

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ФАХОВИЙ КОЛЕДЖ ІНЖЕНЕРІЇ ТА УПРАВЛІННЯ
НАЦІОНАЛЬНОГО АВІАЦІЙНОГО УНІВЕРСИТЕТУ»

ДОПУСТИТИ ДО ЗАХИСТУ

Заступник директора з НР

_____ О.В. Родіонова

« ____ » _____ 2021 р.

КВАЛІФІКАЦІЙНА РОБОТА

ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ

«БАКАЛАВР»

Тема: _____ Програмний модуль «Віртуальна клавіатура»

Автор: _____ О.С Шляма

Керівник проекту: _____ Н.А. Рябчук

Нормконтролер: _____ В.М. Кругляк

Київ 2021

**ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ФАХОВИЙ КОЛЕДЖ ІНЖЕНЕРІЇ ТА УПРАВЛІННЯ
НАЦІОНАЛЬНОГО АВІАЦІЙНОГО УНІВЕРСИТЕТУ»**

Циклова комісія: Інженері програмного забезпечення

Освітнього ступеня: «Бакалавр»

Спеціальність: 123 Комп'ютерна інженерія

Освітньо-професійна програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Голова циклової комісії

_____ Н.А. Рябчук

« ____ » _____ 2021 р.

ЗАВДАННЯ

на виконання дипломного проекту

Шлями Олександра Сергійовича

1. Тема роботи: «Програмний модуль «Віртуальна клавіатура»

затверджена наказом від «15» березня 2021 року № 28-Ст.

2. Термін виконання: з 12.04.2021р. по 20.06.2021р.

3. Вихідні дані: Розробити програмний модуль для написання текстової та цифрової інформації за допомогою екрану та миші

4. Зміст пояснювальної записки:

У 1 розділі проаналізувати завдання на проект та основні методи його розв'язання.

У 2 розділі навести опис основних етапів розробки.

У 3 розділі описати основні вимоги охорони праці.

КАЛЕНДАРНИЙ ПЛАН
виконання дипломної роботи

№ п/п	Етапи виконання кваліфікаційної роботи	Термін виконання етапів	Примітка
1.	Уточнення постановки задачі	12.04.2021	Виконано
2.	Аналіз літературних джерел	15.04.2021	Виконано
3.	Обґрунтування рішення	21.04.2021	Виконано
4.	Збір інформації	01.05.2021	Виконано
5.	Аналіз існуючих методів. Обґрунтування вибору мови програмування	06.05.2021	Виконано
6.	Виконання проєкту	08.06.2021	Виконано
7.	Оформлення і друк пояснювальної записки	15.06.2021	Виконано
8.	Оформлення презентації	17.06.2021	Виконано
9.	Отримання рецензій	20.06.2021	Виконано
10.	Захист проєкту		

Дипломник

(підпис, дата)

Шляма О.С.

(П.І.Б.)

Дипломний керівник

(підпис, дата)

Рябчук Н.А.

(П.І.Б.)

Консультант з охорони праці

(підпис, дата)

Пешков І.В.

(П.І.Б.)

Зміст

Вступ.....	6
1 Загальна частина	7
1.1 Постановка задачі.....	7
1.2 Теоретичні відомості	8
1.3 Аналіз засобів реалізації та обґрунтування вибору мови програмування.....	13
2 Спеціальна частина	17
2.1 Опис алгоритму створення програмного засобу	17
2.2 Функціональні та нефункціональні вимоги	18
2.3 Структура програмного модуля	20
2.4 Опис засобів реалізації	24
2.5 Порівняльний аналіз реалізованого програмного засобу та програм-аналогів.....	29
2.6 Керівництво користувача програмного модуля.....	33
2.7 Тестування роботи програмного модуля.....	37
3 Охорона праці.....	43
3.1 Характеристика умов праці програміста	43
3.2 Вимоги до виробничих приміщень	44
3.3 Розрахунок освітленості і рівня шуму	50
3.4 Заходи та засоби протипожежного захисту	53
Висновки	59
Перелік використаних джерел	60
Додаток А – Код модуля MainWindow	62

Вступ

В умовах сучасного динамічного функціонування будь-яка бізнес-організація прагне зберігати власні конкурентні переваги на ринку. Тому кожна компанія розробляє та впроваджує методи, які дають змогу не лише підтримувати стабільну позицію але й розвиватися та збільшувати масштаби діяльності у певній сфері. До таких методів можна віднести різні інноваційні рішення, залучення висококваліфікованого персоналу, вдосконалення системи управління, зміну технологічних процесів у виробництві тощо.

Але, останнім часом все більшої популярності серед компаній набирають так звані технології віртуальної реальності (VR-технології). Вони набувають істотної ваги через стрімкий розвиток інформаційно-комунікаційних та цифрових технологій, які значно полегшують та урізноманітнюють життя більшості людей і організацій.

Технології віртуальної реальності поступово з'являються в робочих просторах і використовуються в багатьох галузях, таких як: медицина, будівництво та проектування, автомобільна індустрія, військова сфера, логістика, архітектура та дизайн, освіта, спорт, інженерія, дизайн речей, туризм тощо.

Основною перевагою таких технологій є можливість повного занурення у створену реальність, додатково набута здатність відчувати і проаналізувати як позитивні сторони, так і негативні наслідки запланованих дій, а також змоделювати варіанти розвитку майбутніх подій.

Тема дипломного проектування – «Програмний модуль «Віртуальна клавіатура»»

«Віртуальна клавіатура» - програмний засіб, який представляє собою комп'ютерну клавіатуру, зображену на екрані електронно-обчислювального пристрою. Натискання на клавіші здійснюється курсором миші, або, для сенсорних екранів, пальцем користувача або стилусом. Віртуальна клавіатура може використовуватися для зменшення ризику реєстрації натискання клавіш.

1 Загальна частина

1.1 Постановка задачі

«Віртуальна клавіатура» - програма, призначена для використання клавіатури за допомогою екрану та миші.

Задоволеність користувача програмним продуктом або зручністю його використання в значній мірі визначається інтерфейсом користувача. Взагалі, задоволеність користувача – це функція невеликої кількості факторів (фактор - це причина, рушійна сила будь-якого процесу, яка визначає його характер або його окремі риси):

- 1) можливостей інтерфейсу користувача;
- 2) часу відгуку;
- 3) надійності;
- 4) пристосованості до інсталяції;
- 5) інформаційної підтримки;
- 6) пристосованості до супроводження;
- 7) інших факторів.

Методи інтерактивної взаємодії передбачають використання клавіатури або вказівних пристроїв для переміщення, вибору та маніпулювання інформацією на екрані. До методів інтерактивної взаємодії програмного засобу належать:

- 1) можливість використання миші та екрану в залежності від переваг користувача;
- 2) робота з різними типами клавіатури;
- 3) режими вводу тексту;
- 4) буфер обміну.

Реалізований програмний засіб дозволяє виконувати такі функції:

- 1) поле для вводу;
- 2) симулятор короткої текстової клавіатури;

- 3) симулятор короткої цифрової клавіатури;
- 4) можливість налаштування клавіатури під потреби користувача.

Інтерфейс програми складається з основного вікна редагування з набором наявних примітивів і робочої області. Управління програмою проводиться мишею.

Платформа розробки:

- 1) процесор: Intel (R) Core™ i3;
- 2) оперативна пам'ять: 4Gb;
- 3) жорсткий диск: HDD 500Gb;
- 4) монітор;
- 5) клавіатура;
- 6) миша;

Мінімальні вимоги коректного функціонування програмного засобу:

- 1) програмна платформи: *Windows XP/ 7/ 8/ 8.1/ 10*;
- 2) 32-розрядний (x86) або 64-розрядний (x64) процесор з тактовою частотою 1 ГГц;
- 3) відеокарта з 500 Мб відеопам'яті;
- 4) процесор з частотою 1,0 ГГц і 2 ядрами;
- 5) 1 Гб оперативної пам'яті;
- 6) 500 Мб місця на жорсткому диску.

1.2 Теоретичні відомості

Віртуальна реальність - це штучний простір, створений комп'ютерами, який має всі ознаки реальності такої, що піддається проникненню і трансформації ззовні. При цьому у віртуальній реальності можливі комунікації не лише з іншими людьми, але і з віртуальними, штучними персонажами.

Проблеми віртуальності стали актуальними в кінці ХХ ст., коли почали активно розвиватися комп'ютерні технології і пристрої, що здатні відтворювати віртуальний світ. У наш час ця проблема набула великих масштабів. Реальність

відступає на задній план перед нескінченними можливостями нереального, комп'ютерного життя.

Одна з головних проблем це ігроманія.

Ігроманія або лудоманія (лат. ludus - гра) - це занадто сильне захоплення комп'ютерними або азартними іграми.

Говорячи про віртуальні ігри, лудоманією страждає переважно молодь. Підлітки проводять за комп'ютером весь вільний час граючи в різні ігри, де перед ними відкриваються такі можливості, про які мріють всі діти: сила, вміння літати, володіння зброєю.

Віртуальність — вигаданий, уявний об'єкт, суб'єкт, категорія, ставлення, дія тощо, не присутні в цей час у реальному світі, а створені лише грою уяви людської думки, або імітовані за допомогою інших об'єктів.

Використовується в багатьох сферах сучасного життя, переважно стосовно симуляції чогось: віртуальні прилади, віртуальний хокей, віртуальне навчання, віртуальні служби знайомств і навіть віртуальна держава; рідше стосовно можливості існування, стану: віртуальна частинка, віртуальні переміщення. Проте, термін віртуалізація — заміщення реального віртуальним, застосовується зазвичай до кіберпростору — реальності, що створюється за допомогою комп'ютерів.

Автоматизація процесів на сьогоднішній день являє собою одну з концепцій управління ними, відмінна риса якої - використання інформаційних технологій. Вона передбачає широке застосування ЕОМ і програмного забезпечення і забезпечує управління інформацією, ресурсами і діями з мінімальною участю людини в даних процедурах або без такого в принципі.

Головне завдання, яке покликано реалізувати проектування автоматизації процесів - це висновок якісних показників процесів на принципово вищому рівні. Досягається воно головним чином завдяки тому, що основною перевагою автоматизованого режиму над ручним є його велика надійність.

Що, в свою чергу, сприяє:

- 1) зростання продуктивності;
- 2) прискоренню;
- 3) здешевлення;
- 4) збільшення точності і стабільності.

На сьогоднішній день автоматизація процесів в світі використовується повсюдно - від координування найскладніших виробництв до здійснення придбань в супермаркетах. Спрямованість компанії, так само як і її масштаби, в даному випадку не принципові: автоматика буквально пронизує будь-яку з них. А завдяки використанню процесорного підходу, для всієї сукупності процесів застосовуються єдині принципи автоматизації.

Хоча автоматизація процесів і має можливість реалізовуватися на різних рівнях, її принципи для кожного з них, так само як і для будь-якого з процесів, є єдиними. Дана обставина ставить умови максимально ефективної реалізації процесів, що здійснюються в автоматичному режимі і визначає принципи управління ними з боку автоматики.

До таких принципів належать:

1) узгодженість. В процесі, який контролюється автоматикою, всі операції повинні бути узгоджені як взаємно, так і з його входами і виходами. Невиконання цієї умови нерідко призводить до збоїв у виконанні поставленого завдання;

2) інтеграція. Що знаходиться під управлінням автоматики процес повинен мати можливість вбудовуватися в загальну систему організації. Відповідно до рівня автоматизації подібна інтеграція може бути реалізована по-різному. Що, втім, не змінює суті самого принципу, яка залишається без змін. Мета автоматизації будь-якого процесу - це забезпечення його взаємодії з середовищем, яка є по відношенню до нього зовнішньої.

Будь-який процес, який реалізується за допомогою автоматики, повинен реалізовуватися без людської участі або ж зі зведеним до мінімуму рівнем контролю, здійснюваного оператором. При виконанні умови, що процес реалізується коректно, людське втручання в нього не повинно мати місця.

Варто зазначити, що перераховані вище принципи можуть бути істотно деталізовані, що залежить від типу процесів і рівня автоматизації.

Так, автоматизація виробничих процесів передбачає такі принципи, як:

- 1) безперервність;
- 2) спеціалізація;
- 3) пропорційність і т.п.

Автоматизація процесів є необхідною умовою підтримки управління компанією, що зачіпають всі шаблі її ієрархії, а рівень автоматизації визначається відповідно до рівня управління. Основних рівнів управління три:

- 1) оперативний;
- 2) тактичний;
- 3) стратегічний.

На підставі описаної вище класифікації виділяють такі рівні автоматизації:

1) рівень виконавців. Характерними процесами для нього є ті, які виконуються регулярно. Їх автоматизація працює на досягнення оперативних цілей (приклад - реалізація одного з процесів виробництва), підтримка заданих параметрів (приклад - автопілот авіалайнера) і збереження потрібного режиму функціонування (приклад - температурні показники котла);

2) рівень управління виробництвом. В даному випадку метою автоматизації є коректне розподіл завдань між процесами, що знаходяться нижче по ієрархічній драбині. Як приклад можна привести здійснення планування виробництва і обслуговування, а також процеси управління ресурсами, документацією тощо;

3) рівень управління підприємством. Автоматизація, здійснювана на даному рівні, вирішує завдання, пов'язані з аналізом і прогнозуванням. Сприяння ефективній роботі вищого ступеня ієрархії підприємства в фінансово-господарської області - ось головне її призначення.

Завдяки віртуальній (або "екранній") клавіатурі можна, користуючись системою письма рідної мови, легко й правильно вводити текст, незалежно від того, де ви та яким комп'ютером користуєтесь. Нижче наведено деякі переваги віртуальних клавіатур:

- 1) можна вводити текст рідною мовою на клавіатурі з іноземними символами, наприклад, якщо ви подорожуєте за кордоном або живете в іншій країні;
- 2) можна скористатися більш доступним способом введення тексту – кліки на екрані;
- 3) можна швидко та просто переходити між різними наборами символів і/або алфавітами.

Для підвищення рівня безпеки та захисту від різного шкідливого програмного забезпечення (наприклад, від клавіатурних шпигунів):

Безпечні екранні клавіатури стануть добрими помічниками в справі підвищення рівня безпеки при використанні громадських, незахищених або підозрілих комп'ютерів (наприклад, навчальні класи, інтернет-кафе і т.д.).

Безпечні віртуальні клавіатури можуть допомогти захистити ваш комп'ютер від шкідливого програмного забезпечення наступних типів:

- 1) кейлогери (клавіатурні шпигуни) - фіксують всі натискання клавіш;
- 2) логування екрану - створюють скріншоти екрану з регулярними періодами або з яких-небудь певних дій користувача;
- 3) логування буфера обміну - контролюють вміст буфера обміну;
- 4) фіксація положення курсору миші - зберігають всі координати, де були зафіксовані кліки мишкою. в основному, цей тип використовується для злову банківських екранних клавіатур;
- 5) захоплення значень в текстових полях - отримують все значення з текстових полів, навіть прихованих маскою пароля (всім знайомі ****).

