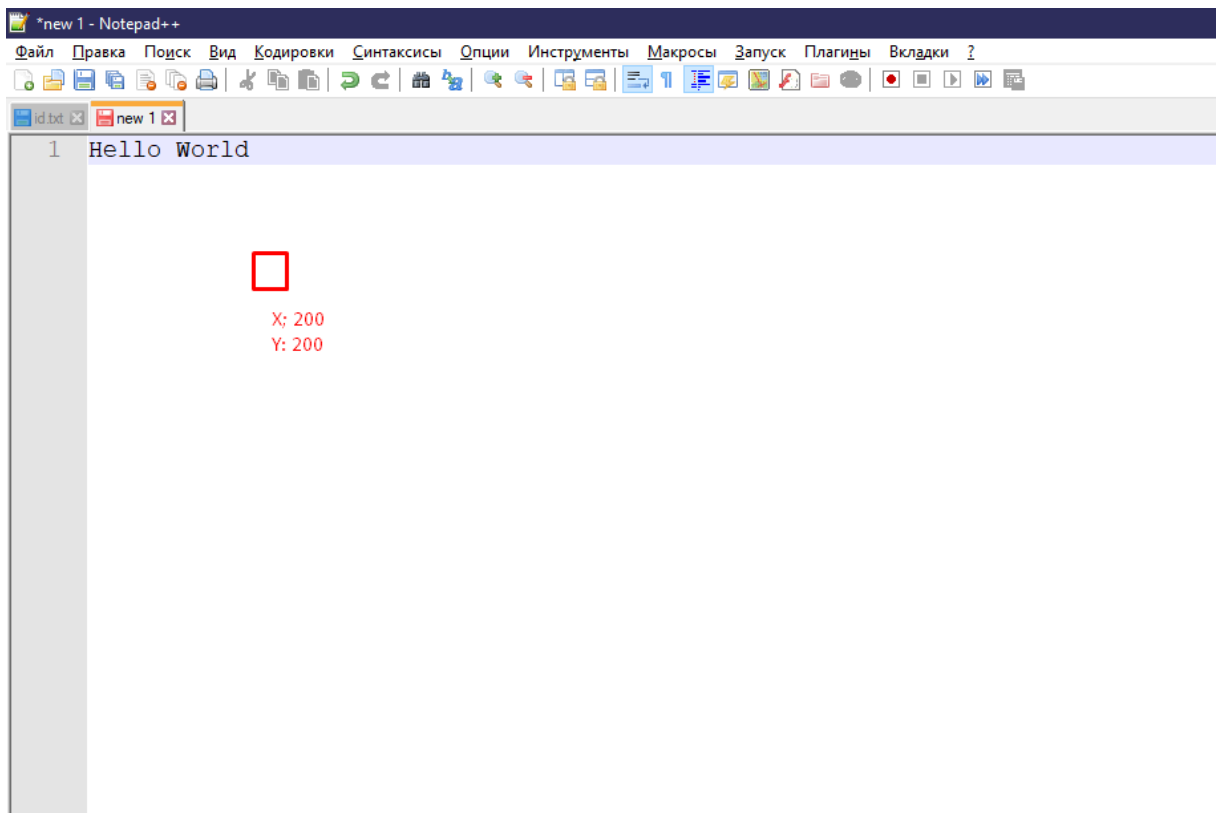


Рис. 2.22 – Виконання дій макросу

3 ОРГАНІЗАЦІЙНО-ЕКОНОМІЧНА ЧАСТИНА

3.1 Оцінка вартості програмного продукту

Прогрес в економіці, промисловості, науці і техніці, у сфері освіти в даний час багато в чому залежить від масового впровадження обчислювальної техніки. Будь-який комп'ютер в процесі роботи використовує засоби програмного забезпечення.

Розробка програмних засобів вимагає певних інтелектуальних і трудових витрат, а також обов'язкового використання комп'ютерної техніки, що визначає особливості розрахунку собівартості програмного продукту.

Оцінка вартості програмування розглядається як оцінка затрат коштів і часу, необхідних для виконання етапів проекту.

Вартість програмного продукту включає:

- собівартість програмного продукту;
- плановий прибуток.

3.1.1 Розрахунок часу

Загальний час на створення програми складається з різних компонентів. Структура загального часу на створення програмного продукту представлена в таблиці 3.1

Таблиця 3.1 – Структура загального часу на створення програмного продукту.

№ етапу	Позначення часу даного етапу	Зміст етапу
1	$T_{\text{ПО}}$	Підготовка опису завдання
2	$T_{\text{О}}$	Опис завдання
3	$T_{\text{А}}$	Розробка алгоритмів
4	$T_{\text{БС}}$	Розробка блок-схеми алгоритмів
5	$T_{\text{П}}$	Проектування інтерфейсу
6	$T_{\text{Н}}$	Написання програми (кодування)
7	$T_{\text{ВТ}}$	Відладка і тестування програми
8	$T_{\text{Д}}$	Оформлення документації, інструкції користувачеві, записки пояснення

Час розраховується в людино-годинах, причому $T_{\text{ПО}}$ та $T_{\text{Д}}$ береться по фактично відпрацьованому часу, а час останніх етапів визначається розрахунковий по умовному числу команд Q .

Умовне число команд Q визначається за формулою

$$Q = q \cdot z, \quad (3.1)$$

де q - коефіцієнт, що враховує умовне число команд залежно від типу завдання. Значення коефіцієнта q вибирається з таблиці 3.2.

Значення коефіцієнта z вибирається з таблиці 3.3.

Таблиця 3.2 – Значення коефіцієнта q.

Тип завдання	Значення коефіцієнта q
Завдання обліку	1400...1500
Завдання оперативного управління	1500...1700
Завдання планування	3000...3500
Багатоваріантні завдання	4500...5000
Комплексні завдання	5000...5500

Для даного завдання приймається коефіцієнт $q = 1450$.

Програмні продукти за мірою новизни можуть бути віднесені до однієї з чотирьох груп:

- група А - розробка принципово нових завдань;
- група Б - розробка оригінальних програм;
- група В - розробка програм з використанням типових рішень;
- група Г - разове типове завдання.

Для даного завдання міра новизни: В

За складністю програмні продукти можуть бути віднесені до однієї з трьох груп:

- 1 - алгоритми оптимізації і моделювання систем;
- 2 - завдання обліку, звітності і статистики;
- 3 - стандартні алгоритми.

Дане завдання може бути віднесено до другої групи складності високого рівня.

Коефіцієнт (з) визначається з таблиці 3.3 в залежності від складності і міри новизни.

Таблиця 3.3 – Значення коефіцієнта з.

Мова програмування	Група складності	Міра новизни			
		А	Б	В	Г
Високого рівня	1	1,38	1,26	1,15	0,69
	2	1,30	1,19	1,08	0,65
	3	1,20	1,10	1,00	0,60
Низького рівня	1	1,58	1,45	1,32	0,79
	2	1,49	1,37	1,24	0,74
	3	1,38	1,26	1,15	0,69

Для даного завдання коефіцієнт $z = 1,00$.

За формулою 3.1 визначається умовне число команд Q .

$$Q = 1450 \cdot 1,00 = 1450,0 \text{ [чол. / годин].}$$

Визначається час, витрачений на кожен етап створення програмного продукту:

- $T_{ПО}$ (час на підготовку опису завдання), береться по факту і для даного завдання

$$T_{ПО} = 30,0 \text{ чол / годин.}$$

- T_O (час на опис завдання) визначається за формулою

$$T_O = Q \cdot B / (50 \cdot K), \quad (3.2)$$

де B - коефіцієнт обліку змін завдання. Коефіцієнт B залежно від складності завдання і числа змін вибирається в інтервалі від 1,2 до 1,5. Для даного завдання $B = 1,3$

K - коефіцієнт, що враховує кваліфікацію програміста. Значення коефіцієнта вибирається з таблиці 3.4.

Таблиця 3.4 – Значення коефіцієнта К.

Стаж програміста	Значення коефіцієнта К
до 2-х років	0,8
від 2 до 3 років	1,0
від 3 до 5 років	1,1...1,2
від 5 до 10 років	1,2...1,3
понад 10 років	1,3...1,5

В даному випадку коефіцієнт $K = 0,8$.

