

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних та видавничих технологій
циклова комісія інформатики та комп'ютерних дисциплін

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач відділення комп'ютерних
та видавничих технологій

Чубей О.О. / _____ /

підпис

« ____ » _____ 2022 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проєкту
освітньо-кваліфікаційного рівня «молодший спеціаліст»
зі спеціальності 122«Комп'ютерні науки»

на тему: «Дитяча розвиваюча гра для дошкільнят»

Студент групи КН-41 Труш Максим

(підпис)

Керівник проєкту Сиротюк Н. С.

(підпис)

Консультанти:

з техніко-економічного Меленчук Л.І.
обґрунтування

(підпис)

нормоконтролер Кульчинська Н.З.

(підпис)

Тернопіль - 2022

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних та видавничих технологій
циклова комісія інформатики та комп'ютерних дисциплін

ЗАТВЕРДЖУЮ

Завідувач відділенням
комп'ютерних та видавничих
технологій

Чубей О.О. / _____ /

підпис

«___» _____ 2021 р.

ЗАВДАННЯ

на дипломне проектування
на здобуття освітньо-кваліфікаційного рівня «молодший спеціаліст»
студенту Трушу Максиму Володимировичу
(прізвище, ім'я та по-батькові студента)

1. Тема проекту «Дитяча розвиваюча гра для дошкільнят»

затверджена наказом по коледжу від «___» _____ 2021р., №___

2. Термін здачі студентом завершеного проекту «___» _____ 2022 р.

3. Вихідні дані до проекту _____

4. Перелік питань, які повинні бути розроблені в проекті:

а) основна частина _____

б) техніко-економічне обґрунтування _____

5. Перелік графічного матеріалу _____

6. Консультанти проекту: _____

Розділ	Консультанти	Підпис, дата	
		Завдання видано	Завдання прийнято
з техніко-економічного обґрунтування	<div>(вчена ступень, звання П.І.Б. консультанта)</div>		

КАЛЕНДАРНИЙ ПЛАН
дипломного проектування

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1.	Вибір теми, ознайомлення з вимогами до дипломного проектування	27.09.21 р.	30.09.21 р.
2.	аналіз типових рішень та написання відповідного розділу ПЗ	20.11.21 р.	27.01.22 р.
3.	Дослідження технологій реалізації та написання відповідного розділу ПЗ	27.01.22 р.	14.02.22 р.
4.	Розробка функціональних вимог до проекту та робота над структурою програмного продукту. Написання відповідного розділу ПЗ.	17.02.22 р.	02.03.22 р.
5.	Встановлення та налаштування середовища реалізації та написання відповідного розділу ПЗ.	02.03.22 р.	16.03.22 р.
6.	Проектування програмного засобу (функціоналу, інтерфейсу, бази даних продукту) та написання відповідного розділу ПЗ	16.03.22 р.	17.04.22 р.
7.	Реалізація та налаштування програмного засобу та написання відповідного розділу ПЗ	17.04.22 р.	04.05.22 р.
8.	Тестування та налагодження програмного продукту та написання відповідного розділу ПЗ	05.05.22 р.	18.05.22 р.
9.	Опрацювання економічного розділу дипломного проекту та оформлення спеціального розділу	18.05.22 р.	19.06.22 р.
10.	Робота над оформленням пояснювальної записки	19.05.22 р.	05.06.22 р.
11.	Оформлення пояснювальної записки	08.06.22 р.	18.06.22 р.
12.	Попередній захист дипломного проекту	15.06.22 р.	22.06.22 р.
13.	Підготовка до захисту дипломного проекту	22.06.22 р.	25.06.22 р.
14.	Захист дипломного проекту	25.06.22 р.	26.06.22 р.

7. Дата видачі завдання “___” _____ 2021 р.

Керівник _____/

Завдання прийняв до виконання _____/

Реферат

Дипломний проєкт. «Дитяча розвиваюча гра для дошкільнят». Труш Максим Галицький фаховий коледж імені В'ячеслава Чорновола, відділення комп'ютерних а видавничих технологій. Спеціальність 122 «Комп'ютерні науки». ГФК, 2022. 64 – сторінок, 34 – рисунків, 11 – джерел.

Дипломне проектування націлене на дослідження для отримання потрібної інформації про предметну область та деякі особливості обраної теми було взято два інтернет ресурси: childdevelop.com.ua; logiclike.com.