Доброю новиною можна назвати те, що є ряд спеціальних безкоштовних програм, які допоможуть вам безпечно вводити текст. Проте, слід зазначити, що немає однієї унікальної програми, яка включала б у себе всі можливі функції.

Віртуальні клавіатури, які вирішують питання доступності, не пропонують ніякої реальної захисту від шкідливого програмного забезпечення. Так і безпечні екранні клавіатури не містять будь-яких особливих наборів функцій для зручності роботи. Так само необхідно підкреслити, що безпечні екранні клавіатури слід розглядати як додатковий засіб для забезпечення безпеки, які ніколи не зможе замінити спеціалізовані програми захисту, такі як фаєрволи, антивіруси і т.д.

1.3 Аналіз засобів реалізації та обґрунтування вибору мови програмування

Оскільки існує багато можливостей для створення програмного забезпечення, в тому числі й мова програмування, за допомогою якої можна найбільш точно втілити проект в реальність.

Для створення програми було обрано мову програмування C#. Такий вибір був зумовлений наступними перевагами для створення програмного засобу:

1) C# - це об'єктно-орієнтована, проста і в той же час потужна мова програмування, яка дозволяє розробникам створювати багатofункціональні програми;

2) C# ставитися до мов компілюемого типу, тому він має всі переваги таких мов;

3) C# об'єднує кращі ідеї сучасних мов програмування Java, C, Visual Basic і т.д;

4) Із-за великого розмаїття синтаксичних конструкцій і можливості працювати з платформою .Net, C# дозволяє швидше, ніж будь-яка інша мова, розробляти програмні рішення;

5) C# відрізняється надійністю і елегантністю;

6) елегантністю C# відрізняється через велике розмаїття синтаксичних конструкцій;

7) велика надійність була досягнута через роботу CLR машини, адже на відміну від інших компілятор CLR запускає розроблений додаток на віртуальному процесорі. Тому у випадках виникнення будь-яких помилок, це ніяк не вплине на

роботу інших програм в системі, але це також означає, що для запуску програми необхідно додатковий час. Відповідно програми написані на мові програмування C# надійніші, але менш швидкі (ніж ті ж програми, написані на C).

Для здійснення даного вибору було порівняно C# и Java.

Обидві мови, C# і Java, – це вдосконалена мова C ++.

Вони були розроблені в різний час, в основному, в конкуруючій середовищі і мають свої подібності та відмінності.

Java був створений під ім'ям Oak Java в 1990 році американською компанією Sun Microsystems. У 1995 році була представлена перша бета-версія цієї мови.

C# з'явився в 2000 році. Незабаром, в 2002, вийшла перша версія фреймворку .Net, який підтримувався цією мовою програмування.

Java був розроблений на основі Objective-C і C, а базою мови C # виступили C ++ і Java. Однак, незважаючи на свою назву, C # ближче до Java, ніж до C ++.

З точки зору розробника, Java і C # мають майже ідентичну структуру. Обидві мови строго типізовані і об'єктно-орієнтовані. Багато функцій запозичені з синтаксису C ++, проте ці мови простіше для вивчення початківцям.

Java і C # можна порівняти за такими характеристиками:

- 1) синтаксична основа;
- 2) механізм роботи з динамічними даними;
- 3) об'єктні кошти;
- 4) типи даних;
- 5) корисні функції.

Обидві мови мають схожу модель роботи з динамічними даними: об'єкти створюються динамічно за допомогою конструкції "new". Середовище виконання відстежує наявність посилань на них. Збирачі сміття періодично очищають пам'ять об'єктів, у яких немає посилань.

У Java і C# є сильні і слабкі посилання на об'єкти. Обидві мови підтримують метод фіналізатор. Через невизначеність моменту видалення об'єкту, фіналізатор не можуть використовуватися для звільнення системних ресурсів, займаних

об'єктом. Це змушує створювати додаткові методи для об'єкта "cleaning" і викликати їх явно.

У C# для об'єктів, які повинні бути видимими в діапазоні збірки (приблизний аналог пакету Java), був введений окремий модифікатор "internal" (аналог «default» в Java). Модифікатор «protected» зберігає свій первісний зміст від C ++ - доступ тільки від класів-нащадків.

Також можливе комбінування модифікаторів "internal" і "protected".

Обидві мови дозволяють визначати клас в класі.

В Java внутрішні класи використовуються для емуляції замикань. Вони мають доступ до нестатичних елементів батьківського класу. Також, в межах методу, можна визначити локальні класи, які мають доступ на читання до локальних змінних; безіменні (анонімні) локальні класи, які фактично дозволяють створювати екземпляри об'єктів і інтерфейсів, а також методи перекриття класу безпосередньо в місці їх використання.

У C# є замикання і лямбда. Підхід C# більше нагадує C ++: внутрішні класи доступні тільки для статичних елементів зовнішнього класу. Для доступу в нестатичні елементи необхідно явно вказати екземпляр зовнішнього класу. У C# не підтримуються локальні внутрішні класи.

Лямбда-вирази з'являються у версії Java 8.

В обох мовах методи визначаються функціями класу. Тіло методу локалізовано в описі класу. Підтримуються статичні і абстрактні методи.

У C# існує очевидна реалізація методів інтерфейсу, яка дозволяє будь-якому класу реалізовувати їх окремо від своїх власних методів або давати різні реалізації однойменних методів, що належать двом різним інтерфейсів.

В Java 8 з'являється оператор "default", який дозволяє визначити реалізацію методів інтерфейсу "за замовчуванням". Таким чином, клас інтерфейсу позбавляється від зобов'язання реалізації методів за замовчуванням, але може замінити їх.

C# повторює концепцію віртуальних методів C ++: віртуальні методи можуть бути оголошені з ключовим словом "virtual", інші методи не є віртуальними. Більш

того, C# вимагає явного оголошення перекриття віртуального методу в похідному класі за допомогою ключового слова "override».

В Java навпаки, всі відкриті методи, крім статичних, є віртуальними і неможливо перевизначити метод без участі віртуального механізму. Цей підхід синтаксично простіше, так як він забезпечує виклик методу класу, до якого належить об'єкт.

Переваги C# над Java:

1) лямбда-функції. Також, як і анонімні внутрішні класи в Java, C# забезпечує більш ефективні лямбда-функції. Це зручний метод для визначення об'єкта анонімної функції безпосередньо в точці виклику або передачі функції в якості аргументу. Як правило, лямбда-вирази використовуються для інкапсуляції переданих в алгоритми або асинхронних методів декількох рядків коду;

2) делегати використовуються в якості методів для вказівки на інші методи. В Java теж є така функція, але в C# вона простіше;

3) мова C# дозволяє виконувати перевантаження операторів для їх використання у власних класах. Це дозволяє досягти природного вигляду визначається користувачем типу даних і використовувати як основний тип даних;

4) за допомогою "yield" можна визначити генератор. Використовуючи генератор, можна створювати обчислення послідовностей «на льоту», що допомагає заощадити пам'ять і виконувати обчислення нескінченних послідовностей;

5) це зручний спосіб розширення існуючих класів, навіть якщо вони готові без реального розширення;

6) він пропонує простий синтаксис для отримання значення посилального типу;

7) ґрунтуючись на перерахованих вище функціях, можна переконатися, що C# – просунута мова програмування, яка була створена для вирішення різних завдань високої складності.

2 Спеціальна частина

2.1 Опис алгоритму створення програмного засобу

При розробці програмного засобу віртуалізації процесів, які виконуються мишкою був створений алгоритм програмного засобу. Дотримання кроків алгоритму є обов'язковим при розробці програмного засобу.

На рис. 2.1 зображена блок-схема програми «Віртуальна клавіатура».



Рисунок 2.1 – Блок-схема алгоритму програми «Віртуальна клавіатура»

Алгоритм описує процес роботи програмного засобу, послідовність виконання функцій програми.

Відповідно до алгоритму, зображеного на рис. 2.1, основні процеси програмного засобу наступні: вивід головного вікна, вивід елементів клавіатури, вибір клавіатури, ввід тексту, можна завершити ввід або перейти до вибору елементів клавіатури, вивести текст та завершити роботу з програмою.

2.2 Функціональні та нефункціональні вимоги

Функціональні вимоги пояснюють, що повинно бути зроблено. Вони ідентифікують завдання або дії, які повинні бути виконані. Функціональні вимоги визначають дії, які система повинна бути здатною виконати, зв'язок входу / виходу в поведінці системи. Що стосується не функціональних вимог - це вимоги, які визначають критерії роботи системи в цілому, а не окремі сценарії поведінки. Нефункціональні вимоги визначають системні властивості такі як продуктивність, зручність супроводу, розширюваність, надійність.

Вимогами до програмного продукту являються властивості та якості програмного продукту, сукупність вимог щодо атрибутів. Специфікація функціональних вимог описує внутрішню роботу всієї системи та її поведінку під час використання. Нефункціональні вимоги відображають критерії роботи в цілому.

Функціональні вимоги регламентують функціонування або поведіння системи (behavioral requirements). Функціональні вимоги відповідають на запитання «що повинна робити система» у тих або інших ситуаціях. Функціональні вимоги визначають основний «фронт робіт» розроблювача, і встановлюють мети, задачі й сервіси, надавані системою замовникові.

Функціональні вимоги записуються, як правило, за допомогою правил. Іншим способом є так звані варіанти використання (uses cases) – популярний і досить продуктивний спосіб подання вимог.

Це – основний, визначальний вид вимог, що буде розглядатися протягом усього лекційного курсу.

Нефункціональні вимоги, відповідно, регламентують внутрішні й зовнішні умови або атрибути функціонування системи. К. Вігерс виділяє наступні основні групи нефункціональних вимог:

- 1) зовнішні інтерфейси (External Interfaces);
- 2) атрибути якості (Quality Attributes);
- 3) обмеження (Constraints).

Серед зовнішніх інтерфейсів у більшості сучасних АІС найбільш важливим є інтерфейс користувача (User Interface, UI). Крім того, виділяються інтерфейси із зовнішніми пристроями (апаратні інтерфейси), програмні інтерфейси й інтерфейси передачі інформації (комунікаційні інтерфейси).

Основні атрибути якості:

- 1) застосовність;
- 2) надійність;
- 3) продуктивність;
- 4) експлуатаційна придатність, досить добре;
- 5) розкриті в моделі FURPS.

Функціональні вимоги програмного засобу «Віртуальна клавіатура» наступні:

- 1) користувач повинен мати можливість запису/перегляду тексту;
- 2) можливість налаштування клавіатури під користувача;
- 3) можливість використання або текстової, або цифрової клавіатури.

До нефункціональних вимог програмного додатку відносять атрибути якості та зовнішній інтерфейс.

Атрибутами якості є наступні характеристики програмного продукту:

- 1) легкість;
- 2) простота використання;
- 3) простота переміщення;
- 4) цілісність;

- 5) ефективність;
- 6) стійкість до збоїв.

2.3 Структура програмного модуля

Життєвий цикл (ЖЦ) містить у собі процес керування і контролю, який базується на сітковому графіку, що враховує роботи з розробки системи, витрати і строки. Спостереження і контроль виконання плану проводить організаційний відділ. У графіку містяться роботи й взаємозв'язки між ними і їхніми виконавцями, а також проектні документи, які розроблюються виконавцями. Результати кожного з процесів ЖЦ контролюються і передаються на наступний етап у вигляді, зручному для подальшої реалізації іншими виконавцями.

П'ять головних модулів підтримки, як процесів ЖЦ з проектування ПЗ:

- 1) вивчення можливості виконання проекту (Feasibility Study Module);
- 2) аналіз вимог (Requirements Analysis Module);
- 3) специфікація вимог (Requirements Specification Module);
- 4) логічна специфікація системи (Logical System Specification Module);
- 5) фізичне проектування (Physical Design Module).

Етап проектування ПЗ передбачає створення наступних діаграм:

- 1) структурна схема ПЗ;
- 2) діаграма класів;
- 3) діаграма варіантів використання;
- 2) діаграма послідовності.

Структурна схема програмного засобу демонструє загальну роботу програмного засобу віртуальної клавіатури (рис 2.2).

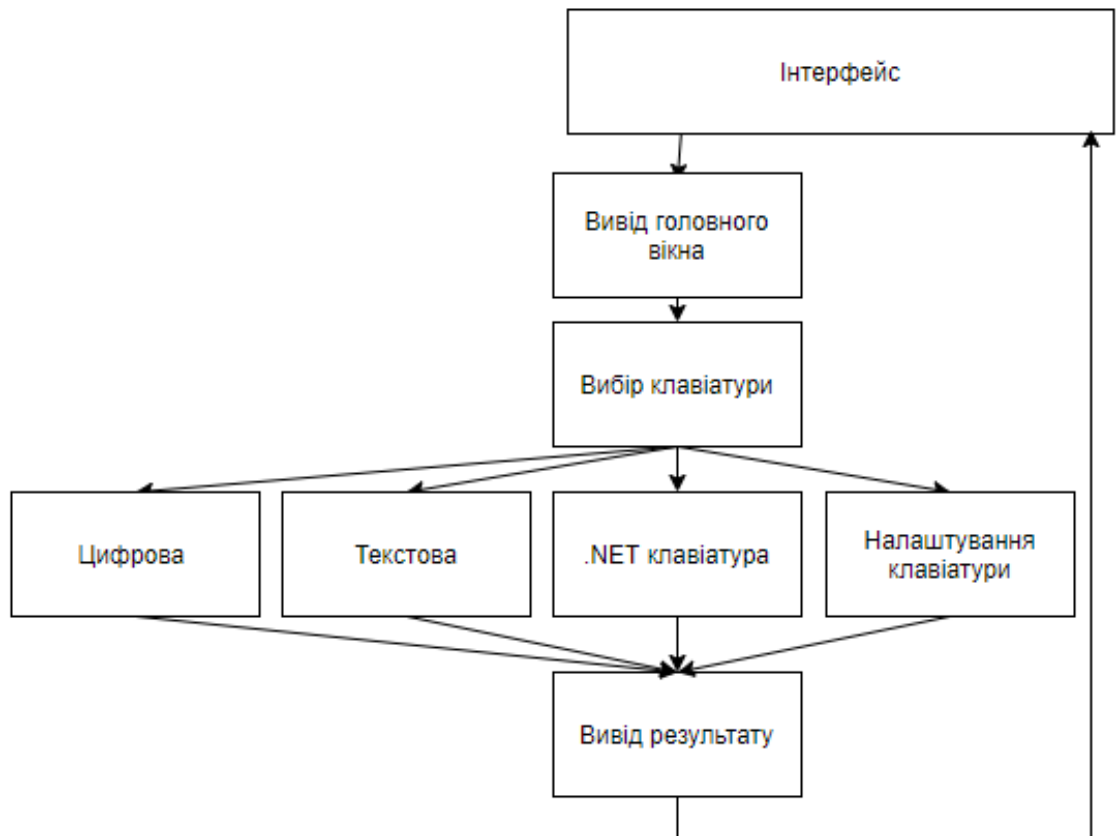


Рисунок 2.2 – Структурна схема програмного засобу

Схема складається з наступних компонентів:

- 1) інтерфейс програми для взаємодії з користувачем;
- 2) вивід головного вікна;
- 3) вибір клавіатури;
- 4) вивід результату.