За формулою 3.2 підраховується час на опис завдання.

$$T_O = 1450,0 \cdot 1,3 / (50 \cdot 0,8) = 47,13 \text{ [чол. / годин]}.$$

- T_A (час на розробку алгоритмів) розраховується за формулою

$$T_A = Q / (50 * K), \quad (3.3)$$

За формулою 3.3 підраховується час на розробку алгоритмів.

$$T_A = 1450,0 / (50 \cdot 0,8) = 36,25 \text{ [чол. / годин]}.$$

- T_{BC} (час на розробку блок-схеми) визначається аналогічно T_A за формулою 3.3 і складає

$$T_{BC} = 1450,0 / (50 \cdot 0,8) = 36,25 \text{ [чол. / годин]}.$$

- T_H (час написання програми на мові програмування) визначається за формулою

$$T_H = Q \cdot 1,5 / (50 \cdot K), \quad (3.4)$$

За формулою 3.4 підраховується час написання програми на мові програмування.

$$T_H = 1450,0 \cdot 1,00 / (50 \cdot 0,8) = 36,25 \text{ [чол. / годин]}.$$

- T_{III} (час на проектування інтерфейсу) визначається за формулою

$$T_{III} = Q / 50, \quad (3.5)$$

За формулою 3.5 підраховується час на проектування інтерфейсу програми.

$$T_{III} = 1450,0 / 50 = 29 \text{ [чол. / годин]}.$$

- T_{BT} (час відладки і тестування програми) визначається за формулою

$$T_{BT} = Q \cdot 4,2 / (50 \cdot K), \quad (3.6)$$

За формулою 3.6 підраховується час відладки і тестування програми.

$$T_{BT} = 1450,0 \cdot 4,2 / (50 \cdot 0,8) = 152,25 \text{ [чол. / годин]}.$$

- T_D (час на оформлення документації), береться по факту і для даного завдання

$$T_D = 35,0 \text{ чол / годин}.$$

Підраховується загальний час на створення програмного продукту за формулою спрощеного варіанту

$$T = T_{III} + T_{BT} + T_D, \quad (3.7)$$

$$T = 29 + 152,25 + 35,0 = 216,25 \text{ [чол. / годин]}.$$

3.1.2 Розрахунок заробітної плати виконавця робіт із створення програмного продукту

Основна ЗП визначається за формулою

$$ЗП_{осн.} = (ЗП_{1р} \cdot Кт \cdot Т) / (Чр \cdot тр.д.) \cdot (1 + П/100), \quad (3.8)$$

де ЗП 1р - місячна зарплата 1-го розряду, грн.;

Кт - тарифний коефіцієнт, відповідний розряду тарифної сітки, за яким працює виконавець;

Т - загальний час на створення програмного продукту, чол. / годин;

Чр - кількість робочих днів в місяць;

тр.д. - тривалість робочого дня в годинах;

П - відсоток премії.

Для даного завдання:

- ЗП 1р = 1011,00 грн.;
- виконавець має 14 розряд;
- для 14 розряду тарифний коефіцієнт Кт = 3,36;
- загальний час створення програмного продукту Т = 216,25 чол. / годин;
- кількість робочих днів Чр = 28;
- тривалість робочого дня тр.д. = 8 годин;
- відсоток премії П=13%.

Таким чином, визначаємо основну заробітну плату виконавця робіт із створення програмного продукту.

$$ЗП_{осн.} = (1011,00 \cdot 3,36 \cdot 216,25) / (28 \cdot 8) \cdot (1 + 13/100) = 3705,75 \text{ [грн]}.$$

Додаткова заробітна плата береться у розмірі 13 % від основної і розраховується за формулою

$$\text{ЗП дод.} = \text{ЗПосн.} \cdot 13 / 100, \quad (3.9)$$

$$\text{ЗП дод.} = 3705,75 \cdot 13 / 100 = 481,75 \text{ [грн]}.$$

Загальна заробітна плата розраховується за формулою

$$\text{ЗПзагальна} = \text{ЗПосн} + \text{ЗПдод.}, \quad (3.10)$$

$$\text{ЗПзагальна} = 3705,75 + 481,75 = 4187,5 \text{ [грн]}.$$

3.1.3 Розрахунок нарахувань на заробітну плату (єдиного соціального внеску)

Єдиний соціальний внесок нараховується на заробітну плату за формулою 3.11 і складає 35,6 % від загальної заробітної плати

$$\text{ЄСВ} = \text{ЗПзагальна} \cdot 35,6 / 100, \quad (3.11)$$

$$\text{ЄСВ} = 4187,5 \cdot 35,6 / 100 = 1490,75 \text{ [грн]}.$$

3.1.4 Розрахунок витрат на збереження та експлуатацію ПЕОМ, що відносяться до даного програмного продукту

Витрати на збереження та експлуатацію ПЕОМ визначаються у розмірі 130% від основної заробітної плати працівників, що забезпечують функціонування ПЕОМ.

До таких витрат відносяться витрати на:

- основну заробітну плату працівників, що забезпечують функціонування ПЕОМ (інженер-електронник; системний програміст; оператор);
- основна ЗП адміністративного і допоміжного персоналу;
- амортизаційні відрахування;
- витрати на електричну енергію;
- витрати на матеріали (до їх числа входять диски, картриджі і папір для принтерів та інше);
- витрати на профілактику ПЕОМ з периферією;
- витрати на опалювання виробничих площ;
- витрати на обслуговування виробничих площ;
- інші виробничі витрати.

Розрахунок витрат на збереження та експлуатацію ПЕОМ, що відносяться до даного програмного продукту за формулою

$$В \text{ зб. експл.} = ЗП \text{ осн.} \cdot 130 / 100, \quad (3.12)$$

$$В \text{ зб. експл.} = 4187,75 \cdot 130 / 100 = 5444,0 \text{ [грн]}.$$

3.2 Розрахунок собівартості програмного продукту

У собівартість програмного продукту входять наступні елементи:

- основна заробітна плата виконавця робіт із створення програмного продукту;
- додаткова заробітна плата виконавця робіт із створення програмного продукту;
- нарахування на заробітну плату (єдиний соціальний внесок);
- витрати на збереження та експлуатацію ПЕОМ, що відносяться до даного програмного продукту;

– інші витрати.

Інші витрати складають 10% від суми перших чотирьох елементів і розраховуються за формулою

$$I.\text{вит.}=(ЗП\text{ осн.} + ЗП\text{ дод.} + \text{ЄСВ} + В\text{ зб. експл.}) \cdot 10 / 100 \quad (3.13)$$

$$I.\text{вит.}=(4187,75 + 481,75 + 1490,75 + 5444,0) \cdot 10 / 100 = 1160,42 \text{ [грн].}$$

Визначається собівартість програмного продукту за формулою

$$\text{Сп.п.} = ЗП\text{ осн.} + ЗП\text{ дод.} + \text{ЄСВ} + В\text{ зб.експл.} + I.\text{вит.} \quad (3.14)$$

$$\text{Сп.п.} = 4187,75 + 481,75 + 1490,75 + 5444,0 + 1160,42 = 12764,67 \text{ [грн].}$$

Структура собівартості програмного продукту відображається в таблиці 3.5.

Таблиця 3.5 – Структура собівартості програмного продукту

Елементи структури	Сума грн.	Питома вага %
1 Основна заробітна плата виконавця	4187,75	32
2 Додаткова заробітна плата виконавця	481,75	4
3 Нарахування на заробітну плату (ЄСВ)	1490,75	11
4 Витрати збереження та експлуатацію ПП	5444,0	42
5 Інші витрати	1160,42	9
Собівартість програмного продукту	12764,67	100

3.3 Розрахунок ціни програмного продукту

Ціна програмного продукту складається з декількох компонентів і розраховуються за формулою

$$Ц = \text{Сп.п.} + П + \text{ПДВ}, \quad (3.15)$$

де Сп.п.- собівартість програмного продукту, грн.;

П - плановий прибуток, грн.;

ПДВ - податок на додану вартість, грн.

П – прибуток складає 40% від повної собівартості програмного продукту і розраховуються за формулою

$$П = \text{Сп.п.} \cdot 40 / 100 \quad (3.16)$$

$$П = 12764,67 \cdot 40 / 100 = 5105,87 \text{ [грн]}.$$

ПДВ - податок на додану вартість, який береться у розмірі 20% від суми собівартості і прибутку розраховуються за формулою

$$\text{ПДВ} = (\text{Сп.п.} + П) \cdot 20 / 100, \quad (3.17)$$

$$\text{ПДВ} = (12764,67 + 5105,87) \cdot 20 / 100 = 3574,10 \text{ [грн]}.$$

За формулою 3.15 визначається ціна програмного продукту.

$$\text{Ц} = 12764,67 + 5105,87 + 3574,10 = 21444,64 \text{ [грн]}.$$

3.4 Зведена таблиця показників

Результати розрахунків відображаються в підсумковій таблиці 3.6.