Мета проєкту – фактичне створення десктопного додатку для платформи Windows, який зможе використовувати дитина дошкільного віку та буде задовольняти концепцію гри для розвитку.

Завдання проєкту – це створення додатку засобами Windows Form, що буде відповідати подальшим поставленим вимогам.

Кінцевим результатом роботи буде десктопний додаток, що буде максимально простим для використання. Легкий для орієнтування інтерфейс та привабливість за рахунок яскравих картинок. Додаток, що буде позитивно впливати на логіку та увагу гравця.

Середовищем розробки додатку виступатиме Visual Studio 2019, що використовується для розробки різноманітних додатків під різні платформи в тому ж числі десктопних для операційної системи Windows. Щоб здійснити створення усього запланованого функціоналу достатньо використання мови програмування C#.

ДЕСКТОПНИЙ ДОДАТОК, МОВА ПРОГРАМУВАННЯ C#, WINDOWS FORM, РОЗВИВАЮЧА ГРА, ОПЕРАЦІЙНА СИСТЕМА WINDOWS

Abstract

Diploma project. "Children's educational game for preschoolers". Trush Maksym Halytsky Vyacheslav Chornovil Vocational College, Department of Computer and Publishing Technologies. Specialty 122 "Computer Science". GVC, 2022. 64 – pages, 34 – figures, 1– sources.

Diploma design is aimed at research to obtain the necessary information about the subject area and some features of the chosen topic, two Internet resources were taken: «Childdevelop.com.ua»; «LogicLike.com».

The aim of the project is to actually create a desktop application for the Windows platform that can be used by preschoolers and will satisfy the concept of the game for development.

The task of the project is to create an application using Windows Form, which will meet the further requirements.

The end result will be a desktop application that will be as easy to use as possible. Easy to navigate interface and attractiveness due to bright pictures. An application that will positively affect the logic and attention of the player.

The application development environment will be Visual Studio 2019, which is used to develop a variety of applications for different platforms, including desktop for Windows. To create all the planned functionality, it is enough to use the C # programming language.

DESKTOP APPLICATION, PROGRAMMING LANGUAGE C#,
WINDOWS FORM, DEVELOPING GAME, WINDOWS OPERATING SYSTEM

ЗМІСТ

Вступ.....	7
1 Дослідження предметної області.....	8
1.1 Характеристика обраної предметної області	8
1.2 Опис функціональних сегментів предметної області.....	9
1.3 Аналіз вимог до ПС.....	12
1.4 Огляд та аналіз існуючих аналогів.....	13
2 Планування архітектурних складових	20
2.1 Аналіз та визначення потрібних UML діаграм	20
2.2 Створення та опис вибраних UML діаграм	22
3 Програмна реалізація спроектованої системи.....	28
3.1 Опис середовища та технології написання програмного коду	28
3.2 Створення програмних рішень проєкту.....	32
3.3 Створення графічного інтерфейсу програми	38
4 Тестування десктопного додатка.....	41
5 Техніко-економічне обґрунтування	44
5.1 Аналіз ринку	44
5.2 Розрахунок витрат на проектування	50
5.3 Обґрунтування необхідності розробки	52
Висновки	54
Перелік джерел посилань	55
Додатки.....	56

					ДП.КН 22.481.32.000 ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Груш.М.			Дитяча розвиваюча гра для дошкільнят	Літ.	Арк.	Аркушів
Перевір.		Сиротюк Н.С.					5	62
рецензен		Кузик В.М.				ГФК.ВКВТ КН-41		
Н.контр.		Кульчинська Н.З.						
Зав. від.		Чубей О.О.						

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ОС – операційна система.

WF- Windows Forms.

CLR – Common Language Runtime.

GUI – Graphical User Interface

IDE – Integrated development environment

					ДП.КН 22.481.32.000	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Сучасні цифрові технології – це те, що розвивалося надзвичайно стрімко за останні 20 років і лише продовжує рости і змінюватись. Технології також вплинули на отримання освіти в тому числі дошкільнят. В деякому виді даний дипломний проект матиме підтекст досліджування впливу використання цифрових технологій для дітей дошкільного віку. Однозначно технології можуть позитивно вплинути на навчання шляхом збільшення інтерактивної взаємодії, приростом мотивації та творчості. В цілому інтерактивні розвиваючі ігри можуть покращити такі важливі навички, як читання, математика, логіка та увага. Технології та їх застосування вчителями є ключовими фасилітаторами для покращення розуміння учнів та відповідне використання технологій. За підтримки шкільних лідерів та районної адміністрації, педагоги можуть покращити результати за допомогою цифрових технологій, а саме розвиваючих ігор.