UML (англ. Unified Modeling Language - уніфікована мова моделювання) - мова графічного опису для об'єктного моделювання в області розробки програмного забезпечення, моделювання бізнес-процесів, системного проектування та відображення організаційних структур.

Одне із завдань UML - служити засобом комунікації всередині команди і при спілкуванні з замовником. Давайте розглянемо можливі варіанти використання діаграм.

Проектування. UML-діаграми допоможуть при моделюванні архітектури великих проектів, в якій можна зібрати як великі, так і більш дрібні деталі і намалювати каркас (схему) додатки. По ньому згодом буде будуватися код.

Реверс-інжиніринг - створення UML-моделі з існуючого коду програми, зворотне побудова. Може застосовуватися, наприклад, на проектах підтримки, де є написаний код, але документація неповна або відсутня.

З моделей можна витягувати текстову інформацію і генерувати щодо легким для читання тексти - документувати. Текст і графіка будуть доповнювати один одного.

Діаграма класів показують набір класів, інтерфейсів, а також їх зв'язку. Діаграми цього виду найчастіше використовуються для моделювання об'єктно-орієнтованих систем. Вони призначені для статичного представлення системи.

Більшість елементів UML мають унікальну і пряму графічну нотацію, яка дає візуальне уявлення найбільш важливих аспектів елемента.

Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

Діаграма класів системи віртуальної клавіатури представлена на рис. 2.3.

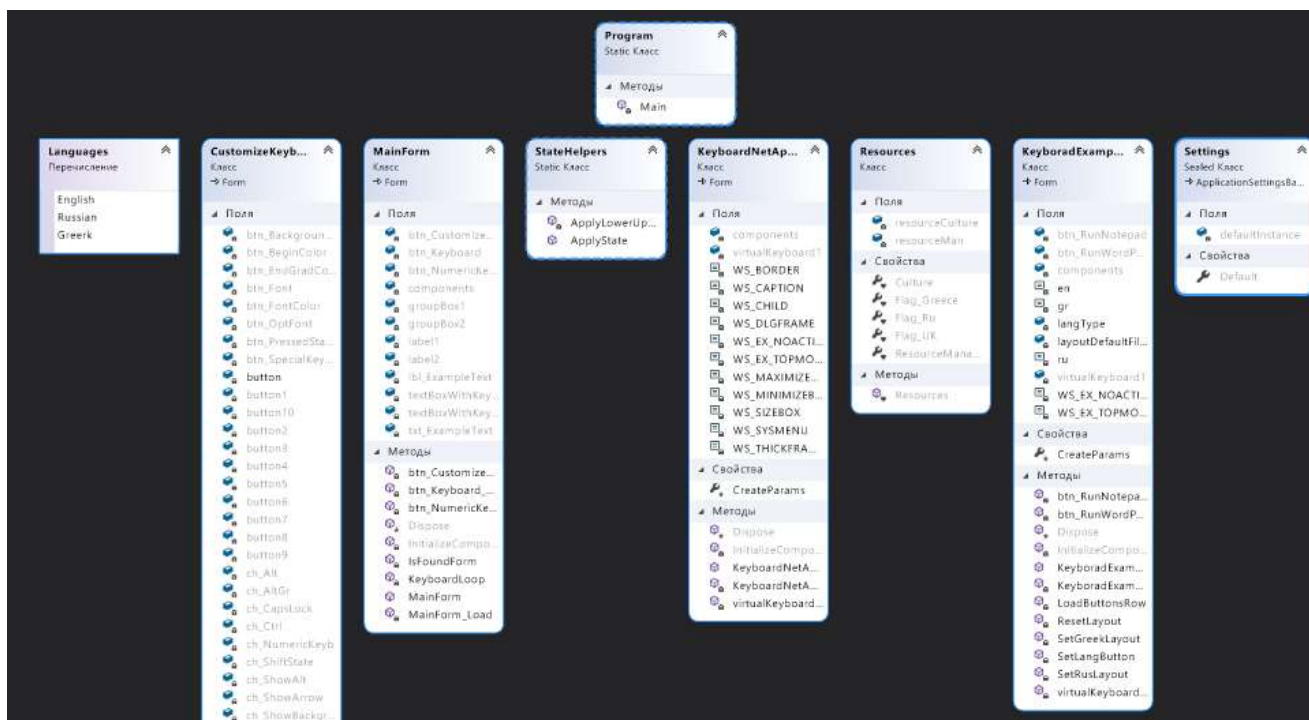


Рисунок 2.3 – Діаграма класів

Діаграма прецедентів використовує 2 основних елементи:

1) Actor (учасник) - безліч логічно пов'язаних ролей, виконуваних при взаємодії з прецедентами або сутностями (система, підсистема або клас).

Учасником може бути людина, роль людини в системі або інша система, підсистема або клас, які представляють щось поза суті.

2) Use case (прецедент) - опис окремого аспекту поведінки системи з точки зору користувача. Прецедент не показує, "як" досягається певний результат, а тільки "що" саме виконується.

Діаграма варіантів використання програмного засобу зображена на рисунку 2.4.

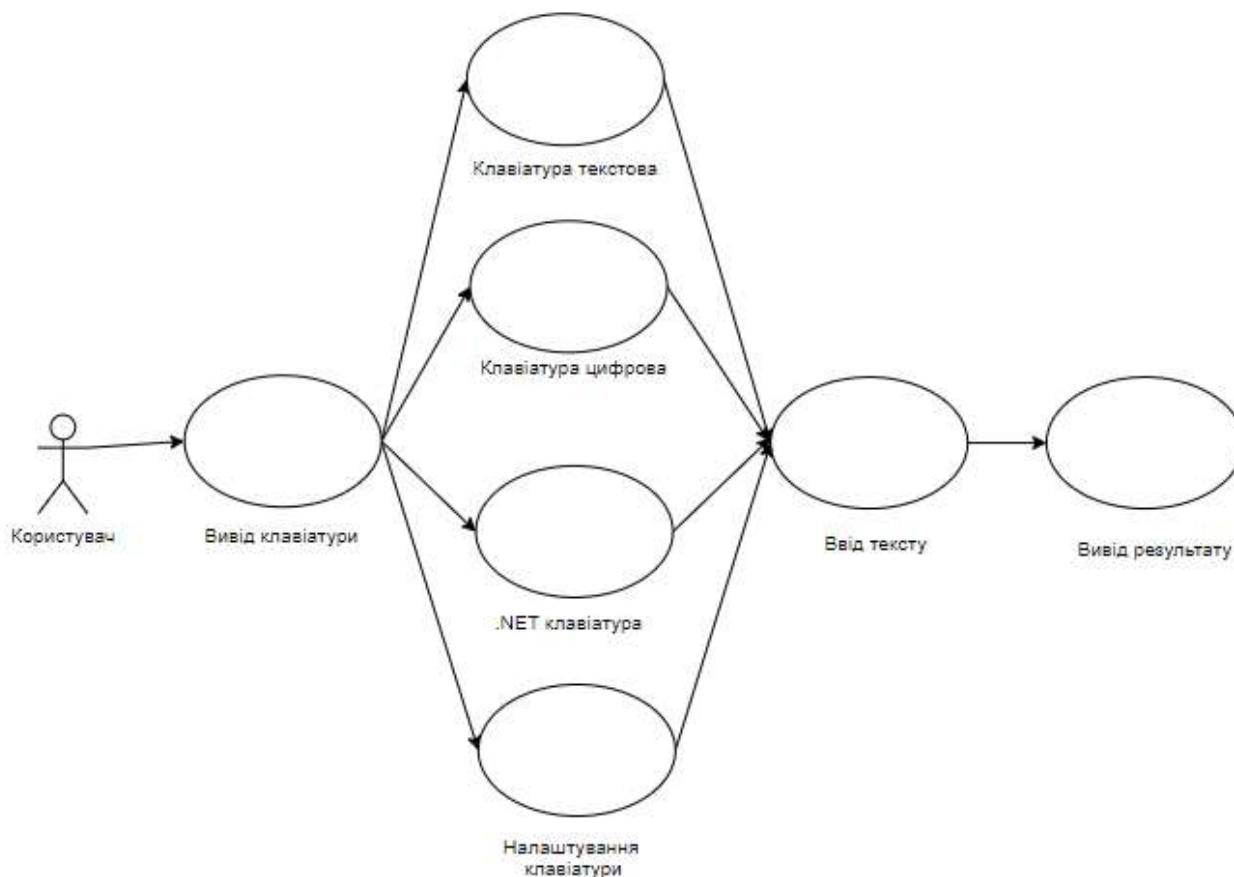


Рисунок 2.4 – Діаграма прецедентів програмного засобу

Використовується для уточнення діаграм прецедентів - описує поведінкові аспекти системи. Діаграма послідовності відображає взаємодію об'єктів в динаміці, в часі. При цьому інформація набуває вигляду повідомлень, а взаємодія об'єктів передбачає обмін цими повідомленнями в рамках сценарію.

В UML взаємодія елементів розглядається в інформаційному аспекті їх комунікації, тобто взаємодіючі об'єкти обмінюються між собою деякою інформацією. При цьому інформація приймає форму закінчених повідомлень.

Іншими словами, хоча повідомлення і має інформаційний зміст, воно набуває додатково властивість надавати направлений вплив на свого одержувача.

Це повністю узгоджується з принципами ООАП, коли будь-які види інформаційної взаємодії між елементами системи повинні бути зведені до відправки і прийому повідомлень між ними. Для моделювання взаємодії об'єктів у мові UML використовуються відповідні діаграми взаємодії. Говорячи про ці діаграми, мають на увазі два аспекти взаємодії.

Діаграма послідовності програмного засобу представлена на рисунку 2.5.

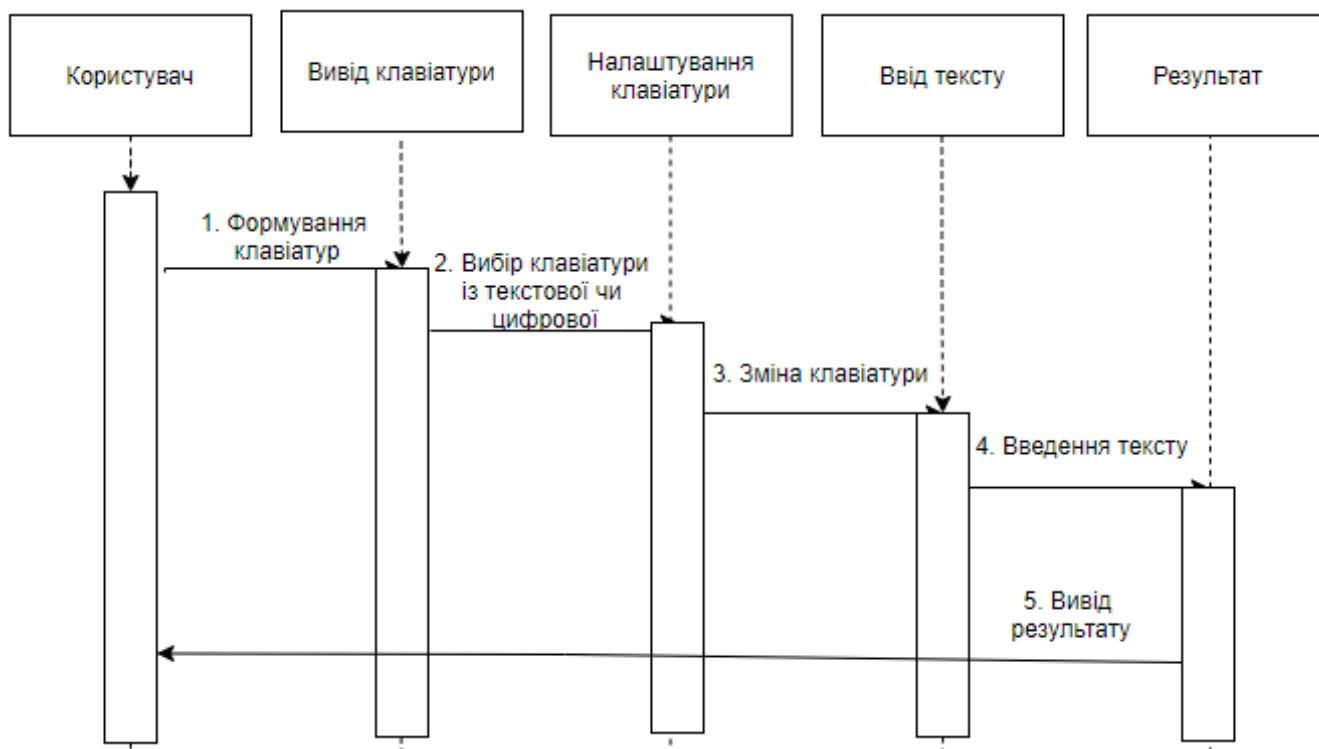


Рисунок 2.5 – Діаграма послідовності програми

2.4 Опис засобів реалізації

Для реалізації системи використовуються такі основні класи, як Button, Label, TextBox, GroupBox, Keyboard.

Button - можна клацнути мишею, ввести клавішу або пробіл, якщо кнопка має фокус.

Встановіть `AcceptButton` властивість або `CancelButton` об'єкта, `Form` щоб дозволити користувачам натиснути кнопку, натиснувши клавіші `Enter` або `ESC`, навіть якщо кнопка не має фокусу. Це дає формі поведінку діалогового вікна.

При відображенні форми за допомогою `ShowDialog` методу можна використовувати `DialogResult` властивість кнопки, щоб вказати значення, що повертається `ShowDialog`.

Ви можете змінити вигляд кнопки. Наприклад, щоб зробити його плоским для веб-пошуку, задайте `FlatStyle` для властивості значення `FlatStyle.Flat`. `FlatStyle`Свойству також можна привласнити значення `FlatStyle.Popup`, яке відображається плоским до тих пір, поки курсор не знайде над кнопкою; потім кнопка приймає стандартний вид кнопки `Windows`.

`Label` елементи управління зазвичай використовуються для надання описового тексту для елемента управління. Наприклад, можна використовувати, `Label` щоб додати описовий текст для `TextBox` елемента управління, щоб інформувати користувача про тип даних, очікуваних в елементі управління. `Label` елементи управління також можна використовувати для додавання тексту опису в, `Form` щоб надати користувачеві корисну інформацію. Наприклад, можна додати `Label` в початок `Form`, що надає користувачеві інструкції щодо введення даних в елементи управління у формі. `Label` елементи управління можна також використовувати для виведення відомостей про час виконання в стані додатки. Наприклад, можна додати в `Label` форму елемент управління для перегляду стану кожного файлу у вигляді списку оброблюваних файлів.

`GroupBox` відображає кадр навколо групи елементів керування з підписом або без нього. Використовуйте `GroupBox`, щоб логічно згрупувати колекцію елементів керування у формі. Групове поле - це елемент керування контейнером, який можна використовувати для визначення груп елементів керування.