Таблиця 3.6 – Результати розрахунків

Найменування показника	Сума, грн.
Собівартість програмного продукту	127
	64,67
Прибуток	510
	5,87
ПДВ	357
	4,10
Ціна програмного продукту	214
	44,81

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ

Безпека життєдіяльності людини - комплексний стан, при якому вірогідність здійснення негативного ризику мінімальна в будь-яких умовах її діяльності. Отже, безпека пов'язана з визначенням ризику. Ризик - вірогідність виникнення події з певними небажаними наслідками - травма, хвороба, смерть, аварія - руйнування об'єкта. Підвищення вірогідності проявлення ризику формує небезпеку, як протилежність безпеки. Безпека життєдіяльності людини стала особливо актуальною в наш час, коли могутність цивілізації стала порівняною з природною. У світі нараховується більше 500 млн інвалідів і кожний п'ятий - результат нещасного випадку у виробництві, побуті або в природі. В середньому протягом року на Землі травмується біля 20 млн, а гине 500 тис. людей. У США щорічно гине у виробничих умовах 14 тис. людей, отримують травми 2,3 млн, в тому числі у дорожньо-транспортних подіях гине більше 60 тисяч і понад 2 млн стають інвалідами. В Україні лише в побутових умовах гине в середньому 50000 людей за рік при щорічному зростанні майже на 10%. Світова статистика свідчить, що 60-80% нещасних випадків це результат невміння передбачити, розпізнати приховану небезпеку, невміння оцінити ризик і узгоджувати його із своїми можливостями, які визначаються психо-фізіологічними властивостями організму та станом техніки.

4.1 Вимоги до приміщень і організації робочих місць

Виробниче середовище – найбільш небезпечне для людини, саме на виробництві на неї діють підвищені рівні шуму, вібрації; гостріше проявляються психофізіологічні фактори (стрес, втома); в повітрі робочої зони часто присутні токсичні речовини; на виробництві людина більше взаємодіє з машинами та механізмами, що є потенційно небезпечним. Проводячи третину свого життя, чи майже половину активного часу на виробництві, необхідно дбати про забезпечення безпечних та прийнятних умов праці.

Площа, відведена на одне робоче місце має становити не менше 6 кв. м., а об'єм – не менше 20 куб. м. Конструкція робочого місця повинна забезпечувати

підтримання оптимальної робочої пози (тобто такої, яка дозволяє працівникові виконувати роботу з мінімальним напруженням тіла, і яка дозволяє уникнути перевтоми в ході і після закінчення робочого процесу). Раціональна робоча поза має важливе значення для збереження здоров'я працівника, оскільки тривале перебування його в незручній і напруженій позі може призвести до таких захворювань, як сколіоз (викривлення хребта), варикозне розширення вен, плоскостопість тощо. Установлено, що робота в зігнутому положенні збільшує затрати енергії на 20%, а при значному нахиленні — на 45% порівняно з прямим положенням корпусу.

Відповідні робочі місця заборонено облаштовувати у підвальних або цокольних приміщеннях будинків. В обладнанні приміщень забороняється використання полімерних матеріалів, що виділяють шкідливі хімічні речовини. Також слід приділити увагу забезпеченню достатнім для здійснення роботи рівнем освітлення (природного та штучного – у темну пору доби) та звукоізоляції. Для регуляції рівня освітлення природним світлом бажано застосовувати жалюзі. Окрім того, у приміщеннях, де здійснюється робота з комп'ютерами, щодня має здійснюватися вологе прибирання з метою недопущення запиленості підлоги та меблів.

Заземлені конструкції, що знаходяться в приміщеннях, де розміщені робочі місця операторів (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном), мають бути надійно захищені діелектричними щитками або сітками з метою недопущення потрапляння людини під напругу.

Особливої уваги заслуговують заходи дотримання протипожежної безпеки. Так, у всьому офісі лінії електромережі мають бути забезпечені від виникнення короткого замикання, а також від перепадів мережевої напруги, що може спричинити збої в роботі електронно-обчислювальної техніки. Приміщення (окрім тих, де розташовуються сервери) мають бути оснащені системою автоматичної пожежної сигналізації та вогнегасниками. Під час монтажу та експлуатації ліній електромережі необхідно повністю унеможливити виникнення електричного джерела загоряння внаслідок короткого замикання та перевантаження проводів, обмежувати застосування проводів з легкозаймистою ізоляцією і, за можливості,

застосовувати негорючу ізоляцію. У приміщенні, де одночасно експлуатуються понад п'ять комп'ютерів, на помітному та доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення.

Загальні вимоги

Облаштування робочих місць, обладнаних відео терміналами, повинно забезпечувати:

- належні умови освітлення приміщення і робочого місця, відсутність відблисків;
- оптимальні параметри мікроклімату (температура, відносна вологість, швидкість руху, рівень іонізації повітря);

- належні ергономічні характеристики основних елементів робочого місця

- а також враховувати такі небезпечні та шкідливі фактори:

- наявність шуму та вібрації
- м'яке рентгенівське випромінювання
- електромагнітне випромінювання;
- ультрафіолетове та ультрачервоне випромінювання;
- електростатичне поле між екраном та оператором;
- наявність пилу, озону, оксидів азоту й аероіонізації.

Будівлі та приміщення, в яких експлуатують ЕОМ та виконують їх обслуговування, налагодження і ремонт, повинні відповідати вимогам:

НиП 2.09.02-85 „Производственные здания”,

СНиП 2.09.04-87 „Административные и бытовые здания”,

„Правил устройства электроустановок”, затверджених
Головдерженергонаглядом СРСР 1984 р. (ПВЕ),

„Правил технической эксплуатации электроустановок потребителей”,
затверджених Головдерженергонаглядом СРСР 21.12.84 (ПТЕ).

Правил безпечної експлуатації електроустановок споживачів, затверджених
наказом Держнаглядохоронпраці 09.01.98 №4, зареєстрованих у Мін'юсті України
10.02.98 № 93/2533 (ПБЕ),

СНиП 2.01.02-85 „Противопожарные нормы”;

ГОСТ 12.1.004 „ССБТ. Пожарная безопасность. Общин требования безопасности”,

Правил пожежної безпеки в Україні, затверджених наказом Управління Державної пожежної охорони МВС України від 14.06.95 № 400, зареєстрованих в Міністерстві юстиції України 14.07.95 за № 219/755,

СНиП 2.08.02-89 „Общественные здания и сооружения” з доповненнями, затвердженими наказом Держкоммістобудування України від 29.12.94 № 106,

Сн 512-78 „Инструкция по проектированию зданий и помещений для электронно-вычислительных машин”, затверджених Держбудом СРСР,

ДСанПіН 3.3.3.-007-98 „Державні санітарні правила і норми роботи з візуальними дисплеями терміналами електронно-обчислювальних машин”, затверджених МОЗ України 10.12.98, а також вимогами нормативно-технічної та експлуатаційної документації заводу-виробника ЕОМ, чинних санітарних норм, санітарних норм і правил, правил у сфері охорони праці.

Для всіх споруд і приміщень, в яких експлуатуються відеотермінали та ЕОМ, повинна бути визначена категорія з вибухопожежної та пожежної безпеки відповідно до ОНТП 24-86 „Определение категорий помещений и зданий по взрывопожарной и пожарной опасности”, затверджених МВС СРСР 27.02.86, та клас зони згідно з ПВЕ. Відповідні позначення повинні бути нанесені на вхідні двері приміщення.

Будівлі і ті їх частини, в яких розташовуються ЕОМ, повинні мати не нижче II ступеня вогнестійкості. Приміщення для обслуговування, ремонту та налагодження ЕОМ повинні належати за пожежовибухобезпекою до категорії В відповідно до ОНТП 24-86, а за класом приміщення – до П-Іа за ПВЕ. Якщо відповідно до СНиП 2.09.02-85 ці приміщення повинні бути відокремленими від приміщень іншого призначення протипожежними стінами, то межа їх вогнестійкості визначається відповідно до СНиП 2.01.02 -85.