В процесі виконання дипломної роботи буде створена десктопна розвиваюча гра-додаток для дошкільнят.

Дипломна робота націлена на пошук та аналіз сучасних технологій для створення інтерактивних ігор для дошкільнят, що зможуть водночас покращити деякі особистісні якості та надати досвід використання фактичного комп'ютерного додатку, що також підсвідому навчас працювати з користувацьким інтерфейсом у найпростішому його вигляді.

В більшості випадків для таких цілей розробляють цілі веб-ресурси з великим різноманіттям запропонованих послуг для обслуговування такого роду предметної області. Такі проекти виступають лідерами і відповідно в їх основі лежить велика кількість ресурсів, як людських так і фінансових. Фактичною ціллю даного дипломного проекту є створення доступного десктопного додатку для власного та локального використання.

					ДП.КН 22.481.32.000	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Характеристика обраної предметної області

У дослідженні цифрового життя та ігор дітей молодшого віку є численні напруження та провокації. Технологію іноді представляли як протилежну до гри. Зовсім недавно були визнані переваги залучення дітей до технологій, хоча не завжди їх ігровий потенціал. Дослідження показало, як специфічні характеристики гри сприяють розвитку та навчанню дітей і як цифрова гра може служити освітнім цілям, якщо вона розроблена з урахуванням конкретних результатів навчання. Існує ризик того, що прив'язування цінності цифрової гри надто тісно до вузького визначення навчання може ризикувати ігнорувати інші важливі функції, які виконує гра. Однак виділення конкретних формально освітніх впливів цифрової гри може мати важливе значення, щоб активувати тенденцію можливостей для гри в багатьох системах освіти.

Від народження діти мають більший доступ до цілого ряду цифрових технологій. Це неминуче впливає на пейзажі їхньої гри. Оскільки гра є загально визнаною важливою для дітей, необхідно зрозуміти природу дитячої гри стосовно техніки. Сім'ям потрібна підтримка для батьківства в цифрову епоху, а вихователі дошкільного віку повинні знати, як впровадити цифрові технології у свою власну професійну практику.

Наслідки для маленьких дітей складні. За допомогою цифрової гри діти розвивають широкий спектр навичок, зокрема: знання та розуміння предмета; цифрові навички; і цілісні навички (соціальні, емоційні, когнітивні, фізичні та творчі). Комерціалізація дитячих ігор за допомогою цифрових технологій була предметом тривалого критичного огляду.

Дитяча гра по відношенню до цифрових технологій одночасно схожа на нецифрову гру і відмінна від неї. Вчені зіставили традиційні типи гри з цифровою грою і виявили, що всі представлені. Однак цифрові контексти дають раніше незадокументовані приклади гри, гідні вивчення самі по собі. Цифрове та нецифрове переплітаються в житті маленьких дітей, тому корисно розглядати гру більш цілісно. Потрібні постійні дослідження, щоб

					ДП.КН 22.481.32.000	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

поглибити знання та розуміння точної природи дитячої гри у зв'язку з цифровим – як загалом, так і більш конкретно щодо виявлених прогалин у знаннях.

Використання технології це звичайне явище, як удома, так і в школі. Цифрові інструменти впливають на успіх в співпраці, зобов'язанні, спонуканні, математиці і грамотності. Ефекти освітніх технологій упевнені, якщо батьки і учителі забезпечені належними інструментами, щоб здійснювати вивчення, цифровими засобами. Проте, з певними цифровими інструментами, є негативні результати, які, можливо, мають місце, якщо їх використано недоречно. Майбутні дослідження повинні продовжувати розбирати ефективність професійного розвитку для учителів, поширеність специфічних цифрових інструментів і додатків, які можуть навчати дошкільнят

Використання технології в ранньому дитинстві – ефективний шлях, щоб майбутні учні придбали упевненість, творчий потенціал і співпрацю в навчанні з іншими учнями. Користуючись цифровими інструментами в класній кімнаті, учителі можуть навчити дошкільнят, як учнів 21-х століття, які користуються технологією, щоб залишатися втягнутими в навчальний процес.