Типовим використанням для групового вікна є вміщення логічної групи елементів керування `RadioButton`. Якщо у вас є два групові поля, кожна з яких містить кілька кнопок опцій (також відомих як перемикачі), кожна група кнопок взаємовиключна, встановлюючи одне значення параметра для групи.

Ви можете додати елементи керування до `GroupBox`, використовуючи метод `Add` властивості `Controls`.

`Keyboard` Клас надає події, методи і властивості, пов'язані з клавіатурою, які надають відомості про стан клавіатури.

Кожне з подій, `Keyboard` визначають як вкладене подія, також повторно надається класами базових елементів `UIElement` і `ContentElement` як нове перенаправлення подія. Як правило, більш зручно управляти подіями клавіатури для додатка в `UIElement` і `ContentElement`, а не за допомогою `Keyboard` подій. Додаткові відомості див. У розділі Загальні відомості про вхідних даних.

Щоб елемент міг отримувати введення з клавіатури, елемент повинен бути в фокусі. Більшість `UIElement` похідних об'єктів є фокусом за замовчуванням. В іншому випадку, щоб зробити елемент сфокусованим, встановіть `Focusable` для властивості базового елемента значення `true`

За допомогою `TextBox` елемента управління користувач може вводити текст в додатку. Цей елемент управління має додаткові функціональні можливості, які не знайдені в стандартному елементі управління "текстове поле `Windows`", включаючи многострочное редагування і маскування символів пароля.

Як правило, `TextBox` елемент управління використовується для виведення або прийому в якості вхідних даних, одного рядка тексту. Можна використовувати `Multiline` Властивості і, `ScrollBars` щоб дозволити відображення або введення декількох рядків тексту. Задайте для `AcceptsTab` властивостей і значення, `AcceptsReturn true` щоб забезпечити обробку більшого тексту в багаторядковому `TextBox` елементі управління.

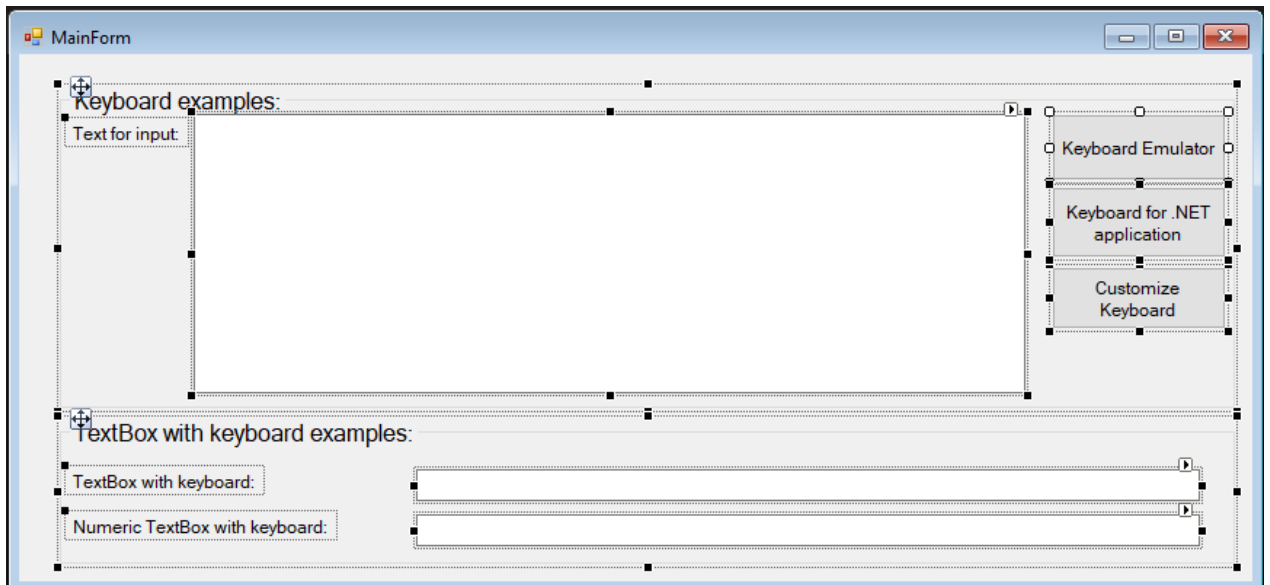


Рисунок 2.6 – Конструктор з основними засобами реалізації

Приклади основних методів, які використовуються для реалізації функцій програмного засобу описані нижче.

Даний метод виконує вибір мови вводу клавіатури.

```
private void virtualKeyboard1_KeyboardButtonPressed(string command, KeyboardButtonEventArgs e)
{
    if (command == KeyboardKeyConstants.LanguageSelection)
    {
        switch (virtualKeyboard1.LanguageButtonTopText)
        {
            case en:
                SetLangButton(ru, gr, Resources.Flag_Ru);
                langType = (int)Languages.Russian;
                SetRusLayout();
                break;
            case ru:
                SetLangButton(gr, en, Resources.Flag_Greece);
                langType = (int)Languages.Greerk;
                SetGreekLayout();
                break;
            case gr:
                SetLangButton(en, ru, Resources.Flag_UK);
                langType = (int)Languages.English;
                ResetLayout();
                break;
        }
    }
}
```

```

    }
else
{
    switch (langType)
    {
        case (int) Languages.English:
            ResetLayout();
            break;
        case (int) Languages.Russian:
            SetRusLayout();
            break;
        case (int)Languages.Greek:
            SetGreekLayout();
            break;
        default:

            break;
    }
}
}
}

```

Наступний описаний метод виконує підвантаження даних клавіатури із файлу.

```

private void SetRusLayout()
{
    var xmlPath = layoutDefaultFilePath + "\\KeyboardLayout_RusLower.XML";

    if (virtualKeyboard1.ShiftState && virtualKeyboard1.CapsLockState)
    {
        xmlPath = layoutDefaultFilePath + "\\KeyboardLayout_RusLower.XML";
    }
    else if (virtualKeyboard1.ShiftState || virtualKeyboard1.CapsLockState)
    {
        xmlPath = layoutDefaultFilePath + "\\KeyboardLayout_Rus.XML";
    }
    else
    {
        xmlPath = layoutDefaultFilePath + "\\KeyboardLayout_RusLower.XML";
    }

    XDocument xmlDocument = XDocument.Load(xmlPath);
    var firstRow = LoadButtonsRow(xmlDocument, 1);
    var secondRow = LoadButtonsRow(xmlDocument, 2);
    var thirdRow = LoadButtonsRow(xmlDocument, 3);
    var fourthRow = LoadButtonsRow(xmlDocument, 4);
    var fifthRow = LoadButtonsRow(xmlDocument, 5);

    virtualKeyboard1.FirstRowCustomButtons = firstRow;
    virtualKeyboard1.SecondRowCustomButtons = secondRow;
    virtualKeyboard1.ThirdRowCustomButtons = thirdRow;
    virtualKeyboard1.FourthRowCustomButtons = fourthRow;
    virtualKeyboard1.FifthRowCustomButtons = fifthRow;
}

```

}

2.5 Порівняльний аналіз реалізованого програмного засобу та програм-аналогів

Віртуальні клавіатури, які вирішують питання доступності, не пропонують ніякої реальної захисту від шкідливого програмного забезпечення. Так і безпечні екранні клавіатури не містять будь-яких особливих наборів функцій для зручності роботи. Так само необхідно підкреслити, що безпечні екранні клавіатури слід розглядати як додатковий засіб для забезпечення безпеки, які ніколи не зможе замінити спеціалізовані програми захисту, такі як фаєрволи, антивіруси і т.д.

Приклади деяких програм-віртуальних клавіатур: Click-N-Type, Microsoft On-Screen Keyboard, Free Virtual Keyboard.

Click-N-Type це відмінна безкоштовна екранна клавіатура, яка дасть фору своїм комерційним аналогам. Програма має безліч функцій і різних налаштувань. Але, на жаль, зовні інтерфейс виглядає трохи застарий. Правда, для користувачів, яким важливіше функціональне наповнення, даний недолік не здається серйозним.

Перш ніж почати, користувачеві необхідно встановити курсор в бажане місце введення тексту. Як аналог, програма пропонує використовувати власні кнопки стрілки для навігації в межах блоку тексту. Так само програма підтримує табуляцію для здійснення переміщення між різними полями введення.

Є цілий ряд різних розкладок на вибір. Якщо вам необхідні клавіші зі стрілками, то не забудьте вибрати необхідну розкладку. Все що необхідно, це відкрити меню File і вибрати. Звичайна розкладка нагадує класичну вкорочену qwerty клавіатуру. Такий вибір цілком зрозумілий, так як до сих пір класична клавіатура являє собою якусь норму того, як повинна виглядати клавіатура.

Програма підтримує велику кількість мов і відповідних розкладок. Ви навіть можете завантажити безкоштовну утиліту з їх сайту, яка допоможе налаштувати вам свій власний набір. Click-N-Type коректно відображає символи кожного з мов. При цьому програма ніяк не залежить від локалізаційних налаштувань Windows.

Тому вам нічого не завадить писати на вашій рідній мові, навіть якщо доводиться набирати текст на чужому комп'ютері (рис. 2.7).

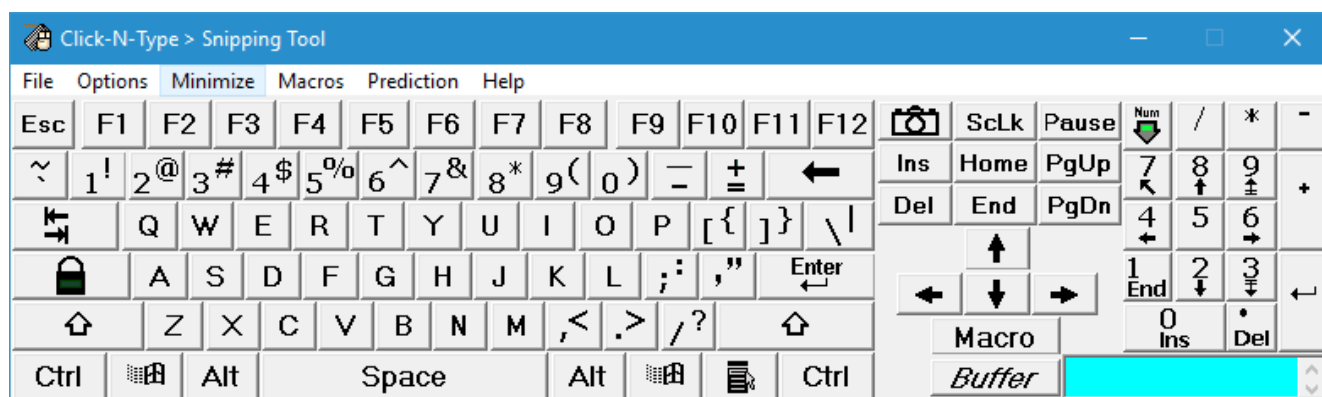


Рисунок 2.7 – Програмний продукт Click-N-Type

Microsoft On-Screen Keyboard (MSOSK) входить в стандартний пакет операційної системи Windows. Знайти її можна за наступним шляхом Пуск / Всі програми / Стандартні / Спеціальні можливості / Екранна клавіатура. Або ж по комбінації Windows + U. Вона має дуже простий і зрозумілий інтерфейс. Включає дві розкладки: розширену і типову, в якій немає числовий колодки і клавіш управління навігацією.

Підтримує три варіанти введення: за переходами на дисплеї, по наведенню курсора і методом сканування. При установці введення по наведенню курсора, спробуйте навести курсор на потрібний символ. І ви побачите як на кнопці буде відобразитися спеціальний індикатор, який покаже коли символ буде вважатися натиснутим. Програма має шість дискретних варіантів затримки від 0,5 до 3,0 секунд. При установці методу сканування, вам буде потрібно всього одна кнопка. Спочатку програма буде послідовно підкреслювати кожен рядок на клавіатурі. Потім, коли ви вибрали необхідну, в рядку будуть послідовно підсвічуватися всі клавіші. Вам залишиться лише тільки вибрати необхідну.

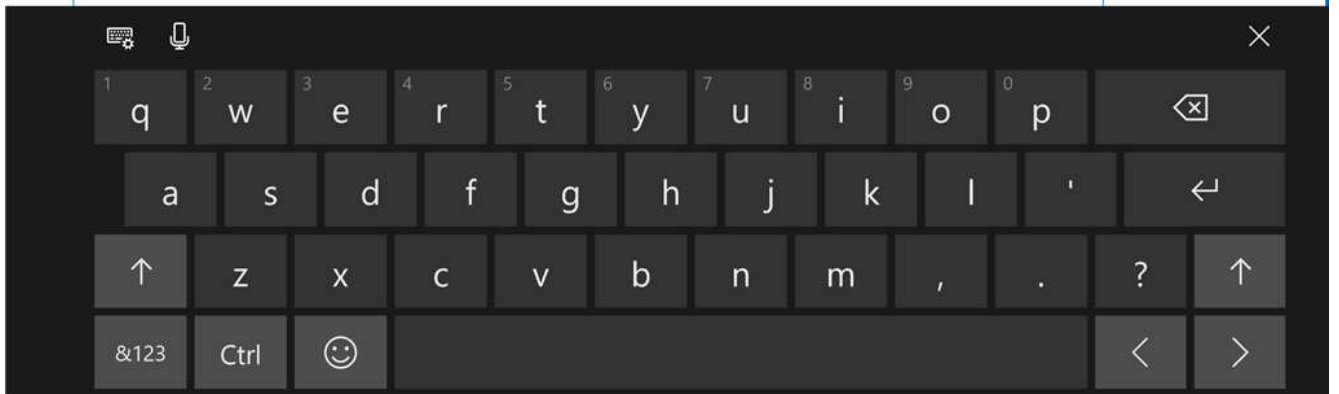


Рисунок 2.8 – Програмний засіб Microsoft On-Screen Keyboard

Free Virtual Keyboard (FVK) ще одна екранна клавіатура. Досить поглянути на її інтерфейс, і відразу стає зрозумілим, що в більшій мірі вона розроблена для людей, що використовують пристрої з сенсорними екранами.

На головному інтерфейсі ви побачите стандартну розкладку клавіатури (без клавіш стрілок і блоку цифр). При цьому добре опрацьований механізм зміни розміру вікна. Розміри шрифтів і розташування кнопок буде змінюватися плавно і без будь-яких стрибків.

Відмінною особливістю можна назвати наявність повзунка, який дозволяє користувачам змінювати прозорість клавіатури з майже непрозорого до майже прозорого. Насправді, це те, чого зазвичай не вистачає на маленьких екранах. І клавіатуру можна розтягнути на весь екран, і бачити написаний текст. Але, тут треба бути акуратним, так як якщо ви зробите клавіатуру занадто прозорою, то буде трохи важко знайти ледве помітний повзунок, щоб повернути все назад. (рис. 2.9).