Неприпустимим є розташування приміщень категорій А іБ (ОНТП 24-86), а також виробництва з мокрими технологічними процесами поряд з приміщеннями, де розташовуються ЕОМ, виконується їх обслуговування, налагодження і ремонт, а також над такими приміщеннями або під ними. Виробничі приміщення, в яких

розташовані ЕОМ, не повинні межувати з приміщеннями, де рівні шуму та вібрації перевищують норму (механічні цехи, майстерні тощо).

Робочі місця з відеотерміналами або персональними ЕОМ у приміщеннях з джерелами шкідливих виробничих факторів повинні розміщуватися в ізольованих кабінах з обладнаним повітрообміном. Стіни кабін виготовляються з негорючих матеріалів. Дозволяється виготовляти їх зі скла та металевих конструкцій. У кабіні мусить бути оглядове вікно (вікна). Висота оглядового вікна мусить бути не менше 1,5 м, а відстань від підлоги не більше 0,8 м.

Відповідно до ДСанПіН 3.3.3-007-98 „Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин”, затверджених МОЗ України 10.12.98, є неприпустимим розташування приміщень для роботи з відеотерміналами та ЕОМ у підвалах та цокольних поверхах.

Площу приміщень, в яких розташовують відеотермінали, визначають згідно з чинними нормативними документами з розрахунку на одне робоче місце, обладнане відеотерміналом: площа – не менше 6,0 м², обсяг – не менше 20,0 м³, з урахуванням максимальної кількості осіб, які одночасно працюють у зміні. Стіни, стеля, підлога приміщень, де розміщені ЕОМ, повинні виготовлятися з матеріалів, дозволених для оздоблення приміщень органами державного санітарно-епідеміологічного нагляду.

Обслуговування, налагодження і ремонт ЕОМ, вузлів та блоків ЕОМ слід виконувати в окремому приміщенні (майстерні).

При цьому робочі місця електромеханіків повинні бути оснащені спеціальним обладнанням та захисними засобами відповідно до вимог до організації робочого місця з обслуговування, ремонту та налагодження ЕОМ.

У приміщеннях для обслуговування, ремонту та налагодження ЕОМ слід передбачити можливість вологого очищення поверхонь комунікації та опалювальних приладів.

Підлога всієї зони обслуговування, ремонту та налагодження ЕОМ, вузлів та блоків ЕОМ має бути вкрита діелектричними килимками, термін використання

яких після їх випробування на електричну міцність не закінчився, або викладена ізольованими підстилками (шириною не менше ніж 0,75-0,8 м) для ніг.

Приміщення, в яких проводиться паяння, крім того, повинні відповідати вимогам СП 952-72 „Санитарные правила организации процессов пайки мелких изделий сплавами, содержащими свинец”, затверджених головним санітарним лікарем СРСР 20.03.72.

Приміщення комп'ютерних класів (залів), в яких проводиться навчання на ЕОМ, крім зазначених у пункті 1.2, повинні мати суміжне приміщення (лаборантську) площею не менше 18 м² з двома входами: в учбове приміщення та в коридор (на сходову клітку).

Заземлені конструкції, що знаходяться в приміщеннях (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном тощо), мають бути надійно захищені діелектричними щитками або сітками від випадкового дотику. У приміщеннях з ЕОМ слід щоденно проводити вологе прибирання.

У приміщеннях з ЕОМ повинні бути медичні аптечки першої допомоги.

Приміщення з ЕОМ, крім приміщень, в яких розміщуються ЕОМ типу ЕС, СМ та інші великі ЕОМ загального призначення, повинні бути оснащені системою автоматичної пожежної сигналізації відповідно до вимог Переліку однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежегасіння та пожежної сигналізації, затвердженого наказом Міністерства внутрішніх справ України від 20.11.97 №779 і зареєстрованого в Міністерстві юстиції України 28.11.97 за №567/2371, та СНиП 2.04.09-84 „Пожарная автоматика зданий и сооружений” з димовими пожежними сповіщувачами та переносними вуглекислотними вогнегасниками з розрахунку 2 шт. На кожні 20 м² площі приміщення з урахуванням гранично допустимих концентрацій вогнегасної рідини відповідно до вимог Правил пожежної безпеки в Україні. В інших приміщеннях допускається встановлювати теплові пожежні сповіщувачі.

Приміщення, в яких розміщуються ЕОМ типу ЕС, СМ та інші великі ЕОМ загального призначення, обладнуються системою автоматичної пожежної сигналізації та засобами пожежегасіння відповідно до вимог Переліку однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками

пожежегасіння та пожежної сигналізації, СНиП 2.04.09-84, СН 512-78, Правил пожежної безпеки в Україні та вимог нормативно-технічної та експлуатаційної документації заводу-виробника.

Піходи до засобів пожежегасіння повинні бути вільними.

Приміщення для відпочинку осіб, які працюють з ЕОМ, призначені для приймання їжі, психологічного розвантаження, та інші побутові приміщення повинні обладнуватись відповідно до вимог СНиП 2.09.04-87 „Административные и бытовые здания”, з урахуванням максимальної кількості працівників, що одночасно працюють у зміні.

Власник організує проведення досліджень шкідливих та небезпечних факторів виробничого середовища і трудового процесу на робочих місцях осіб, які працюють з ЕОМ, відповідно до чинних законодавчих та інших нормативно-правових актів і Порядку проведення атестації робочих місць за умовами праці, затвердженого постановою Кабінету Міністрів України від 01.08.92 №442.

Санітарно-гігієнічні вимоги

Умови праці осіб, які працюють з ЕОМ, повинні відповідати I або II класу згідно з Гігієнічною класифікацією праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу №4137-86, затвердженою МОЗ СРСР 12.08.86.

Вимоги до освітлення

Приміщення з ЕОМ повинні мати природне і штучне освітлення відповідно до СПиП II-4-79 „Естественное и искусственное освещение”.

Природне світло повинно проникати через бічні світлопрорізи, зорієнтовані, як правило, на північ чи північний схід, і забезпечувати коефіцієнт природної освітленості (КПО) не нижче 1,5%. Розрахунки КПО проводяться відповідно до СНиП II-4-79.

При виробничій потребі дозволяється експлуатувати ЕОМ у приміщеннях без природного освітлення за узгодженням з органами державного нагляду за охороною праці ц органами та установами санітарно-епідеміологічної служби.

Вікна приміщень з відеотерміналами повинні мати регульовальні пристрої для відкривання, а також жалюзі, штори, зовнішні козирки тощо.

Штучне освітлення приміщення з робочими місцями, обладнаними відеотерміналами ЕОМ загального та персонального користування, має бути обладнане системою загального рівномірного освітлення. У виробничих та адміністративно-громадських приміщеннях, де переважають роботи з документами, допускається вживати систему комбінованого освітлення (додатково до загального освітлення встановлюються світильники місцевого освітлення). Загальне освітлення має бути виконане у вигляді суцільних або переривчастих ліній світильників, що розміщуються збоку від робочих місць (переважно зліва) паралельно лінії зору працівників. Допускається застосувати світильники таких класів світлорозподілу:

- світильники прямого світла – П;
- переважно прямого світла – Н;
- переважно відбитого світла – В.

При розташуванні відеотерміналів ЕОМ за периметром приміщення лінії світильників штучного освітлення повинні розміщуватися локально над робочими місцями.

Для загального освітлення необхідно застосовувати світильники із розсіювачами та дзеркальними екранними сітками або віддзеркалювачами, укомплектовані височастотними пускорегулювальними апаратами (ВЧ ПРА). Допускається застосовувати світильники без ВЧ ПРА тільки при використанні моделі з технічною назвою „Кососвет”. Застосування світильників без розсіювачів та екранних сіток забороняється.

Як джерело світла при штучному освітленні повинні застосовуватися, як правило, люмінесцентні лампи типу ЛБ. При обладнанні відбивного освітлення у виробничих та адміністративно-громадських приміщеннях можуть застосовуватися металогалогенові лампи потужності до 250 Вт. Допускається у світильниках місцевого освітлення застосовувати лампи розжарення.

Яскравість світильників загального освітлення в зоні кутів випромінювання від 50° до 90° відносно вертикалі в подовжній і поперечній площинах повинна складати не більше 200 кд/м^2 , а захисний кут світильників повинен бути не більшим за 40° .

Коефіцієнт запасу (Кз) відповідно до СНиП II-4-79 для освітлювальної установки загального освітлення слід приймати рівним 1,4.

Коефіцієнт пульсації повинен не перевищувати 5% і забезпечуватися застосуванням газорозрядних ламп у світильниках загального і місцевого освітлення.