1.2 Опис функціональних сегментів предметної області

Для належного функціонування розвиваючої гри для дошкільнят потрібно забезпечити виконання таких основних процесів, які будуть визначатись, як сегменти (рисунок 1.1):

- 1) сегмент пошуку відмінностей;
- 2) сегмент відображення меню;
- 3) сегмент тестування на логіку.

Головним сегментом виступають процеси запуску та ініціалізації всіх необхідних процесів да змінних, що відбувається при запуску додатку. Після запуску та ініціалізації вже стають доступні три інші сегменти.

					ДП.КН 22.481.32.000	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

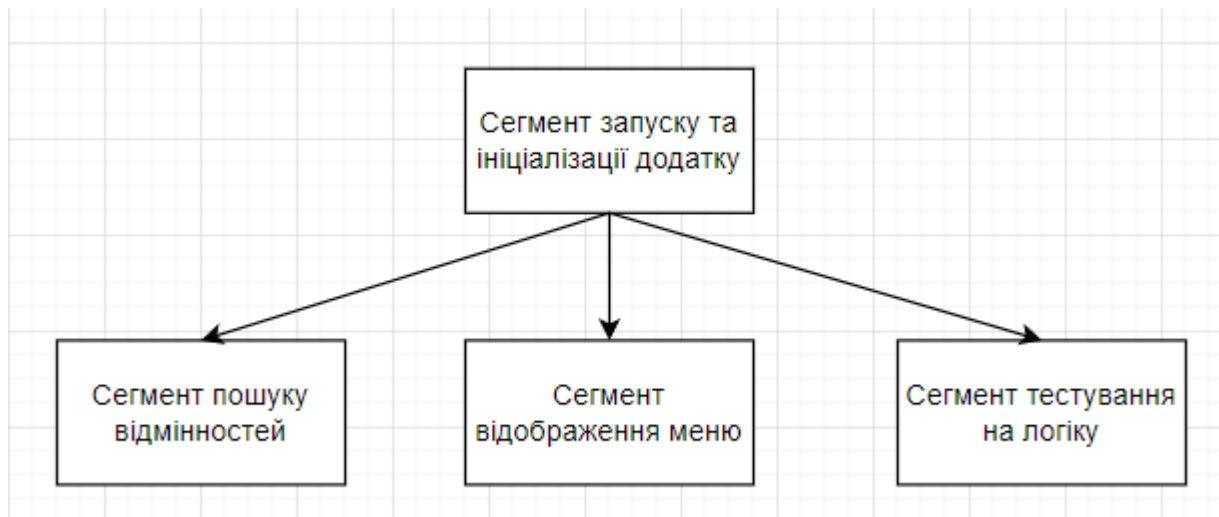


Рисунок 1.1 – Діаграма сегментів розроблюваного програмного продукту

Розглянемо детальніше кожен з вище представлених сегментів. На рисунку 1.2 бачимо діаграму функцій, які потрібні для забезпечення роботи сегменту пошуку відмінностей. Буде описано основні та очевидні функції, що входять в описуваний сегмент, в деякому розумінні буде проведена низькорівнева декомпозиція.