Рисунок 2.9 – Програмний засіб Free Virtual Keyboard

«Віртуальна клавіатура» – програмний засіб, що виконує функції клавіатури за допомогою інших пристроїв вводу та екранізації. Програмне забезпечення дозволяє здійснювати вибір клавіатур із наявних, а також виконувати налаштування клавіатури під запити користувача. До додаткових функцій відноситься відкриття інших програмних засобів через даний продукт. Програма має зручний, простий та зрозумілий користувацький інтерфейс (рис. 2.10).

Програмний засіб може використовуватися для навчання, у професійній діяльності різних напрямків, а також усіма зацікавленими особами для саморозвитку. Програмний засіб доступний для завантаження для усіх користувачів, є абсолютно безкоштовним та простим у використанні.

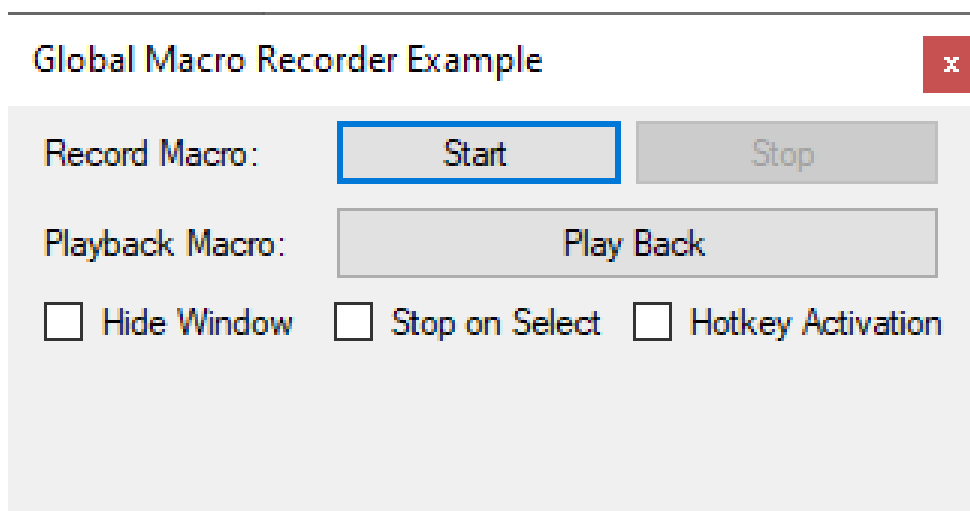


Рисунок 2.10 – Програмний засіб «Віртуальна клавіатура»

Після огляду та порівняння різних програмних засобів для виконання дій клавіатури можна стверджувати, що найбільш простим та зручним у використанні є програмний засіб «Віртуальна клавіатура».

Даний програмний засіб дозволяє використовувати вбудовані та додаткові функції та обирати потрібні.

Програмний засіб може бути застосований у будь-якій галузі для полегшення роботи із рутинною роботою.

«Віртуальна клавіатура» є абсолютно безкоштовним та знаходиться у вільному доступі в мережі Інтернет.

2.6 Керівництво користувача програмного модуля

Програмний додаток віртуалізації клавіатури запускається подвійним кліком миші на файл VirtualKeyboard.exe (рисунок 2.11).

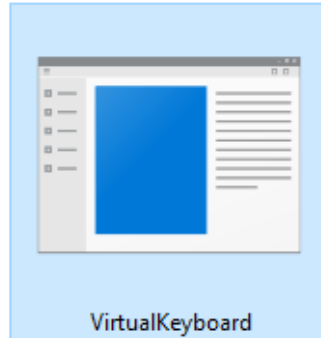


Рисунок 2.11 – Файл запуску програмного засобу

Після чого виконується відкриття головного вікна програми (рисунок 2.12).

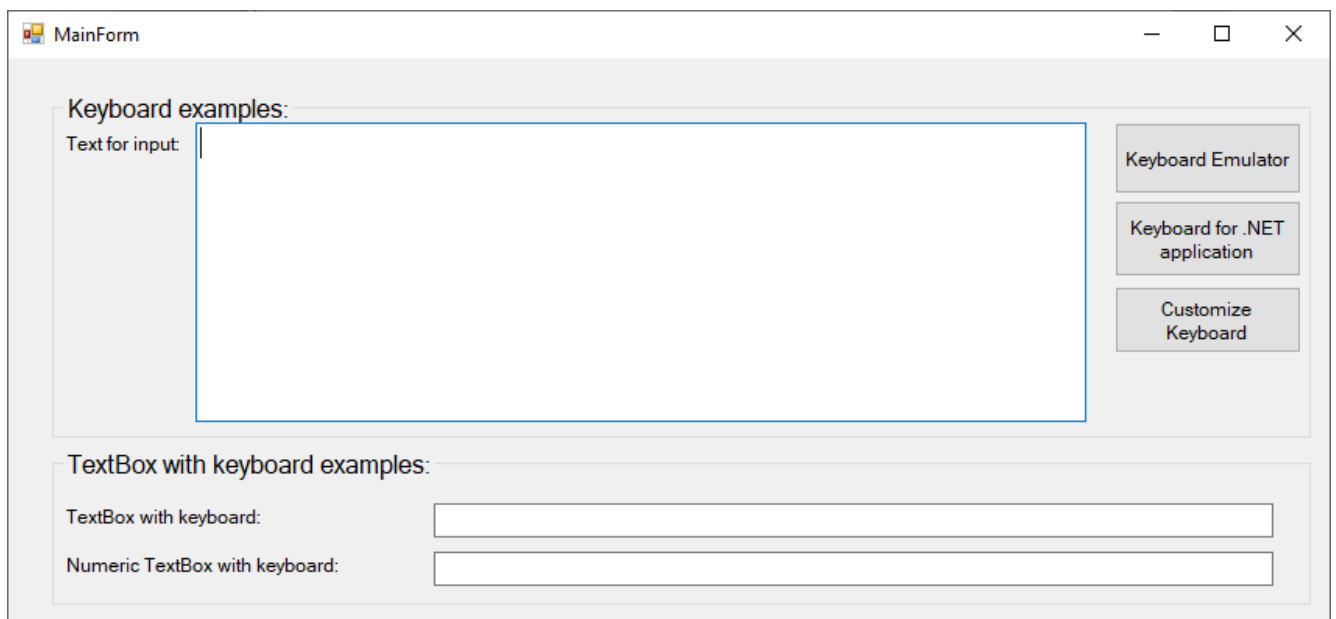


Рисунок 2.12 – Вікно програми

Вікно програми містить поле для прикладу вводу тексту за допомогою різних типів клавіатур (рисунок 2.13).

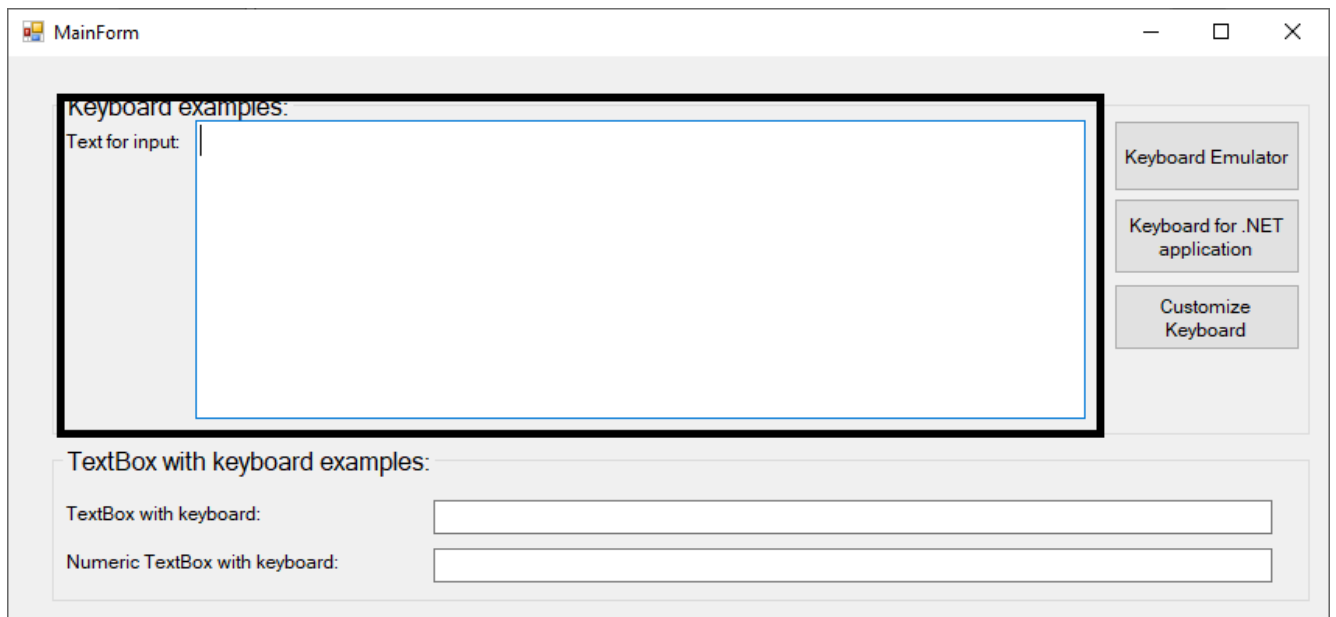


Рисунок 2.13 – Поле вводу тексту

У правій частині вікна можна обрати відповідні кнопки, щоб відкрити одну із клавіатур: симулятор клавіатури, клавіатура для .NET, користувацька клавіатура. (рис. 2.14).

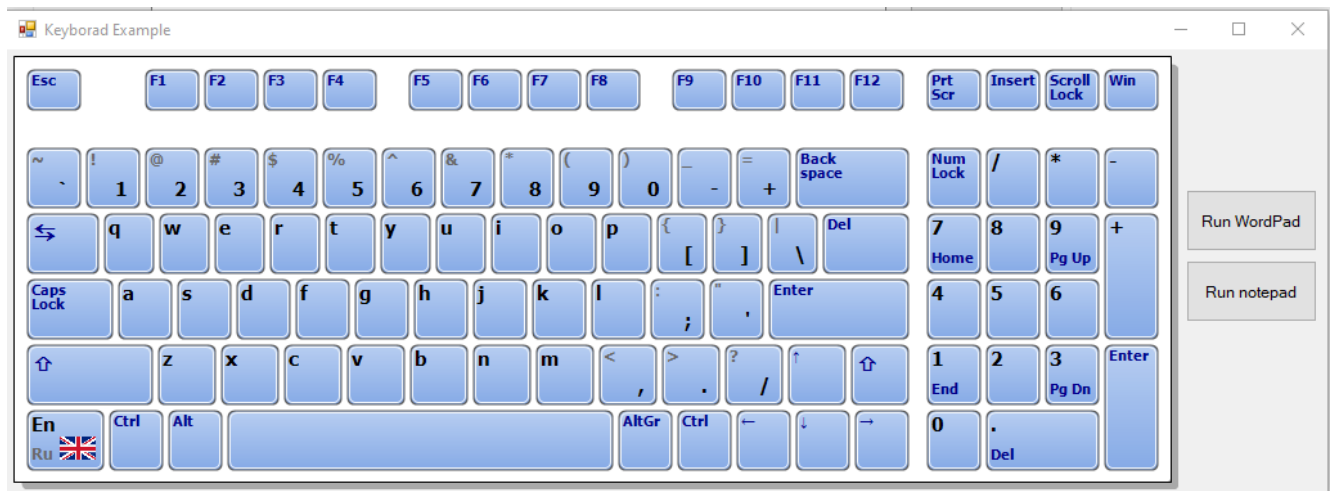


Рисунок 2.14 – Симулятор клавіатури

Наступна кнопка показує клавіатуру для .NET (рис. 2.15).



Рисунок 2.15 – Клавіатура для .NET

Остання кнопка виконує відкриття клавіатури із налаштуваннями користувача. Можна налаштувати: основні властивості, кольори, шрифти, функції кнопок (рис. 2.16).



Рисунок 2.16 – Користувацька клавіатура

Також головне вікно програми містить приклад клавіатури для вводу тексту, яка з'являється при активованому текстовому полі. (рисунок 2.17).



Рисунок 2.17 – Текстова клавіатура

Також є приклад аналогічної попередній клавіатури, але із цифрами (рисунок 2.18).

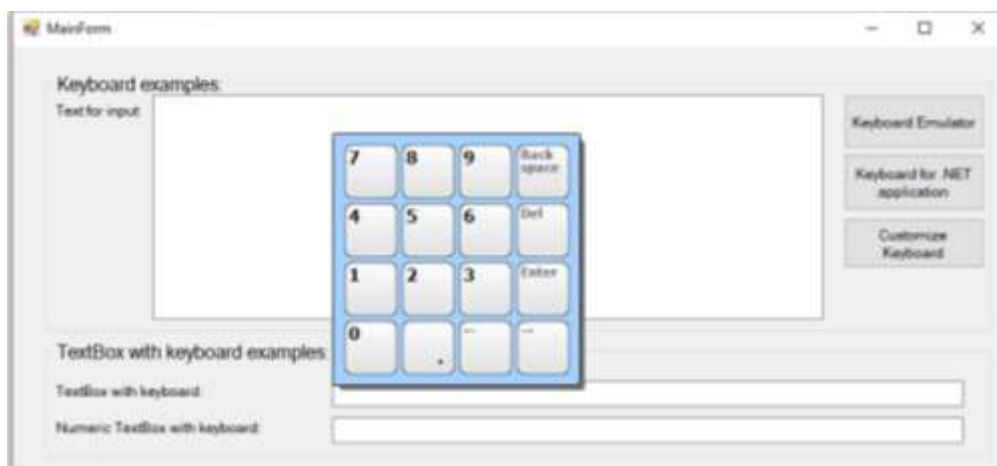


Рисунок 2.18 – Цифрова клавіатура

На екрані симулятора клавіатури знаходиться кнопка запуску Word Pad та NotePad (рисунок 2.19).



Рисунок 2.19 – Запуск інших програм

Після виконання необхідних функцій потрібно коректно завершити роботу програмного засобу, натиснувши на хрестик в правому верхньому куті вікна.

2.7 Тестування роботи програмного модуля

Мета: визначення якості програмного продукту по відношенню до поставленої задачі.

Рівень плану: Тест План (*Test Plan*).

Склад документа: опис обсягу робіт з тестування, опис об'єкта тестування, стратегії, розкладу, критеріїв початку і закінчення тестування.

План проекту: тестування програмного засобу.

Опис тестованого продукту: програмний модуль віртуальної клавіатури, що виконує функції звичайної клавіатури за допомогою засобів вводу та екранізації. Засіб розроблений в середовищі Microsoft Visual Studio за допомогою об'єктно-орієнтованої мови C#.

Відповідний стандарт: *IEEE 829-1998 Format*.

Тестові завдання. Контроль якості програмного засобу та виявлення дефектів.

Тестування повинно бути проведено в таких частинах:

- 1) програмний засіб;
- 2) ввід/редагування тексту;
- 3) вибір клавіатури;
- 4) налаштування клавіатури;
- 5) коректність функціонування інтерфейсу.