При відсутності світильників або розташовані поруч світильники загального освітлення необхідно підключати до різних фаз трифазної мережі.

Рівень освітленості на робочому столі в зоні розташування документів має бути в межах 300-500 лк. У разі неможливості забезпечити даний рівень освітленості системою загального освітлення допускається застосування світильників місцевого освітлення, але при цьому не повинно бути відблисків на поверхні екрану та збільшення освітленості екрану більше ніж 300 лк.

Світильники місцевого освітлення повинні мати напівпрозорий відбивач світла з захисним кутом не меншим за 40°.

Необхідно передбачити обмеження прямої близькості від джерела природного та штучного освітлення, при цьому яскравість поверхонь, що світяться (вікна, джерела штучного світла) і перебувають у полі зору, повинна бути не більшою за 200 кд/м².

Необхідно обмежувати відбитий блиск шляхом правильного вибору типу світильників та розміщення робочих місць відносно джерел природного та штучного освітлення. При цьому яскравість відблисків на екрані відеотерміналу не повинна перевищувати 40 кд/м².

Необхідно обмежувати нерівномірність розподілу яскравості в полі зору осіб, що працюють з відеотерміналом, при цьому відношення значень яскравості робочих поверхонь не повинно перевищувати 3:1, а робочих поверхонь і навколишніх предметів (стіни, обладнання) – 5:1.

Необхідно використовувати систему вимикачів, що дозволяє регулювати інтенсивність штучного освітлення залежно від інтенсивності природного, а також дозволяє освітлювати тільки потрібні для роботи зони приміщення.

Для забезпечення нормованих значень освітлення в приміщеннях з відеотерміналами ЕОМ загального та персонального користування необхідно

очищати віконне скло та світильники не рідше ніж 2 рази на рік, та своєчасно проводити заміну ламп, що перегоріли

4.2 Вимоги охорони праці при експлуатації комп'ютерних систем

Щодня перед початком роботи оператор повинен:

- оглянути своє робоче місце; про виявлення ознак пошкодження обладнання інформувати свого безпосереднього керівника;
- відрегулювати освітленість на робочому місці, переконатися в відсутності відблисків на екрані комп'ютера, відсутності зустрічного світла;
- перевірити правильність підключення обладнання ЕОМ до електромережі;
- очистити екран комп'ютера від пилу та інших забруднень;
- перевірити правильність організації робочого місця й за необхідності провести відповідні коригування.

Оператор під час роботи зобов'язаний:

- виконувати тільки ту роботу, яку йому було доручено;
- підтримувати порядок і чистоту на робочому місці;
- тримати відкритими всі вентиляційні отвори обладнання;
- коректно закрити всі активні завдання у разі припинення роботи з комп'ютером;
- негайно відключити комп'ютером від електричної мережі у разі виникнення аварійної ситуації.

У ході виконання робіт оператор комп'ютера повинен:

- витримувати відстань від очей до екрана комп'ютером в межах 60 - 70см;
- дотримуватися внутрішньозмінного режиму праці та відпочинку, регламентованих перерв у роботі, а саме (при 8-годинній денній робочій зміні):
 - для розробників програм - тривалістю 15 хвилин через кожен годину роботи;

- для інших категорій працівників - тривалістю 15 хвилин через кожні дві години роботи;
- для операторів комп'ютерного набору - тривалістю 10 хвилин, після кожної години роботи.

Під час регламентованих перерв рекомендується виконувати комплекси вправ для очей, рук, хребта, поліпшення мозкового кровообігу тощо. Про виявлення несправності обладнання або інших факторів, які створюють загрозу для життя або здоров'я працівників, необхідно негайно інформувати свого безпосереднього керівника.

Не допускається:

- виконання ремонту та налагодження комп'ютерної техніки безпосередньо на робочому місці оператора;
- зберігання біля комп'ютера паперу, дискет, інших носіїв інформації, запасних блоків, деталей тощо, якщо вони не використовуються для поточної роботи;
- відключення захисних пристроїв, самочинні зміни в конструкції комп'ютера;
- використання комп'ютерів, на екранах яких під час роботи з'являються нехарактерні сигнали, нестабільне зображення на екрані тощо;
- доторкання до задньої панелі системного блоку при включеному живленні;
- вимикання живлення під час виконання активного завдання;
- попадання вологи на поверхню системного блоку, монітора, клавіатури, дисководів, принтерів та інших пристроїв;
- приймання напоїв та їжі на робочому місці.

Після закінчення роботи з використанням необхідно дотримуватися такої послідовності вимикання обладнання:

- закрити всі активні завдання;
- переконатися у відсутності дискет та дисків у дисководах;

- використавши опцію "Завершення роботи" у меню "Пуск", вимкнути живлення системного блоку;
- вимкнути живлення всіх комп'ютерів;
- вимкнути блок аварійного живлення (за наявності);
- відключити комп'ютер від електромережі, при цьому забороняється тягнути штепсельну вилку за дріт.

У випадку виникнення аварійної ситуації оператор зобов'язаний:

- у всіх випадках виявлення пошкодження проводів електричного живлення, несправності заземлення та інших пошкодженнях електрообладнання, виникненні запаху гарі, диму - негайно вимкнути електричне живлення і повідомити про аварійну ситуацію свого безпосереднього керівника й чергового електрика;
- при попаданні людини під електричну напругу негайно звільнити її від дії струму шляхом вимкнення електричного живлення, до прибуття лікаря надати потерпілому долікарську медичну допомогу;
- при будь-яких випадках порушень роботи технічного обладнання або програмного забезпечення негайно викликати представника технічної служби з питань експлуатації обчислювальної техніки;
- у випадку виникнення різі в очах, різкого погіршення зору, виникнення головного болю, больових відчуттів у пальцях та кистях рук, посилення серцебиття - негайно припинити роботу з використанням ЕОМ, повідомити про те, що сталося, свого безпосереднього керівника й звернутися до медичної установи;
- при загорянні обладнання негайно відключити його від електромережі;
- про загорання повідомити свого безпосереднього керівника, оперативного чергового, пожежну службу; ужити заходів щодо ліквідації вогню за допомогою вуглекислотного або порошкового вогнегасника.

В наступній частині статті йтиметься про особливості виробничого травматизму та професійних захворювань, пов'язаних з використанням комп'ютерів, а також буде висвітлено проблему відповідальності за порушення

чинного законодавства України, якими регулюються правила роботи з комп'ютерною технікою.

Користувачі комп'ютерних систем повинні слідкувати за тим, щоб відеотермінали, комп'ютерних систем, периферійні пристрої комп'ютерних систем та устаткування для обслуговування, ремонту та налагодження комп'ютерних систем були справними і випробуваними відповідно до чинних нормативних документів.

Щоденно перед початком роботи необхідно проводити очищення екрану відеотерміналу від пилу та інших забруднень.

Після закінчення роботи відеотермінал та персональна комп'ютерних систем повинні бути відключені від електричної мережі.

У разі виникнення аварійної ситуації необхідно негайно відключити відеотермінал та комп'ютерні системи від електричної мережі.

При використанні комп'ютерних систем та відеотерміналами лазерних принтерів потрібно дотримуватись вимог Санітарних норм та правил устрою та експлуатації лазерів №5804-91, затверджених Міністерством охорони здоров'я СРСР в 1991р.

При потребі, для захисту від електромагнітних, електростатичних та інших полів можуть застосовуватися спеціальні технічні засоби, що мають відповідний сертифікат або санітарно-гігієнічний висновок акредитованих органів щодо їх захисних властивостей.

Є неприпустимими такі дії:

- виконання обслуговування, ремонту та
- налагодження комп'ютерних систем безпосередньо на робочому місці користувача комп'ютерної системи;
- зберігання біля відеотермінала та комп'ютерних систем паперу, дискет, інших носіїв інформації, запасних блоків, деталей тощо, якщо вони не використовуються для поточної роботи;

- відключення захисних пристроїв, самочинне проведення змін у конструкції та складі комп'ютерних систем, устаткування або їх технічне налагодження;
- робота з відеотерміналами, в яких під час роботи з'являються нехарактерні сигнали, нестабільне зображення на екрані тощо;
- праця на матричному принтері зі знятою (трохи піднятою) верхньою кришкою

Вплив шуму. Відомо, що шум він несприятливо діє на слуховий аналізатор та інші органи та системи організму людини. Визначальне значення щодо такої дії має інтенсивність шуму, його частотний склад, тривалість щоденного впливу, індивідуальні особливості людини, а також специфіка виробничої діяльності. Ті види діяльності, у яких поєднується напружена розумова робота та інтенсивне використання комп'ютера (редагування тексту, верстка оригіналу, "запуск" та відлагодження програм тощо) характеризується відчутним впливом навіть незначних рівнів шуму. Цей вплив виражається у зниженні розумової працездатності, швидкій втомлюваності, послабленні уваги, появі головного болю та ін.