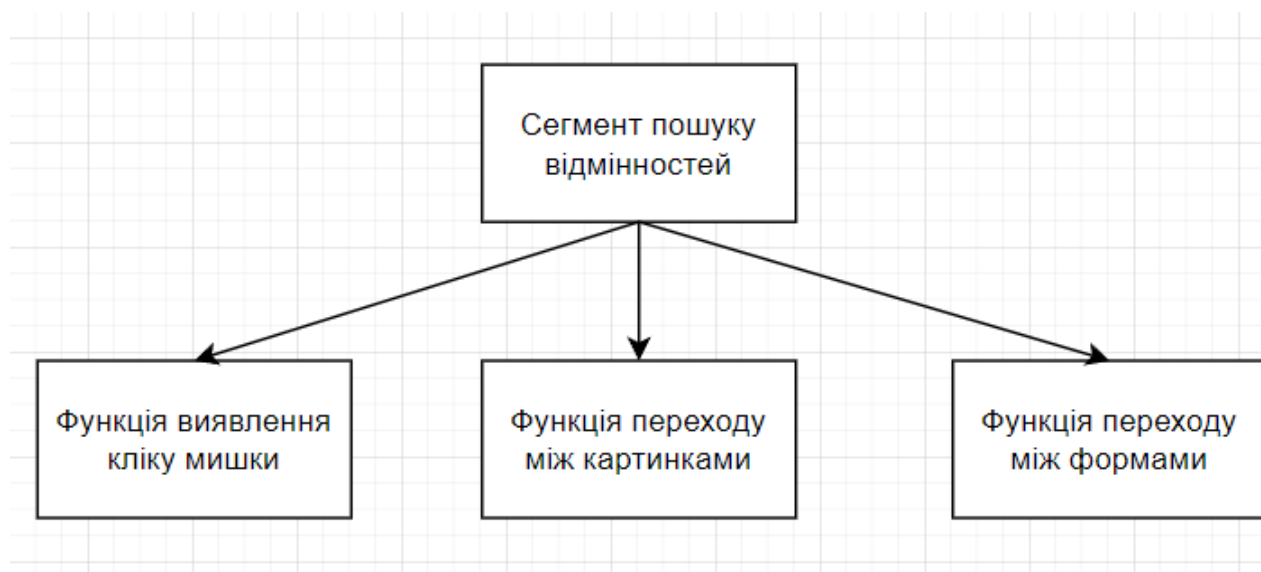


Рисунок 1.2 – Діаграма функцій сегменту пошуку відмінностей

Цей сегмент відповідає за окрему гру пошуку відмінностей. Будуть виконуватись функції взаємодії з картинками, які будуть мати певні відмінності, які потрібно буде відшукати, а саме натиснувши курсором мишки в місце, де присутня деяка відмінність. Така функція собою запускатиме певну

кількість логіки роботи додатку, що буде описано в інших розділах зв'язаних з безпосередньою розробкою гри, як програмного застосунку. Також сегмент відповідає за перехід між різними картинками для знаходження відмінностей та функцію переходу на форму головного меню.

На рисунку 1.3 бачимо діаграму функцій, які потрібні для забезпечення виконання сегменту відображення меню.



Рисунок 1.3 – Діаграма функцій сегменту відображення меню

Цей сегмент є початковим при запуску додатку і відповідає за вибір гри, яка буде відображатись для користувача. Єдиний функціонал, що тут буде виконуватись це перехід між формами, тобто з головного меню на одну з двох обраних користувачем.

На рисунку 1.4 бачимо діаграму функцій, які потрібні для забезпечення виконання сегменту тестування на логіку.

Це другий сегмент, що відповідає за функціонування окремої гри, яка відповідає за тренування логіки дошкільнят. Гра реалізовуватиметься у вигляді тестування, варіанти відповіді та саме завдання якого створюватиметься у вигляді дитячих зрозумілих дошкільнятам картинок.