Функціональне тестування фокусується на бізнес-вимогах програмного продукту. Основне призначення — переконатися, що весь функціонал ПЗ працює саме так, як зазначено в технічному завданні. Коли ми проводимо функціональне тестування, нас цікавить лише результат роботи системи, а не її проміжні стани. Якщо ви колись чули про техніку чорного ящика (де вас цікавить не внутрішня реалізація, а лише отриманий результат), то це якраз про функціональне тестування.

Нефункціональне тестування оцінює інші аспекти системи: перевірку продуктивності, зручності, надійності тощо.

Обидва типи тестування однаково важливі, адже вони допомагають краще зрозуміти різні особливості системи. Найважливіша мета таких тестувань — забезпечити кінцевих користувачів якісним програмним продуктом.

Функціональні вимоги включають в себе:

- 1) функціональна придатність;
- 2) точність;
- 3) можливість до взаємодії;
- 4) відповідність стандартам та правилам;
- 5) захищеність.

Нефункціональне тестування описує тести, необхідні для визначення характеристик ПЗ, які можуть бути виміряні різними величинами. В цілому, це тестування того, «як» система працює. Далі перелічені основні види нефункціональних тестів:

- 1) всі види тестування продуктивності;
- 2) об'ємне тестування;
- 3) інсталяційне тестування;
- 4) тестування зручності користування;
- 5) тестування на «відмову» та відновлення;
- 6) конфігураційне тестування.

Підхід до тестування. Рівень тестування: системне, тестується інтегрована система на її відповідність вимогам з точки зору кінцевого користувача.

Спеціальні засоби тестування: відсутні, тестування буде виконуватися вручну.

Метрики: в рамках даного плану передбачається створити комплект тестів, повний щодо метрики за тестовими випадками (*Test Cases*) (відповідно до міжнародного стандарту *ISO 14598*).

Особливі вимоги до тестування: відсутні, звичайний режим тестування.

Сегмент компонентів: певний сегмент компонентів повинен бути протестований разом.

Обмеження для тестування: проведення тестування без спеціалізованих засобів.

Рекомендована методика тестування: ручна, тому що зменшує матеріальні та програмні витрати на проведення тестування.

Дуже важливим аспектом тестування є рішення про те, в якому обсязі тестування достатньо і коли необхідно завершити процес тестування. Ретельні вимірювання, такі як досягнуте покриття коду тестами або охоплення функціональності, безумовно, дуже корисні. Однак, самі по собі вони не можуть визначити критеріїв достатності тестування. Прийняття рішення про закінчення тестування також включає розгляд вартості і ризиків, пов'язаних з потенційними збоями і порушеннями надійності функціонування тестованої програмної системи. У той же час, вартість самого тестування також є одним з обмежень, на основі яких приймається рішення про продовження тих чи інших пов'язаних з проектом робіт (з Зокрема, тестування) або про їх припинення.

Критерії успішності та припинення тестування. Критерії успішності тестування:

1) Розроблені тести виконуються без помилок, система передається в експлуатацію;

2) Вдале тестування на виправлену помилку при повторній передачі на тестування.

Критерій виходу дозволяє встановити, який обсяг тестування слід вважати достатнім.

Тестування може бути завершено коли:

- 1) всі 100% вимог враховані;
- 2) дефекти встановлені / очікуване число дефектів виявлено;
- 3) всі дефекти, що відносяться до класу Show Stopper або Blocker, усунуті, ні у одного з критичних дефектів немає статусу «відкритий»;
- 4) всі дефекти з високим пріоритетом ідентифіковані і виправлені;
- 5) Defect Rate (швидкість дефектоутворення) нижче встановленого допустимого рівня;

б) дуже невелика кількість дефектів середнього рівня критичності «відкриті», їх розбір проведений;

7) число «відкритих» дефектів середнього рівня, які не впливають на користування системою, дуже невелика;

8) всі дефекти з високим рівнем пріоритету закриті і відповідні регресивні сценарії успішно проведені.

Тест кейси зображені на таблицях 2.1, 2.2, 2.3.

Таблиця 2.1 – Тест кейс для запису/відтворення макросу

Дія	Очікуваний результат	Фактичний результат
Передумова		
Відкрити програмний засіб	Програмний засіб відкритий та доступний для використання	Пройдено
Кроки тесту		
Вибір клавіатури	Клавіатура успішно відкрита	Пройдено
Налаштування клавіатури	Клавіатура успішно налаштована	Пройдено
Ввід/редагування тексту	Текст успішно вводиться/редагується	Пройдено
Постумова		
Завершити роботу програмного засобу	Робота програмного засобу завершена	Пройдено

Таблиця 2.2 – Тест кейс для визначення дій засобів вводу

Дія	Очікуваний результат	Фактичний результат
Передумова		
Відкрити програмний засіб	Програмний засіб відкритий та доступний для використання	Пройдено
Кроки тесту		
Вибір текстового поля із текстовою клавіатурою	Текстова клавіатура успішно виведена на екран	Пройдено
Вибір поля із цифровою клавіатурою	Цифрова клавіатура успішно виведена на екран	Пройдено
Постумова		
Завершити роботу програмного засобу	Робота програмного засобу завершена	Пройдено

Таблиця 2.3 – Тест кейс додаткових функцій користувача

Дія	Очікуваний результат	Фактичний результат
Передумова		
Відкрити програмний засіб	Програмний засіб відкритий та доступний для використання	Пройдено
Кроки тесту		
Відкрити Вордпад	Програма успішно відкрита	Пройдено

Відкрити Ноутпад	Програма успішно відкрита	Пройдено
Постумова		
Завершити роботу програмного засобу	Робота програмного засобу завершена	Пройдено

Висновком даного тестового випадку (табл. 2.1) є те, що програмний засіб коректно виконує основні функції: виконує вибір та відкриття потрібної клавіатури, здійснює її налаштування та ввід/редагування тексту.

Висновком даного тестового випадку (табл. 2.2) є те, що визначення дій клавіатури програмного засобу виконуються коректно, а саме: виконується відкриття текстової чи цифрової клавіатури при активації певного поля.

Висновком даного тестового випадку (табл. 2.3) є те, що програмний засіб дозволяє виконувати такі додаткові функції, як: відкриття сторонніх програм через створений програмний засіб.

3 Охорона праці

З розвитком науково-технічного прогресу важливу роль грає можливість безпечного виконання людьми своїх трудових обов'язків. У зв'язку з цим була створена і розвивається наука про охорону праці і життєдіяльності людини.

Безпека життєдіяльності (БЖД) - це комплекс заходів, спрямованих на забезпечення безпеки людини в середовищі проживання, збереження його здоров'я, розробку методів і засобів захисту шляхом зниження впливу шкідливих і небезпечних факторів до допустимих значень, вироблення заходів по обмеженню збитку в ліквідації наслідків надзвичайних ситуацій мирного і воєнного часу.

Охорона здоров'я трудящих, забезпечення безпеки умов праці, ліквідація професійних захворювань і виробничого травматизму складає одну з головних турбот людського суспільства. Звертається увага на необхідність широкого застосування прогресивних форм наукової організації праці, зведення до мінімуму ручної, малокваліфікованої праці, створення обстановки, що виключає професійні захворювання і виробничий травматизм.

Даний розділ дипломного проекту присвячений розгляду наступних питань:

- 1) визначення оптимальних умов праці техника-програміста;
- 2) розрахунок освітленості;
- 3) розрахунок рівня шуму.

3.1 Характеристика умов праці програміста

Науково-технічний прогрес вніс серйозні зміни в умови виробничої діяльності робітників розумової праці. Їх праця стала більш інтенсивним, напруженим, які вимагають значних витрат розумової, емоційної і фізичної енергії. Це зажадало комплексного рішення проблем ергономіки, гігієни і організації праці, регламентації режимів праці та відпочинку.

В даний час комп'ютерна техніка широко застосовується у всіх областях діяльності людини. При роботі з комп'ютером людина піддається дії ряду небезпечних і шкідливих виробничих факторів: електромагнітних полів (діапазон радіочастот: ВЧ, УВЧ і СВЧ), інфрачервоного і іонізуючого випромінювань, шуму і вібрації, статичної електрики і ін.

Робота з комп'ютером характеризується значною розумовою напругою і нервово-емоційним навантаженням операторів, високою напруженістю зорової роботи і достатньо великим навантаженням на м'язи рук при роботі з клавіатурою ЕОМ. Велике значення має раціональна конструкція і розташування елементів робочого місця, що важливо для підтримки оптимальної робочої пози людини-оператора.

У процесі роботи з комп'ютером необхідно дотримувати правильний режим праці та відпочинку. В іншому випадку у персоналу наголошуються значна напруга зорового апарату з появою скарг на незадоволеність роботою, головні болі, дратівливість, порушення сну, втому і хворобливі відчуття в очах, в поясниці, в області шиї і руках.

3.2 Вимоги до виробничих приміщень

Забарвлення і коефіцієнти віддзеркалення. Забарвлення приміщень і меблів повинні сприяти створенню сприятливих умов для зорового сприйняття, гарного настрою.

Джерела світла, такі як світильники і вікна, які дають віддзеркалення від поверхні екрану, значно погіршують точність знаків і тягнуть за собою перешкоди фізіологічного характеру, які можуть виразитися в значній напрузі, особливо при тривалій роботі. Віддзеркалення, включаючи віддзеркалення від вторинних джерел світла, повинне бути зведено до мінімуму. Для захисту від надмірної яскравості вікон можуть бути застосовані штори і екрани.

Освітлення. Правильно спроектоване і виконане виробниче освітлення покращує умови зорової роботи, знижує стомлюваність, сприяє підвищенню продуктивності праці, благотворно впливає на виробниче середовище, надаючи

позитивну психологічну дію на працюючого, підвищує безпеку праці і знижує травматизм.

Недостатність освітлення приводить до напруги зору, ослабляє увагу, приводить до настання передчасної стомленості. Надмірно яскраве освітлення викликає засліплення, роздратування і різь в очах. Неправильний напрямок світла на робочому місці може створювати різкі тіні, відблиски, дезорієнтувати працюючого. Всі ці причини можуть призвести до нещасного випадку або профзахворювань, тому такий важливий правильний розрахунок освітленості.

Існує три види освітлення - природне, штучне і поєднане (природне і штучне разом).

Природне освітлення - освітлення приміщень денним світлом, що потрапляє через світлові прорізи в зовнішніх огорожуючих конструкціях приміщення. Природне освітлення характеризується тим, що змінюється в широких межах залежно від часу дня, пори року, характеру області і ряду інших чинників.

Штучне освітлення застосовується при роботі в темний час доби і вдень, коли не вдається забезпечити нормовані значення коефіцієнта природного освітлення (похмура погода, короткий світловий день). Освітлення, при якому недостатнє за нормами природне освітлення доповнюється штучним, називається змішаним освітленням.

Згідно СНіП II-4-79 в приміщень обчислювальних центрів необхідно застосувати систему комбінованого освітлення.

При виконанні робіт категорії високої зорової точності (найменший розмір об'єкту розрізнення 0,3 ... 0,5 мм) величина коефіцієнта природного освітлення (КЕО) повинна бути не нижче 1,5%, а при зоровій роботі середньої точності (найменший розмір об'єкту розрізнення 0,5 ... 1,0 мм) КЕО повинен бути не нижче 1,0%. В якості джерел штучного освітлення звичайно використовуються люмінесцентні лампи типу ЛБ, або ДРЛ, які попарно об'єднуються в світильники, які повинні розташовуватися рівномірно над робочими поверхнями.

Вимоги до освітленості в приміщеннях, де встановлені комп'ютери, наступні: при виконанні зорових робіт високої точності загальна освітленість повинна

складати 300лк, а комбінована - 750лк; аналогічні вимоги при виконанні робіт середньої точності - 200 і 300лк відповідно.

Параметри мікроклімату. Параметри мікроклімату можуть мінятися в широких межах, у той час як необхідною умовою життєдіяльності людини є підтримка постійності температури тіла завдяки терморегуляції, тобто здатності організму регулювати віддачу тепла в навколишнє середовище. Принцип нормування мікроклімату - створення оптимальних умов для теплообміну тіла людини з навколишнім середовищем.

Обчислювальна техніка є джерелом істотних тепловиділень, що може привести до підвищення температури і зниження відносної вологості в приміщенні. У приміщеннях, де встановлені комп'ютери, повинні дотримуватися певні параметри мікроклімату. У санітарних нормах СН-245-71 встановлені величини параметрів мікроклімату, що створюють комфортні умови. Ці норми встановлюються в залежності від пори року, характеру трудового процесу і характеру виробничого приміщення (табл. 3.1).

Об'єм приміщень, в яких розміщені працівники обчислювальних центрів, не повинен бути меншим $19,5 \text{ м}^3$ / людини з урахуванням максимального числа одночасно працюючих в зміну. Норми подачі свіжого повітря в приміщення, де розташовані комп'ютери, приведені в табл. 3.2.

Таблиця 3.1 - Параметри мікроклімату для приміщень, де встановлені комп'ютери

Період року	Параметр мікроклімату	Величина
Холодний	Температура повітря в приміщенні	22 ... 24 ° С
	Відносна вологість	40 ... 60%
	Швидкість руху повітря	до 0,1 м / с
Теплий	Температура повітря в приміщенні	23 ... 25 ° С
	Відносна вологість	40 ... 60%
	Швидкість руху повітря	0,1 ... 0,2 м / с

Таблиця 3.2 - Норми подачі свіжого повітря в приміщення, де розташовані комп'ютери

Характеристика приміщення	Об'ємна витрата подається в приміщення свіжого повітря, м ³ / на одну людину в годину
Об'єм до 20м ³ на особу	Не менше 30
20 ... 40м ³ на особу	Не менше 20
Більш 40м ³ на особу	Природна вентиляція

Для забезпечення комфортних умов використовуються як організаційні методи (раціональна організація проведення робіт залежно від пори року і доби, чергування праці і відпочинку), так і технічні засоби (вентиляція, кондиціонування повітря, опалювальна система).

Шум і вібрація. Шум погіршує умови праці надаючи шкідливу дію на організм людини. Працюючі в умовах тривалої шумової дії випробовують дратівливість, головні болі, запаморочення, зниження пам'яті, підвищену

стомлюваність, зниження апетиту, біль у вухах і т.д. Такі порушення в роботі ряду органів і систем організму людини можуть викликати негативні зміни в емоційному стані людини аж до стресових. Тривала дія інтенсивного шуму [вище 80 дБ (А)] на слух людини приводить до його часткової або повної втрати.

У табл. 3.3 вказані граничні рівні звуку залежно від категорії тяжкості і напруженості праці, що є безпечними відносно збереження здоров'я і працездатності.

Таблиця 3.3 - Граничні рівні звуку, дБ, на робочих місцях.

Категорія напруженості праці	Категорія важкості праці			
	I. Легка	II. Середня	III. Важка	IV. Дуже важка
I. Мало напружений	80	80	75	75
II. Помірно напружений	70	70	65	65
III. Напружений	60	60	-	-
IV. Дуже напружений	50	50	-	-

Ергономічні вимоги до робочого місця. Проектування робочих місць, забезпечених відеотерміналами, відноситься до числа важливих проблем ергономічного проектування в області обчислювальної техніки.