Основними заходами та засобами боротьби з шумом є:

- зниження рівнів шуму в джерелі його утворення (застосовується, як правило, в процесі проектування);
- використання звукопоглинаючих та звукоізолюючих засобів;
- раціональне планування виробничих приміщень та робочих місць.

На комп'ютеризованих робочих місцях основними джерелами шуму є вентилятори системного блоку, накопичувачі, принтери ударної дії. Для зниження рівнів шуму на робочих місцях рекомендується розмістити друкувальні пристрої ударної дії (матричні, шрифтові принтери тощо) в іншому приміщенні, або огородити їх звукоізолюючими екранами.

Під час виконання робіт з ВДТ і ПК у виробничих приміщеннях значення характеристик вібрації на робочих місцях не повинні перевищувати допустимих значень, визначених СН 3044-84 та ГОСТ 12.1.012-90.

Для зниження вібрації обладнання, пристрої, пристосування необхідно встановлювати на спеціальні амортизуючі прокладки, передбачені нормативними документами.

Головним небезпечним фактором безпеки при експлуатації та ремонті пристрою є електробезпека.

Електрика — сукупність явищ, зумовлених існуванням, рухом взаємодією електрично заряджених тіл або часток. Електричний струм — це упорядкований (спрямований) рух електрично заряджених часток. Струм у металах зумовлений наявністю вільних електронів, у електролітах — іонів. Звичайно силою, яка викликає такий рух, є сила з боку електричного поля усередині провідника, яке визначається електричною напругою на кінцях провідника. Наявність електричного струму в провідниках призводить до їх нагрівання, зміни хімічного складу, створення магнітного поля.

Електричні прилади, установки, обладнання, з яким людина має справу, становлять для неї велику потенційну небезпеку, яка посилюється тим, що органи чуття людини не можуть на відстані виявити наявності електричної напруги, як, наприклад, теплову, світлову чи механічну енергію. Тому захисна реакція організму виявляється тільки після безпосереднього потрапляння під дію електричного струму.

Другою особливістю дії електричного струму на організм людини є те, що струм, проходячи через людину, діє не тільки в місцях контактів і на шляху протікання через організм, а й викликає рефлекторні порушення нормальної діяльності окремих органів (серцево-судинної системи, системи дихання). Третя особливість — це можливість одержання електротравм без безпосереднього контакту із струмопровідними частинами — при переміщенні по землі поблизу ушкодженої електроустановки (у випадку замикання на землю), ураження через електричну дугу.

Електричний струм, проходячи через тіло людини, зумовлює перетворення поглинутої організмом електричної енергії в інші види і спричиняє термічну, електролітичну, механічну і біологічну дію. Найбільш складною є біологічна дія, яка притаманна тільки живим організмам.

Статистика свідчить, що більше половини всіх електротравм становлять опіки. Вони важко піддаються лікуванню, тому що глибоко проникають у тканини організму. В електроустановках напругою до 1 кВ найчастіше спостерігаються опіки контактного виду при дотиканні тіла до струмопровідних частин.

Опіки можливі при проходженні через тіло людини струму більше 1 А. Тільки при великому струмі тканини, які уражаються, нагріваються до температури 60—70°C і вище, при якій згортається білок і з'являються опіки.

В електроустановках напругою вище 1 кВ опіки можуть виникнути при випадковому наближенні частин тіла людини до струмопровідних частин на небезпечну відстань; при цьому збільшується напруга електричного поля і внаслідок ударної іонізації діелектрика (повітряного проміжку) опір цього проміжку зменшується, його «пробиває» електричний розряд — електрична дуга, температура якої досягає приблизно 4000°C. Електричний струм протікає через дугу і тіло людини. За такої високої температури і великої кількості тепла, яка виділяється при проходженні струму через тіло, потерпілий одержує тяжкі опіки, його м'язи скорочуються, дуга і ланцюг струму розриваються.

Майже у всіх випадках включення людини в електричний ланцюг на її тілі і в місцях дотикання спостерігаються «електричні знаки» сіро-жовтого кольору круглої або овальної форми.

При опіках від впливу електричної дуги можлива металізація шкіри частками металу дугової плазми. Уражена ділянка шкіри стає твердою, набуває кольору солей металу, які потрапили в шкіру.

Електролітична дія струму виявляється у розкладанні органічної рідини, в тому числі крові, яка є електролітом, та в порушенні її фізико-хімічного складу.

Біологічна дія струму виявляється через подразнення і збудження живих тканин організму, а також порушення внутрішніх біологічних процесів.

Механічна дія струму призводить до розриву тканин організму внаслідок електродинамічного ефекту, а також миттєвого вибухоподібного утворення пари з тканинної рідини і крові.

Внаслідок дії електричного струму або електричної дуги виникає електротравма. Електротравми умовно поділяють на загальні і місцеві. До місцевих

травм належать опіки, електричні знаки, електрометалізація шкіри, механічні пошкодження, а також електрофтальмія, запалення очей внаслідок впливу ультрафіолетових променів електричної дуги). Загальні електротравми називають також електричними ударами. Вони є найбільш небезпечним видом електротравм.

При електричних ударах виникає збудження живих тканин, судомне скорочення м'язів, параліч м'язів опорно-рухового апарату, м'язів грудної клітки (дихальних), м'язів шлуночків серця. У першому випадку судомне скорочення м'язів не дозволяє людині самостійно уникнути дотикання з електроустановкою. При паралічі дихання припиняється газообмін і постачання організму киснем, внаслідок чого настає задуха. При паралічі серця його функції або припиняються повністю, або деякий час продовжуються в режимі тріпотіння (фібриляції).

Фібриляція — це безладдя в скороченні серцевих м'язів. При цьому порушується кровообмін, що також спричиняє смерть. Медичною практикою встановлено, що після припинення роботи серця і дихання внаслідок кисневого голодування через 5—6 хвилин гинуть клітини центральної нервової системи, відбувається втрата свідомості і припиняється управління функціями всіх органів тіла. Цей стан має назву «клінічна (уявна) смерть», оскільки клітини інших органів тіла ще живі. Але при більшій тривалості відсутності дихання і кровообміну відбувається припинення життєдіяльності решти клітин і органів і настає біологічна смерть. Отже, якщо відразу після звільнення людини з-під впливу електричного струму, не пізніше перших 5—6 хвилин, приступити до долікарської допомоги шляхом штучного дихання і непрямого масажу серця, то існує ймовірність запобігти смерті потерпілого.

Розрізняють три ступені впливу струму при проходженні через організм людини (змінний струм):

- відчутний струм — початок болісних відчуттів (до 0—1,5мА);
- невідпускний струм — судоми і біль, важке дихання (10—15 мА);
- фібриляційний струм — фібриляція серця при тривалості дії струму 2—3 с, параліч дихання (90—100мА).

Змінний струм небезпечніший за постійний. При струмі 20—25 мА пальці судомно стискають узятий в руку предмет, який опинився під напругою, а м'язи

передпліччя паралізуються і людина не може звільнитися від дії струму. У багатьох паралізуються голосові зв'язки: вони не можуть покликати на допомогу.

За статистикою, найменший струм, за якого настає смерть, становить 0,8 мА, можливість смерті людей від слабкого струму пояснюється тим, що результат електроураження залежить не тільки від дії струму на серце або органи дихання, але і від впливу на нервову систему з її індивідуальними особливостями.

Електричний опір людини в основному визначається опором рогового шару шкіри, який можна розглядати як тонкий і недосконалий діелектрик, а м'язи і кров — як провідник. Опір шкіри залежить від площі поверхні і щільності контакту, а також від сили струму і тривалості його дії.

Чим вони більші, тим менший опір шкіри (із збільшенням тривалості протікання струму збільшується нагрівання шкіри, потовиділення, в ній відбуваються електролітичні зміни). Опір шкіри також залежить від прикладеної напруги, оскільки вже при напрузі 10—38В пробивається верхній роговий шар шкіри, вона втрачає властивості діелектрика і стає провідником — через тіло людини проходить струм. При напрузі 127—220В і вище шкіра вже майже не впливає на опір тіла.