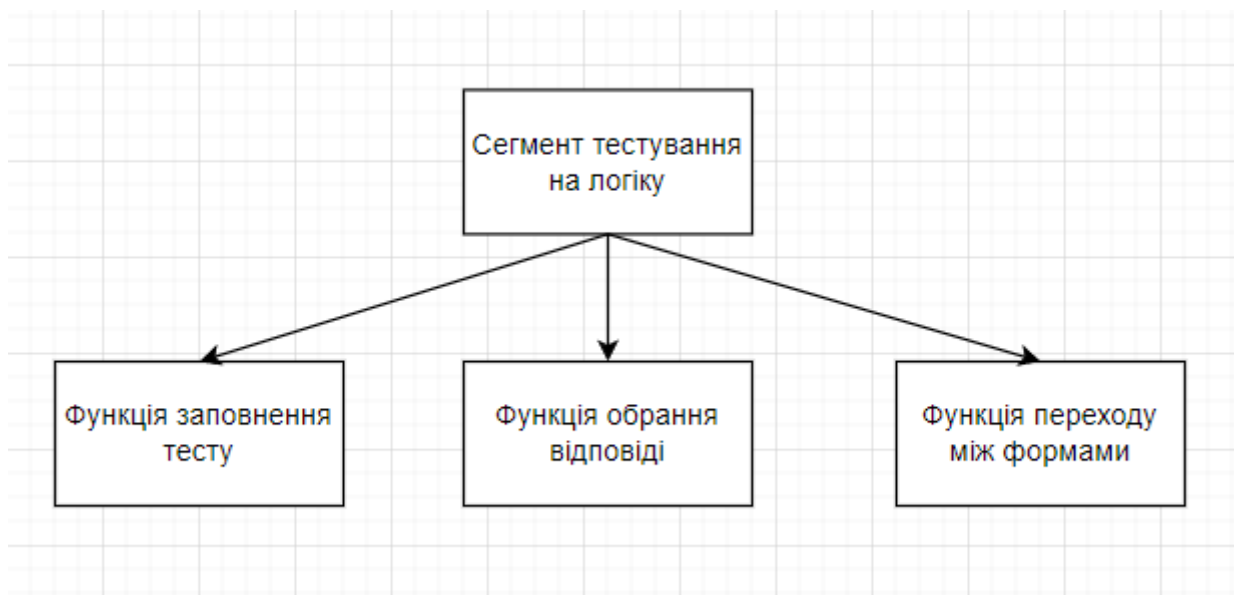


Рисунок 1.4 – Діаграма функцій сегменту тестування на логіку

Кожен з тестів це виконання функції заповнення тесту цього сегменту. При обранні правильного варіанту виконується функція обрання відповіді, що буде за собою вести відповідну програмну логіку. Останньою функцією, як і в інших сегментах являється перехід між формами. Наявність в кожному сегменті такої функцію дозволяє користувачу переміщатись між формами, що є необхідним у подібних додатках.

1.3 Аналіз вимог до ПС

Що стосується вимог до реалізації дипломного проекту, то зазвичай до кожного проекту вони схожі. В основному потрібно працювати над ним у сторону користувача, тобто унеможливити конфлікти між програмою та користувачем, повідомляти користувача при помилці та давати чіткі вказівки, щодо використання програми. Також не мало важливим є той фактор, що дипломний проект являється грою, в яку пропонується грати дітям, які з більшою можливістю по відношенню до дорослих можуть незрозуміти основні вимоги до використання програми. Тому основними вимогами до ПС є:

Реалізувати зручний користувацький інтерфейс з використанням технології CLR Windows Forms.

Для введення/виведення даних використати відповідні елементи

					ДП.КН 22.481.32.000	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

управління.

Створити меню програми, використати зображення.

Обов'язково перевіряти вхідні дані на відповідність та коректність. Хоча в даному випадку вхідними даними являються натиснення відповідних кнопок управління, що не може бути некоректною дією.

Важливою вимогою є збереження даних, які можуть знадобитися після роботи програми, проте це не для даного проекту. Адже при кожному запуску програми відбувається створення всіх даних з нуля і збереження будь-яких даних є непотрібним.

У ході ознайомлення з завданням проекту виникають додаткові вимоги, які потрібно реалізувати при розробці проекту, такі як очищення полотна у зручний для користувача момент, акцентування уваги при обранні правильної відповіді, чітке відображення знайденої відмінності та деякі інші.

1.4 Огляд та аналіз існуючих аналогів

Подібні методи цифрового навчання дошкільнят у вигляді гри та привабливого графічного оформлення для дошкільнят, що сприяє більшій концентрації на навчальному процесі.

Звичайно існує чимало максимально подібних рішень, но зазвичай вони для локального використання або доступні по підписці чи повній купівлі продукту. Значно більше подібних проектів реалізованих у вигляді веб-сервісів чи веб-сайтів.

Для аналізу подібних реалізацій, щоб отримати інформацію для подальшої розробки запланованого програмного продукту було обрано один веб-сервіс childdevelop.com.ua та один веб-сайт logiclike.com. Далі буде проведено деякий аналіз можливостей та наповнення обраних ресурсів. Обидва варіанти відображають в собі різний підхід до обраної предметної області.

Першим розглянемо веб-сервіс childdevelop.com.ua та чому в нього така ідентифікація. Цей ресурс надає такі можливості та матеріали:

					ДП.КН 22.481.32.000	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

- практичні завдання різного профілю;
- можливості генерувати власні унікальні завдання використовуючи вже готовий шаблон та конструктор;
- шаблони дипломів для дітей;
- готові інструкції наочних експериментів та програм для творчості;
- тематичні статі для ознайомлення з предметною областю для покращення власного розуміння можливостей застосування матеріалів сервісу;
- навчальні програми на повноцінні заняття.

Отже розглянемо функціонал та подані матеріали поближче та з наглядною демонстрацією контенту. Практичні завдання це файли в форматі pdf різної тематики. Деякі матеріали можна завантажити безкоштовно після реєстрації на ресурсі, що є однозначно позитивним моментом, но якщо захочеться користуватись цим на постійній основі потрібно оплатити відповідні послуги та доступи до матеріалів. Практичні завдання також можна відфільтрувати та знайти матеріали не лише для дошкільнят, а й для інших рівнів учнів. На рисунку 1.5 зображено вибірку деяких практичних завдань, що пропонуються для дошкільнят.