Робоче місце і взаємне розташування всіх його елементів повинне відповідати антропометричним, фізичним і психологічним вимогам. Велике значення має також характер роботи.

Ергономічними аспектами проектування відеотермінальних робочих місць, зокрема, є: висота робочої поверхні, розміри простору для ніг, вимоги до розташування документів на робочому місці (наявність і розміри підставки для

документів, можливість різного розміщення документів, відстань від очей користувача до екрану, документа, клавіатури і т.д.), характеристики робочого крісла, вимоги до поверхні робочого столу, урегульованість елементів робочого місця.

Головними елементами робочого місця програміста є стіл і крісло. Основним робочим положенням є положення сидячи.

Оптимальне розміщення предметів праці і документації в зонах досяжності:

- 1) дисплей розміщується в зоні а (у центрі);
- 2) системний блок розміщується в передбаченій ніші столу;
- 3) клавіатура – у зоні г/д;
- 4) «миша» – в зоні в справа;
- 5) сканер в зоні а/б (зліва);
- 6) принтер знаходиться в зоні а (праворуч);
- 7) документація: необхідна при роботі – в зоні легкої досяжності долоні – в, а у висувних ящиках столу – література, невикористовувана постійно.

Для комфортної роботи стіл повинен задовольняти наступним умовам:

- 1) висота столу повинна бути вибрана з урахуванням можливості сидіти вільно, в зручній позі, при необхідності спираючись на підлокітники;
- 2) нижня частина столу повинна бути сконструйована так, щоб програміст міг зручно сидіти, не був змушений підбирати ноги;
- 3) поверхня столу повинна мати властивості, що виключають появу відблисків у поле зору програміста;
- 4) конструкція столу повинна передбачати наявність висувних ящиків (не менше 3 для зберігання документації, лістингів, канцелярських приналежностей);
- 5) висота робочої поверхні рекомендується в межах 680-760мм. Висота поверхні, на яку встановлюється клавіатура, повинна бути близько 650мм.

Велике значення надається характеристикам робочого крісла. Так, рекомендована висота сидіння над рівнем підлоги перебуває в межах 420-550мм. Поверхня сидіння м'яка, передній край закруглений, а кут нахилу спинки - регульований.

Велике значення також надається правильній робочій позі користувача. При незручній робочій позі можуть з'явитися болі в м'язах, суглобах і сухожиллях. Вимоги до робочої пози користувача відеотерміналу наступні:

- 1) голова не повинна бути нахилена більш ніж на 20° ;
- 2) плечі повинні бути розслаблені;
- 3) лікті - під кутом $80^{\circ} \dots 100^{\circ}$;
- 4) передпліччя і кисті рук - в горизонтальному положенні.

Причина неправильної пози користувачів обумовлена наступними чинниками: немає хорошої підставки для документів, клавіатура знаходиться дуже високо, а документи - низько, нікуди покласти руки і кисті, недостатній простір для ніг.

3.3 Розрахунок освітленості і рівня шуму

Розрахунок освітленості робочого місця зводиться до вибору системи освітлення, визначенню необхідного числа світильників, їхнього типу і розміщення. Виходячи з цього, розрахуємо параметри штучного освітлення.

Зазвичай штучне освітлення виконується за допомогою електричних джерел світла двох видів: ламп накаливання і люмінесцентних ламп. Використовуватимемо люмінесцентні лампи, які порівняно з лампами розжарювання мають ряд істотних переваг:

- 1) за спектральним складом світла вони близькі до денного, природного світла;
- 2) володіють більш високим ККД (у 1,5-2 рази вище, ніж ККД ламп розжарювання);
- 3) мають підвищену світловіддачу (в 3-4 рази вище, ніж у ламп розжарювання);
- 4) більш тривалий термін служби.

Розрахунок освітлення проводиться для кімнати площею 15 м^2 , ширина якої 5 м , висота - 3 м . Для визначення кількості світильників визначається світловий потік, де

F - розраховується світловий потік, Лм;

E - нормована мінімальна освітленість, Лк (визначається за таблицею). Роботу програміста, відповідно до цієї таблиці, можна віднести до розряду точних робіт, отже, мінімальна освітленість $E = 300\text{ лк}$;

S - площа освітлюваного приміщення (у нашому випадку $S = 15\text{ м}^2$);

Z - відношення середньої освітленості до мінімальної (звичайно приймається рівним $1,1 \dots 1,2$, нехай $Z = 1,1$);

K - коефіцієнт запасу, враховує зменшення світлового потоку лампи в результаті забруднення світильників у процесі експлуатації (його значення залежить від типу приміщення й характеру проведених у ньому робіт і в нашому випадку $K = 1,5$);

n - коефіцієнт використання, (виражається відношенням світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в частках одиниці; залежить від характеристик світильника, розмірів приміщення, фарбування стін і стелі, які характеризуються коефіцієнтами відображення від стін (R_C) і стелі (R_{Π})), значення коефіцієнтів R_C і R_{Π} були зазначені вище: $R_C = 40\%$, $R_{\Pi} = 60\%$. Значення n визначимо по таблиці коефіцієнтів використання різних світильників. Для цього обчислимо індекс приміщення:

S - площа приміщення, $S = 15\text{ м}^2$;

h - розрахункова висота підвісу, $h = 2.92\text{ м}$;

A - ширина приміщення, $A = 3\text{ м}$;

B - довжина приміщення, $B = 5\text{ м}$.

Знаючи індекс приміщення I , за таблицею 7 [23] знаходимо $n = 0,22$

Для освітлення вибираємо люмінесцентні лампи типу ЛБ40-1, світловий потік яких $F = 4320\text{ Лк}$.

Розрахуємо необхідну кількість ламп:

N - обумовлений число ламп;

F - світловий потік, $F = 33750$ Лм;

$F_{л}$ - світловий потік лампи, $F_{л} = 4320$ Лм.

При виборі освітлювальних приладів використовуємо світильники типу ОД. Кожен світильник комплектується двома лампами.

Розрахунок рівня шуму.

Одним з несприятливих факторів виробничого середовища в ІОЦ є високий рівень шуму, створюваний друкованими пристроями, обладнанням для кондиціонування повітря, вентиляторами систем охолодження в самих ЕОМ.

Для вирішення питань про необхідність і доцільність зниження шуму необхідно знати рівні шуму на робочому місці оператора.

Рівень шуму, що виникає від декількох некогерентних джерел, що працюють одночасно, підраховується на підставі принципу енергетичного підсумовування випромінювань окремих джерел:

L_i - рівень звукового тиску i -го джерела шуму;

n - кількість джерел шуму.

Отримані результати розрахунку порівнюється з допустимим значенням рівня шуму для даного робочого місця. Якщо результати розрахунку вище допустимого значення рівня шуму, то необхідні спеціальні заходи щодо зниження шуму. До них відносяться: облицювання стін і стелі залу звукопоглинальними матеріалами, зниження шуму в джерелі, правильне планування обладнання і раціональна організація робочого місця оператора.

Рівні звукового тиску джерел шуму, що діють на оператора на його робочому місці представлені в табл. 3.4.

Таблиця 3.4 - Рівні звукового тиску різних джерел.

Джерело шуму	Рівень шуму, дБ
Жорсткий диск	40

Вентилятор	45
Монітор	17
Клавіатура	10
Принтер	45
Сканер	42

Зазвичай робоче місце оператора оснащено наступним обладнанням: вінчестер в системному блоці, вентилятор (и) систем охолодження ПК, монітор, клавіатура, принтер і сканер.

Підставивши значення рівня звукового тиску для кожного виду обладнання у формулу, отримаємо:

$$L_{\Sigma} = 10 \cdot \lg (10^4 + 10^{4,5} + 10^{1,7} + 10^1 + 10^{4,5} + 10^{4,2}) = 49,5 \text{ дБ}$$

Отримане значення не перевищує допустимий рівень шуму для робочого місця оператора, рівний 65 дБ (ГОСТ 12.1.003-83). І якщо врахувати, що навряд чи такі периферійні пристрої як сканер і принтер будуть використовуватися одночасно, то ця цифра ще нижчою. Крім того при роботі принтера безпосередню присутність оператора необов'язково, тому що принтер обладнаний механізмом автоподачі аркушів.

3.4 Заходи та засоби протипожежного захисту

Під пожежною безпекою розуміють такий стан промислового або цивільного об'єкта, за якого з регламентованою ймовірністю виключається можливість виникнення і розвитку пожеж та впливу на людей небезпечних чинників пожежі, а також забезпечується захист матеріальних цінностей та довкілля.

Пожежна безпека об'єкта – доволі складне і багатоаспектне завдання, тому для його вирішення потрібно підходити комплексно. Комплекс заходів та засобів щодо пожежної безпеки складається із відповідних систем, зокрема:

1) системи запобігання пожеж, що містить підсистеми запобігання утворенню горючого середовища та виникненню в горючому середовищі джерела запалювання;

2) системи протипожежного захисту, що, своєю чергою, містить такі підсистеми: підсистема обмеження розвитку пожежі; підсистема забезпечення безпечної евакуації людей та майна; підсистема створення умов для успішного гасіння пожежі;

3) системи організаційно-технічних заходів, що передбачає організаційні, технічні, режимні та експлуатаційні заходи.

Організаційні заходи пожежної безпеки передбачають організацію пожежної охорони на об'єкті, проведення навчань з питань пожежної безпеки (інструктажі та пожежно-технічні мінімуми), застосування наочних засобів протипожежної пропаганди та агітації, організацію ДПД та ПТК, проведення перевірок, оглядів стану пожежної безпеки приміщень, будівель, об'єкта загалом та ін.

До технічних заходів належать суворе дотримання правил і норм, визначених чинними нормативними документами при реконструкції приміщень, будівель та об'єктів, технічному переоснащенні виробництва, експлуатації чи можливому переобладнанні електромереж, опалення, вентиляції, освітлення тощо.

Заходи режимного характеру передбачають заборону куріння та застосування відкритого вогню у недозволених місцях, недопущення появи сторонніх осіб у вибухонебезпечних приміщеннях чи об'єктах, регламентацію пожежної безпеки при проведенні вогневих робіт тощо.

Експлуатаційні заходи передбачають своєчасне проведення профілактичних оглядів, випробувань, ремонтів технологічного та допоміжного устаткування, а також інженерного господарства (електромереж, електроустановок, опалення, вентиляції).

Система запобігання пожежі – це комплекс заходів і технічних засобів, які запобігають виникненню пожежі.

Керівники підприємств повинні визначити обов'язки посадових осіб (у тому числі заступників керівника) з забезпечення пожежної безпеки, призначити

відповідальних за пожежну безпеку окремих будівель, споруд, приміщень, діляниць, технологічного та інженерного обладнання, а також за зберігання та експлуатацію технічних засобів протипожежного захисту.

Обов'язки осіб, відповідальних за забезпечення пожежної безпеки, утримання та експлуатації засобів протипожежного захисту слід відобразити у відповідних документах.

Керівник підприємства зобов'язаний вживати (в межах наданих йому повноважень) відповідних заходів реагування на факти порушень чи невиконання іншими працівниками підприємства встановленого протипожежного режиму, вимог правил пожежної безпеки та нормативних актів, що діють у цій сфері.

Керівники підприємств повинні:

1) організувати розроблення комплексних заходів для забезпечення пожежної безпеки, впроваджувати на підприємстві досягнення науки і техніки, позитивний досвід;

2) відповідно до нормативних актів з пожежної безпеки розробляти і затверджувати положення, інструкції та інші нормативні акти, що діють у межах підприємства, здійснювати постійний контроль за їх додержанням;

3) забезпечувати додержання протипожежних вимог стандартів, норм, правил, а також виконання вимог приписів і постанов органів держпожнадзора;

4) організовувати навчання працівників правилам пожежної безпеки та пропаганду заходів для їх забезпечення;

5) в разі відсутності в нормативних актах вимог, потрібних для гарантування пожежної безпеки - вживати відповідних заходів, узгоджуючи їх з органами держпожнадзора;

6) тримати у справному стані засоби протипожежного захисту і зв'язку, пожежну техніку, обладнання та інвентар, не допускати їх використання не за призначенням;

7) створювати в разі потреби відповідно до встановленого порядку підрозділи пожежної охорони та потрібну для їх функціонування матеріально-технічну базу;

8) подавати на вимогу Державної пожежної охорони відомості та документи про стан пожежної безпеки підприємства (об'єкта) і продукції, яку підприємство виробляє;

9) вживати заходів з впровадження автоматичних засобів виявлення і гасіння пожеж та використання з цією метою виробничої автоматики;

10) своєчасно інформувати пожежну охорону про несправність пожежної техніки, систем протипожежного захисту, водопостачання, а також завчасно інформувати про закриття доріг і проїздів на своїй території;

11) проводити службове розслідування випадків пожеж.

До первинних засобів пожежогасіння належать:

- 1) вогнегасники;
- 2) пожежні крани-комплекти, ручні насоси
- 3) лопати, ломы, сокири, гаки, пили, багри;
- 4) ящики з піском, бочки з водою;
- 5) азбестові полотнища, повстяні мати та ін.

Первинні засоби пожежогасіння розміщують на пожежних щитах, які встановлюють на території об'єкта з розрахунку один щит на 5000 м². Вони мають бути пофарбовані у червоний колір, а пожежний інструмент у чорний.

Серед первинних засобів пожежогасіння найважливішу роль відіграють вогнегасники різних типів: водяні, водо-пінні, порошкові, вуглекислотні, газові.

Залежно від способу транспортування вони бувають: переносні (до 20 кг) та пересувні (до 450 кг).

Залежно від об'єму вогнегасники бувають малолітражні (до 5л), ручні (до 10 л), пересувні (понад 10л).

Вогнегасники маркують буквами, що означає їх вид та цифрами, що визначають їх об'єм.

Найбільш перспективними є порошкові вогнегасники, які застосовують для гасіння лужних металів, ЛЗР і ТР, електрообладнання, що горить під напругою до 1000В, твердих та газоподібних речовин.

Найбільш розповсюдженими є:

- 1) ОП-1, ОП-2, ОП-9, ОП-10 — переносні;
- 2) ОПА-50, ОПА-100 — пересувні.

Вони відрізняються між собою лише складом порошку та пристроєм для його подачі.

Вуглекислотні вогнегасники застосовуються для гасіння загорянь на машинах, автомобілях і для невеликих об'ємів нафтопродуктів, а також електроустановок під напругою до 1000В.

У корпусі вогнегасника міститься вуглекислий газ у рідкому стані під високим тиском 6МПа (ручні) і 15 мПа (переносні). У горловині балону змонтований спеціальний пусковий пристрій із сифонною трубкою, який приводиться у дію за допомогою вентильного або пістолетного пристрою. Виходячи з балону назовні, зріджений двооксид вуглецю перетворюється на снігоподібну масу за температури - 80°C.

Вибір типу вогнегасника визначається розмірами загоряння і можливих осередків пожеж.