Суха шкіра має опір приблизно 100 кОм. Опір внутрішніх органів, тканин і судин значно менший (приблизно 800 Ом). При розрахунках опір тіла людини приймають за 1кОм. Має значення шлях струму через тіло і особливо місця входу і виходу струму. Із можливих шляхів проходження струму через тіло людини найбільш небезпечним є той, при якому уражається головний мозок (голова — руки, голова — ноги), серце і легені (руки — ноги).

Несприятливий мікроклімат (підвищена температура, вологість, струмопровідний пил) збільшують небезпеки ураження струмом, тому що волога (піт), пил знижують опір шкіри. При ураженні електричним струмом насамперед необхідно надати потерпілому першу долікарську допомогу.

Висновки

В даній дипломній роботі було створено системне програмне забезпечення, яке дозволяє користувачу створювати власні макроси для подальшого їх використання.

Дипломна робота була написана у середовищі програмування Visual Studio мовою програмування C#.

Проект розвивається, до нього додаються нові функції та можливості. Також, оскільки програмна система використовує для керування комп'ютером набір функцій ядра операційної системи було вивчено особливості виклику та виконання базового набору команд, що дозволяють виконувати програмно основні дії користувача.

В результаті було створено програмну систему, яка дозволяє навіть недосвідченому користувачеві виконувати основні дії по керуванню набором програм персонального комп'ютера.

Список використаних джерел

1. Константайн Л., Локвуд Л. Разработка программного обеспечения: Пер. с англ. - Питер, 2004, -470 с.
2. Мандел Т.: Разработка пользовательского интерфейса Пер. с англ. – М. : ДМК Пресс, 2001, - 416 с.
3. Закон України Про оплату праці.
4. Закон України Про підприємництво.
5. Закон України Про оподаткування прибутку підприємств.
6. ГОСТ 19.001-77 ЄСПД. Загальні положення.
7. ГОСТ 19.002-80 ЄСПД. Схеми алгоритмів і програм. Правила виконання.
8. ГОСТ 19.003-80 ЄСПД. Схеми алгоритмів і програм. Позначення умовні графічні.
9. ГОСТ 19.105-78 ЄСПД. Загальні вимоги до програмних документів.
10. ДСТУ – 2293 – 93. Система стандартів безпеки праці. Охорона праці. Терміни та визначення.
11. СН 4088-86 Мікроклімат виробничих приміщень.
12. А.Матросов. HTML 4.
13. Кузнецов М.В. и Симдянов И.В. Учебник РНР ГОСТ 19.401-78 ЄСПД. Текст програми. Вимоги до змісту і оформлення.
14. ГОСТ 19.402-78 ЄСПД. Опис програми.
15. ГОСТ 19.504-79 ЄСПД. Інструкція програміста.
16. ГОСТ 19.505-79 ЄСПД. Інструкція оператора.
17. ГОСТ 19.506-79 ЄСПД. Опис мови. Вимоги до змісту і оформлення
18. Бойко В., Савинков В. Проектирование баз данных информационных систем. -М.: Финансы и статистика, 1989.
19. Дейт К. Введение в систему баз данных.- М.:Мир, 1998.
20. Джексон Г. Проектирование реляционных баз данных. -М.: Мир, 1991.

Додаток А
Код програми
Form1.cs

```
using System;
using System.Diagnostics;
using System.Drawing;
using System.IO;
using System.Text;
using System.Windows.Forms;
using GlobalKeyboardHook;

namespace IdOpener
{
    public partial class Form1 : Form
    {
        GKH gkh = new GKH();
        string id = "";
        int time = 30;
        string keyy = "";
        bool keys = false;
        bool ActOn = true;
        bool sCoord = false;

        public Form1()
        {
            InitializeComponent();
            notifyIcon1 = new NotifyIcon();
            notifyIcon1.Visible = true;
            notifyIcon1.Click += NotifyIcon1_Click;
            notifyIcon1.DoubleClick += NotifyIcon1_DoubleClick;
            notifyIcon1.ContextMenuStrip = contextMenuStrip1;
        }
    }
}
```

```
        notifyIcon1.Icon = IdOpener.Properties.Resources.ico;
    }

private void NotifyIcon1_DoubleClick(object sender, EventArgs e)
{
    if (FileHandler.FileExists())
    {
        FileHandler.OpenFile();
    }
    else
    {
        FileHandler.CreateFile();
    }
}

private void NotifyIcon1_Click(object sender, EventArgs e)
{
    return;
}

private void Form1_Load(object sender, EventArgs e)
{
    gkh.KeyDown += Gkh_KeyDown;
    if (!FileHandler.FileExists())
    {
        FileHandler.CreateFile();
    }
    WindowState = FormWindowState.Minimized;
    this.ShowInTaskbar = false;
    notifyIcon1.Visible = true;
    this.Hide();
}
```

```
}
```

```
private void Gkh_KeyDown(object sender, KeyEventArgs e)
```

```
{
```

```
    if(ActOn == false) { return; }
```

```
    if (timer1.Enabled == false) { timer1.Start(); }
```

```
    time = 30;
```

```
    if (e.KeyCode.ToString().StartsWith("D") &&
```

```
!e.KeyCode.ToString().EndsWith("D"))
```

```
    {
```

```
        id += e.KeyCode.ToString().Replace("D", "");
```

```
    }
```

```
    else
```

```
    {
```

```
        if (e.KeyCode.ToString().Contains("NumPad"))
```

```
        {
```

```
            id += e.KeyCode.ToString().Replace("NumPad", "");
```

```
        }
```

```
    else
```

```
    {
```

```
        id += e.KeyCode.ToString();
```

```
    }
```

```
}
```

```
using (StreamReader sr = new StreamReader("id.txt", Encoding.Default))
```

```
{
```

```
    string line = "";
```



```

while ((line = sr.ReadLine()) != null)
{
    if (!line.StartsWith("//"))
    {
        string[] splitt = line.Split(new Char[] { '>' });
        if (id.ToLower() == splitt[0].ToLower())
        {
            try
            {
                if (splitt[1].StartsWith("+"))
                {
                    System.Threading.Thread.Sleep(300);
                    string text = splitt[1].Replace("+", "");
                    int lenght = id.Length;
                    for (int i = 0; i < lenght; i++)
                    {
                        SendKeys.Send("{BS}");
                    }
                    for(int i = 0; i < text.Length; i++)
                    {
                        if (text[i].ToString() == "{")
                        {
                            keys = true;
                        }
                        if (keys == false)
                        {
                            SendKeys.Send(text[i].ToString());
                        }
                        if (text[i].ToString() == "}")
                        {
                            keyy = keyy.Replace("{", "");
                        }
                    }
                }
            }
        }
    }
}

```

```

if (keyy.StartsWith("POS:"))
{
    keyy = keyy.Replace("POS:", "");
    keyy = keyy.Replace(" ", "");
    string[] pos = keyy.Split(new char[] { ',' });
    int x = Convert.ToInt32(pos[0]);
    int y = Convert.ToInt32(pos[1]);
    Cursor.Position = new Point(x, y);
    goto end;
}
if (keyy.StartsWith("WAIT:"))
{
    keyy = keyy.Replace("WAIT:", "");
    keyy = keyy.Replace(" ", "");
    int wait = Convert.ToInt32(keyy);
    System.Threading.Thread.Sleep(wait);
    goto end;
}
if (keyy.StartsWith("SEND:"))
{
    keyy = keyy.Replace("SEND:", "");
    keyy = keyy.Replace(" ", "");
    if(keyy == "LMB")
    {
        uint X = (uint)Cursor.Position.X;
        uint Y = (uint)Cursor.Position.Y;
        MouseHandler.Click(MouseHandler.Keys.Left,
X, Y);

        goto end;
    }
    if (keyy == "RMB")

```

```

    {
        uint X = (uint)Cursor.Position.X;
        uint Y = (uint)Cursor.Position.Y;
        MouseHandler.Click(MouseHandler.Keys.Right,
X, Y);

        goto end;
    }
    SendKeys.Send("{ " + keyy + "}");
}
if(keyy.StartsWith("OPEN:"))
{
    keyy = keyy.Replace("OPEN:", "");
    keyy = keyy.Replace(" ", "");
    System.Threading.Thread.Sleep(500);
    Process.Start(keyy);
}
if (keyy.StartsWith("TASKKILL:"))
{
    keyy = keyy.Replace("OPEN:", "");
    keyy = keyy.Replace(" ", "");
    System.Threading.Thread.Sleep(500);
    Process.Start("taskkill", "/F /IM " + keyy + ".exe");
}
if (keyy == "SHUTDOWN")
{
    var p = new ProcessStartInfo("shutdown", "/s /t 0");
    p.CreateNoWindow = true;
    p.UseShellExecute = false;
    Process.Start(p);
}
if (keyy == "REBOOT")