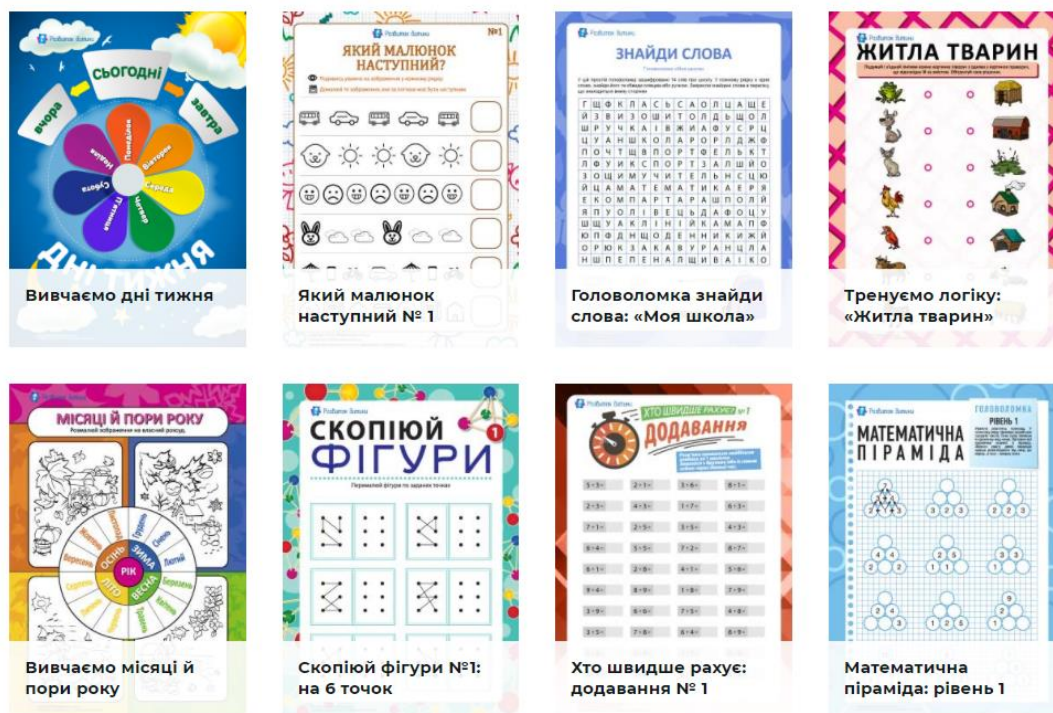


Рисунок 1.5 – Практичні завдання для дошкільнят

					ДП.КН 22.481.32.000	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Генератори завдань представляють собою деякі популярні ігри чи головоломки, які можна створювати по шаблону та користуватись для навчання учнів. На рисунку 1.6 зображені деякі з них.

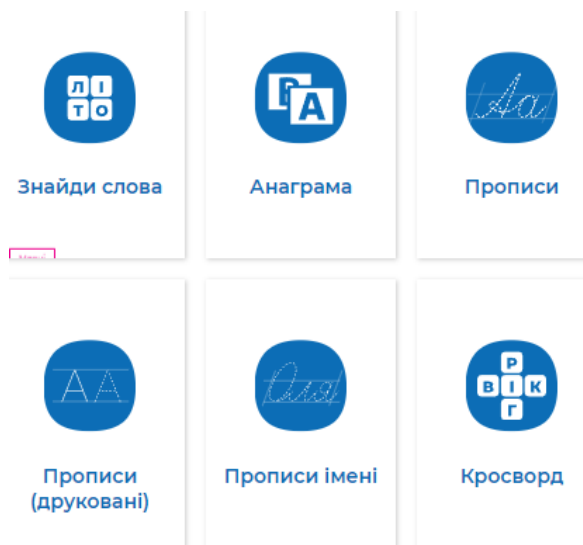


Рисунок 1.6 – Генератори завдань по шаблону

Генератори таких завдань не призначені для рівня дошкільнят, такі досить просунуті завдання зможуть вирішувати учні молодших класів. Хоча для дошкільнят також знайдеться декілька корисних генераторів. Приклад використання генератора Анаграма зображено на рисунку 1.7.