Утворенню горючого середовища запобігають застосуванням герметичного виробничого устаткування, максимально можливою заміною в технологічних процесах горючих речовин та матеріалів негорючими, обмеженням кількості пожежо-, вибухонебезпечних речовин та матеріалів під час використання та зберігання, а також правильним їх розміщенням, ізоляцією горючого та вибухонебезпечного середовища, організацією контролю за складом повітря в приміщенні та контролю за станом середовища в апаратах, застосуванням робочої та аварійної вентиляції, відведенням горючого середовища в спеціальні пристрої та безпечні місця, застосуванням в установках з горючими речовинами пристроїв від пошкодження та аварій, використанням інгібувальних (хімічно активні компоненти, що сприяють припиненню пожеж) та флегматизаційних (інертні компоненти, що роблять середовище негорючим) речовин.

Виникненню в горючому середовищі джерела запалювання запобігають використанням устаткування та пристроїв, при роботі яких не виникає джерел запалювання, використанням електроустаткування, що відповідає за

досягнення класу пожежо- та вибухонебезпеки приміщеннями та зонами груп і категорій вибухонебезпечної суміші, виконанням вимог щодо сумісного зберігання речовин та матеріалів, використанням устаткування, що задовольняє вимоги електростатичної іскробезпеки, улаштуванням блискавкозахисту, організацією автоматичного контролю параметрів, що визначають джерела запалювання, використанням швидкодіючих засобів захисного вимкнення, заземленням устаткування, видовжених металоконструкцій, використанням при роботі з ЛЗР інструментів, що не допускають іскроутворення, ліквідацією умов для самоспалахування речовин і матеріалів, усуненням контакту з повітрям пірофорних речовин, підтриманням температури нагрівання поверхні устаткування, пристроїв, речовин та матеріалів, які можуть контактувати з горючим середовищем нижче гранично допустимої (80 %) температури займання.

У даному розділі дипломного проекту були викладені вимоги до робочого місця техника-програміста. Створені умови повинні забезпечувати комфортну роботу. На підставі вивченої літератури з даної проблеми, були зазначені оптимальні розміри робочого столу і крісла, робочої поверхні, а також проведено вибір системи і розрахунок оптимального освітлення виробничого приміщення, а також розрахунок рівня шуму на робочому місці. Дотримання умов, що визначають оптимальну організацію робочого місця інженера - програміста, дозволить зберегти гарну працездатність протягом усього робочого дня, підвищить як в кількісному, так і в якісному відношенні продуктивність праці програміста, що в свою чергу сприятиме якнайшвидшій розробці і налагодженню програмного продукту.

Висновки

Дипломний проект має мету - реалізацію програмного модуля «Віртуальна клавіатура». Програмний засіб представляє собою програмний додаток, який дозволяє виконувати роботу як клавіатурою, мишею та екраном комп'ютера. Модуль має такі додаткові функції, як налаштування клавіатури для потреб користувача, вибір текстового поля для вводу текстового та цифрового тексту, а також відкриття інших програм через даний модуль.

У першому розділі дипломного проекту описані основні задачі для програмного засобу, мінімальні вимоги до системи для коректного функціонування засобу, короткі теоретичні відомості про віртуалізацію, віртуальну реальність та віртуальні клавіатури. Також у даному розділі обрано середовище для розробки програмного засобу та мову програмування.

У другому розділі роботи представлений алгоритм розробки програмного модуля, основні діаграми та схеми функціонування програмного засобу, аналіз вимог, способи, засоби та методи реалізації основних функцій, порівняння даного засобу із програмами-аналогами, детальна інструкція користувача програмного продукту, тестування програми.

У розділі «Охорона праці» описуються загальні вимоги до робочого місця програміста. У вимогах до виробничих приміщень були зазначені оптимальні розміри робочого столу і крісла, робочої поверхні, а також проведено вибір системи і розрахунок оптимального освітлення виробничого приміщення, а також розрахунок рівня шуму на робочому місці.

В подальшому можливе розширення функціональності програмного засобу. Зазвичай аналогічні програмні засоби використовують для того, аби полегшити способи вводу тексту або знайти альтернативні методи вводу та зміни тексту.

У майбутньому можливе розширення функціональності програмного засобу шляхом додавання інших мов, а також можна додати можливість голосового вводу та вводу за допомогою пера.

Перелік використаних джерел

1. Аббасов, И.Б. Двухмерное и трехмерное моделирование в *3ds MAX* / И.Б. Аббасов. - М.: ДМК, 2012. - 176 с.
2. Ганеев, Р.М. 3D-моделирование персонажей в *Maya*: Учебное пособие для вузов / Р.М. Ганеев. - М.: ГЛТ, 2012. - 284 с.
3. Климачева, Т.Н. *AutoCAD*. Техническое черчение и 3D-моделирование. / Т.Н. Климачева. - СПб.: BHV, 2008. - 912 с.
4. Пекарев, Л. Архитектурное моделирование в *3ds Max* / Л. Пекарев. - СПб.: BHV, 2007. - 256 с.
5. Петелин, А.Ю. 3D-моделирование в *Google Sketch Up* - от простого к сложному. Самоучитель / А.Ю. Петелин. - М.: ДМК Пресс, 2012. - 344 с.
6. Погорелов, В. *AutoCAD 2009*: 3D-моделирование / В. Погорелов. - СПб.: BHV, 2009. - 400 с.
7. Полещук, Н.Н. *AutoCAD 2007*: 2D/3D-моделирование. / Н.Н. Полещук. - М.: Русская редакция, 2007. - 416 с.
8. Тозик, В.Т. 3ds Max Трехмерное моделирование и анимация на примерах / В.Т. Тозик. - СПб.: BHV, 2008. - 880 с.
9. Швембергер, С.И. *3ds Max*. Художественное моделирование и специальные эффекты / С.И. Швембергер. - СПб.: BHV, 2006. - 320 с.
10. Еріх Гамма, Річард Хелм, Джон Вліссідес, Ральф Джонсон. Паттерны проектирования. – Питер, 2001, – 395 с.
11. А. Пол. Объектно-ориентированное программирование в действии.– СПб.: Питер, 1997, 447 с.
12. Дж. Ликнесс Приложения для *Windows 8* на *C#* и *XAML*. – Питер, 2013, – 368 с.
13. Святослав Куликов – Тестирование программного обеспечения Базовый курс (2-е издание), 2018.

14. Сью Блэкман, *Beginning 3D Game Development with Unity*/ Сью Блэкман; Apress, 2011, 992 с.
15. Фохт, Д. Проектирование и программная реализация систем на персональных ЭВМ/ Д. Фохт СПб.: БХВ - Петербург, 2005. - 96 с.
16. Крейтон, P.X. *Unity Game Development Essentials* / P.X. Крейтон Packt Publishing, 2010, 83 с.
17. Мартынов, Н.Н. *C# для начинающих* / Н.Н. Мартынов. - Москва; Кудиц-пресс, 2007. - 272 с.
18. Павловская, Т.А. *C#. Программирование на языке высокого уровня* / Т.А. Павловская - Спб; ПИТЕР, 2009 432 с.
19. Фленов, М. Библия C#/Фленов М. СПб.: БХВ - Петербург, 2011. -560с.
20. Фленов, М. *C#. Секреты программирования* / М. Фленов - Спб, ПИТЕР, 2006. - 457с
21. Дубовці В.А. Безпека життєдіяльності. / Учеб. посібник для дипломників. - К.: вид. Кірпи, 1992.
22. Мотузко Ф.Я. Охорона праці. - М.: Вища школа, 1989. - 336с.
23. Безпека життєдіяльності. / Под ред. Н.А. Белова - М.: Знання, 2000 - 364с.
24. Самгін Е.Б. Освітлення робочих місць. - М.: МІРЕА, 1989. - 186с.
25. Довідкова книга для проектування електричного освітлення. / Под ред. Г.Б. Кнорринга. - Л.: Енергія, 1976.
26. Боротьба з шумом на виробництві: Довідник / Є.Я. Юдін, Л.А. Борисов; За заг. ред. Є.Я. Юдіна - М.: Машинобудування, 1985. - 400с., Іл.Зінченко В.П. Основи ергономіки. - М.: МГУ, 1979. - 179с.
27. Методичні вказівки з виконання дипломної роботи (проекту) для студентів спеціальності 5.05010301 «Розробка програмного забезпечення».

Додаток А – Код модуля MainWindow

```
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Windows.Forms;
using System.Xml.Linq;
using System.Xml.XPath;
using Example.Properties;
using Keyboard;

namespace Example
{
    public enum Languages
    {
        English = 1,
        Russian = 2,
        Greerk = 3
    }

    public partial class KeyboradExampleForm : Form
    {
        const uint WS_EX_NOACTIVATE = 0x08000000;
        const uint WS_EX_TOPMOST = 0x00000008;
        private const string en = "En";
        private const string gr = "Gr";
        private const string ru = "Ru";
        private int langType = 1;
        private string layoutDefaultFilePath = Application.StartupPath + "\\KeyboardLayouts";

        protected override CreateParams CreateParams
        {
            get
            {
                CreateParams baseParams = base.CreateParams;
                baseParams.ExStyle |= (int)(WS_EX_NOACTIVATE | WS_EX_TOPMOST);
                return baseParams;
            }
        }
    }
}
```

```

public KeyboradExampleForm()
{
    InitializeComponent();
}

private void virtualKeyboard1_KeyboardButtonPressed(string command, KeyboardButtonEventArgs e)
{
    if (command == KeyboardKeyConstants.LanguageSelection)
    {
        switch (virtualKeyboard1.LanguageButtonTopText)
        {
            case en:
                SetLangButton(ru, gr, Resources.Flag_Ru);
                langType = (int)Languages.Russian;
                SetRusLayout();
                break;
            case ru:
                SetLangButton(gr, en, Resources.Flag_Greece);
                langType = (int)Languages.Greerk;
                SetGreekLayout();
                break;
            case gr:
                SetLangButton(en, ru, Resources.Flag_UK);
                langType = (int)Languages.English;
                ResetLayout();
                break;
        }
    }
    else
    {
        switch (langType)
        {
            case (int) Languages.English:
                ResetLayout();
                break;
            case (int) Languages.Russian:
                SetRusLayout();
                break;
            case (int)Languages.Greerk:

```

```

        SetGreekLayout();
        break;
    default:

        break;
    }
}

private void SetRusLayout()
{
    var xmlPath = layoutDefaultFilePath + "\\KeyboardLayout_RusLower.XML";

    if (virtualKeyboard1.ShiftState && virtualKeyboard1.CapsLockState)
    {
        xmlPath = layoutDefaultFilePath + "\\KeyboardLayout_RusLower.XML";
    }
    else if (virtualKeyboard1.ShiftState || virtualKeyboard1.CapsLockState)
    {
        xmlPath = layoutDefaultFilePath + "\\KeyboardLayout_Rus.XML";
    }
    else
    {
        xmlPath = layoutDefaultFilePath + "\\KeyboardLayout_RusLower.XML";
    }
    XDocument xmlDocument = XDocument.Load(xmlPath);
    var firstRow = LoadButtonsRow(xmlDocument, 1);
    var secondRow = LoadButtonsRow(xmlDocument, 2);
    var thirdRow = LoadButtonsRow(xmlDocument, 3);
    var fourthRow = LoadButtonsRow(xmlDocument, 4);
    var fifthRow = LoadButtonsRow(xmlDocument, 5);

    virtualKeyboard1.FirstRowCustomButtons = firstRow;
    virtualKeyboard1.SecondRowCustomButtons = secondRow;
    virtualKeyboard1.ThirdRowCustomButtons = thirdRow;
    virtualKeyboard1.FourthRowCustomButtons = fourthRow;
    virtualKeyboard1.FifthRowCustomButtons = fifthRow;
}

private void SetGreekLayout()
{
    var xmlPath = layoutDefaultFilePath + "\\KeyboardLayout_GrLower.XML";

```



```

if (virtualKeyboard1.AltGrState)
{
    xmlPath = layoutDefaultFilePath + "\\KeyboardLayout_GrAltGr.XML";
}
else
{
    if (virtualKeyboard1.ShiftState && virtualKeyboard1.CapsLockState)
    {
        xmlPath = layoutDefaultFilePath + "\\KeyboardLayout_GrLower.XML";
    }
    else if (virtualKeyboard1.ShiftState || virtualKeyboard1.CapsLockState)
    {
        xmlPath = layoutDefaultFilePath + "\\KeyboardLayout_Gr.XML";
    }
    else
    {
        xmlPath = layoutDefaultFilePath + "\\KeyboardLayout_GrLower.XML";
    }
}
XDocument xmlDocument = XDocument.Load(xmlPath);
var firstRow = LoadButtonsRow(xmlDocument, 1);
var secondRow = LoadButtonsRow(xmlDocument, 2);
var thirdRow = LoadButtonsRow(xmlDocument, 3);
var fourthRow = LoadButtonsRow(xmlDocument, 4);
var fifthRow = LoadButtonsRow(xmlDocument, 5);

virtualKeyboard1.FirstRowCustomButtons = firstRow;
virtualKeyboard1.SecondRowCustomButtons = secondRow;
virtualKeyboard1.ThirdRowCustomButtons = thirdRow;
virtualKeyboard1.FourthRowCustomButtons = fourthRow;
virtualKeyboard1.FifthRowCustomButtons = fifthRow;
}

private void SetLangButton(string topText, string bottomText, Image img)
{
    virtualKeyboard1.LanguageButtonTopText = topText;
    virtualKeyboard1.LanguageButtonBottomText = bottomText;
    virtualKeyboard1.LanguageButtonImage = img;
}

private void btn_RunWordPad_Click(object sender, System.EventArgs e)
{

```

```

        System.Diagnostics.Process.Start("wordpad.exe");
    }

    private void btn_RunNotepad_Click(object sender, System.EventArgs e)
    {
        System.Diagnostics.Process.Start("notepad.exe");
    }

    private void KeyboradExampleForm_Load(object sender, System.EventArgs e)
    {
        virtualKeyboard1.ShowLanguageButton = true;
        SetLangButton(en, ru, Resources.Flag_UK);
        langType = (int)Languages.English;
        ResetLayout();
    }

    private ButtonsCollection LoadButtonsRow(XDocument xmlDocument, int rowNumber)
    {
        var buttonsCollection = new ButtonsCollection();

        // Loads buttons text by XML row number
        var node = xmlDocument.XPathSelectElement(string.Format("//Row[RowNumber = {0}]/Buttons",
rowNumber));
        var dataText = (from contact in node.Descendants("Text")
            select new
            {
                TopText = contact.Element("UpText").Value,
                BottomText = contact.Element("BottomText").Value
            }).ToList();

        foreach (var text in dataText)
        {
            var kbButton = new VirtualKbButton();
            kbButton.TopText = text.TopText;
            kbButton.BottomText = text.BottomText;
            buttonsCollection.Add(kbButton);
        }
        return buttonsCollection;
    }

    private void ResetLayout()
    {
        StateHelpers.ApplyState(virtualKeyboard1);
    }
}
}

```