```

```
    {
        var p = new ProcessStartInfo("shutdown", "/r /t 0");
        p.CreateNoWindow = true;
        p.UseShellExecute = false;
        Process.Start(p);
    }
end:
keys = false;
keyy = "";
}

if (keys == true)
{
    keyy += text[i].ToString();
}
}
else
{
    if (splitt[1].Contains("|"))
    {
        string[] sline = splitt[1].Split(new Char[] { '|' });
        for (int i = 0; i < sline.Length; i++)
        {
            System.Threading.Thread.Sleep(200);
            Process.Start(sline[i]);
        }
    }
    else
    {
        System.Threading.Thread.Sleep(500);
```

```
        Process.Start(splitt[1]);
    }
    id = "";
    }
}
catch {
}
break;
}
}
}
}
}
```

```
private void timer1_Tick(object sender, EventArgs e)
{
    time--;
    if (time == 0)
    {
        id = "";
        time = 30;
    }
    label1.Text = id;
}
```

```
private void Form1_Deactivate(object sender, EventArgs e)
{
    if (this.WindowState == FormWindowState.Minimized)
    {
        this.ShowInTaskbar = false;
        notifyIcon1.Visible = true;
    }
}
```

```
    }  
}
```

```
bool cClose = true;
```

```
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
```

```
{  
    e.Cancel = cClose;  
    this.Hide();  
}
```

```
private void выходToolStripMenuItem_Click(object sender, EventArgs e)
```

```
{  
    cClose = false;  
    Environment.Exit(0);  
}
```

```
private void активацияONToolStripMenuItem_Click(object sender,  
EventArgs e)
```

```
{  
    if(ActOn == true)  
    {  
        активацияONToolStripMenuItem.Text = "Выключено";  
        ActOn = false;  
    }  
    else  
    {  
        активацияONToolStripMenuItem.Text = "Включено";  
        ActOn = true;  
    }  
}
```

```

    }

    private void timer2_Tick(object sender, EventArgs e)
    {
        x0Y0ToolStripMenuItem.Text = "X:" + MousePosition.X + ",Y:" +
MousePosition.Y;
    }

    private void x0Y0ToolStripMenuItem_Click(object sender, EventArgs e)
    {
        if(sCoord == false)
        {
            timer2.Start();
            sCoord = true;
        }
        else
        {
            timer2.Stop();
            x0Y0ToolStripMenuItem.Text = "Show MousePos";
            sCoord = false;
        }
    }
}
}

```

FileHandler.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using System.Windows.Forms;
using System.Diagnostics;

namespace IdOpener
{
    class FileHandler
    {
        public static bool FileExists()
        {
            if (File.Exists("id.txt"))
                return true;
            else
                return false;
        }

        public static void CreateFile()
        {
            using (StreamWriter sw = new StreamWriter("id.txt"))
            {
                sw.WriteLine("//Example 1: 12345>D:\\ - При вводе на клавиатуре
12345 открывает локальный диск D");
            }
        }
    }
}
```


sw.WriteLine("//Example 2: 45678>www.google.com - При вводе на клавиатуре 45678 открывает google.com");

sw.WriteLine("//Example 3: 56789>C:|D:|E:|www.google.com - При вводе 56789 открывает диски C,D,E и сайт www.google.com");

sw.WriteLine("//Example 4: 78901>+Hello World - При вводе на клавиатуре 78901 пишет текст Hello World");

sw.WriteLine("//Example 5: 11223>+Hello{SEND:ENTER}How are you?{SEND:ENTER}What are you doing? - При вводе на клавиатуре 11223 пишет текст и нажимает на клавиши");

sw.WriteLine("//Example 6:
33445>+{WAIT:500}{POS:100,100}{SEND:LMB} - Через 500 мс мышь будет установлена на координаты X:100, Y:100 и нажата ЛКМ");

sw.WriteLine("//Example 7: 55667>+{SHUTDOWN} - При вводе на клавиатуре 55667 выключает компьютер. {REBOOT} - перезагрузить");

sw.WriteLine("//Example 8: 77889>+{TASKKILL:explorer} - При вводе 77889 завершает процесс explorer.exe");

}

}

public static void OpenFile()

{

Process.Start("id.txt");

}

}

}

MouseHandler.cs

```
using System;
using System.Runtime.InteropServices;

namespace IdOpener
{
    class MouseHandler
    {
        [DllImport("user32.dll", CharSet = CharSet.Auto, CallingConvention =
CallingConvention.StdCall)]
        public static extern void mouse_event(uint dwFlags, uint dx, uint dy, uint
cButtons, uint dwExtraInfo);

        //Mouse actions
        private const int MOUSEEVENTF_LEFTDOWN = 0x02;
        private const int MOUSEEVENTF_LEFTUP = 0x04;
        private const int MOUSEEVENTF_RIGHTDOWN = 0x08;
        private const int MOUSEEVENTF_RIGHTUP = 0x10;

        public enum Keys
        {
            Left = 1,
            Right = 2
        }

        public static void Click(Keys key, uint x, uint y)
        {
            if (key == Keys.Left)
            {
                mouse_event(MOUSEEVENTF_LEFTDOWN |
MOUSEEVENTF_LEFTUP, x, y, 0, 0);
            }
        }
    }
}
```

```
        if (key == Keys.Right)
        {
            mouse_event(MOUSEEVENTF_RIGHTDOWN |
MOUSEEVENTF_RIGHTUP, x, y, 0, 0);
        }
    }
}
```

GKH.cs

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Runtime.InteropServices;
using System.Windows.Forms;

namespace GlobalKeyboardHook {

    class GKH {

        public delegate int keyboardHookProc(int code, int wParam, ref
keyboardHookStruct lParam);

        public struct keyboardHookStruct {
            public int vkCode;
            public int scanCode;
            public int flags;
            public int time;
            public int dwExtraInfo;
        }
    }
}
```

```

const int WH_KEYBOARD_LL = 13;
const int WM_KEYDOWN = 0x100;
const int WM_KEYUP = 0x101;
const int WM_SYSKEYDOWN = 0x104;
const int WM_SYSKEYUP = 0x105;

public List<Keys> HookedKeys = new List<Keys>();

IntPtr hhook = IntPtr.Zero;

private keyboardHookProc hookProcDelegate;

public event KeyEventHandler KeyDown;

public event KeyEventHandler KeyUp;

public GKH()
{
    hookProcDelegate = hookProc;
    hook();
}

~GKH() {
    unhook();
}

```

```

public void hook()
{
    IntPtr hInstance = LoadLibrary("User32");
    hhook = SetWindowsHookEx(WH_KEYBOARD_LL,
hookProcDelegate, hInstance, 0);
}

public void unhook() {
    UnhookWindowsHookEx(hhook);
}

public int hookProc(int code, int wParam, ref keyboardHookStruct
IParam) {
    if (code >= 0) {
        Keys key = (Keys)IParam.vkCode;
        //if (HookedKeys.Contains(key)) {
            KeyEventArgs kea = new KeyEventArgs(key);
            if ((wParam == WM_KEYDOWN || wParam ==
WM_SYSKEYDOWN) && (KeyDown != null)) {
                KeyDown(this, kea) ;
            } else if ((wParam == WM_KEYUP || wParam ==
WM_SYSKEYUP) && (KeyUp != null)) {
                KeyUp(this, kea);
            }
            if (kea.Handled)
                return 1;
        //}
    }
    return CallNextHookEx(hhook, code, wParam, ref IParam);
}

```

```
[DllImport("user32.dll")]
static extern IntPtr SetWindowsHookEx(int idHook,
keyboardHookProc callback, IntPtr hInstance, uint threadId);
```

```
[DllImport("user32.dll")]
static extern bool UnhookWindowsHookEx(IntPtr hInstance);
```

```
[DllImport("user32.dll")]
static extern int CallNextHookEx(IntPtr idHook, int nCode, int
wParam, ref keyboardHookStruct lParam);
```

```
[DllImport("kernel32.dll")]
static extern IntPtr LoadLibrary(string lpFileName);
}
}
```