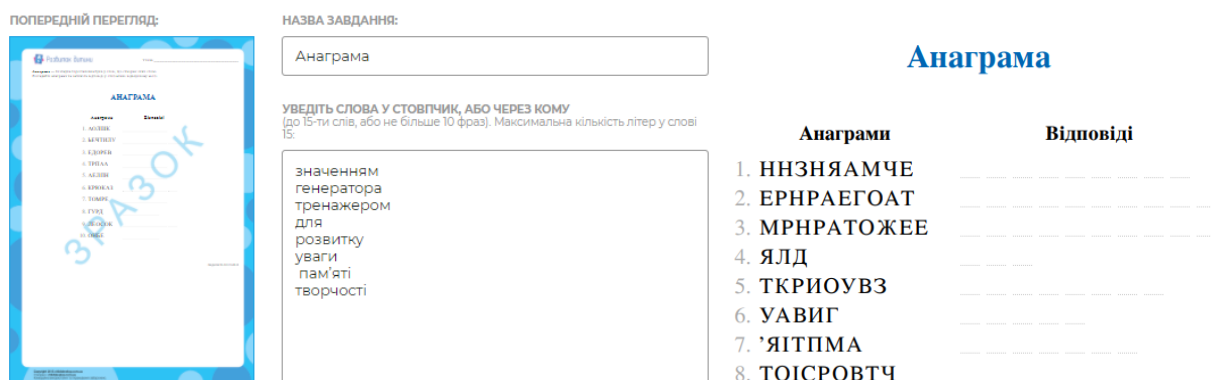


Рисунок 1.7 – Робота та результат генератора Анаграма

Дипломи для дітей це колекція шаблонів для нагородження та заохочення дітей в різних дисциплінах. В такий спосіб можна заохотити дошкільнят працювати старанніше. Вигляд колекції пропонувананих до завантаження шаблонних нагород зображено на рисунку 1.8.

					ДП.КН 22.481.32.000	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.8 – Шаблонні нагороди та грамоти

Розділ експериментів та творчості це інструкції та описи різних дослідів, що можуть пояснити дошкільнятам природу та роботу якогось непростого для них явища більш доступно та наочно, що однозначно сприятиме засвоєнню матеріалу. Цей розділ найкраще дає зрозуміти чому це саме веб-сервіс.[7]

Цей ресурс надає матеріали для вихователів та вчителів, які вже самостійно мають обробляти отримані матеріали та застосовувати їх на своїй практиці зрозуміло та доступно навіть для дошкільнят. Список декількох експериментів зображена на рисунку 1.9.

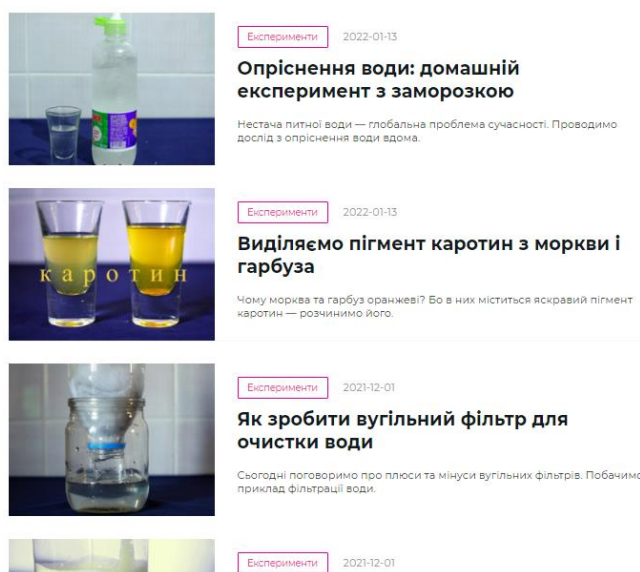


Рисунок 1.9 – Запропоновані експерименти для проведення

Навчальні програми самою назвою дають зрозуміти, що це комплексні збірники конкретної теми або галузі, що пропонується для вивчення дошкільнятам та іншим учням. Кожна окрема тема містить відповідну кількість матеріалів у вигляді файлів для скачування, кожен таку тему можна розбити на декілька занять та поступово занурювати учнів у тему. Різноманіття тем та вигляд матеріалів конкретної теми зображено на рисунку 1.10.

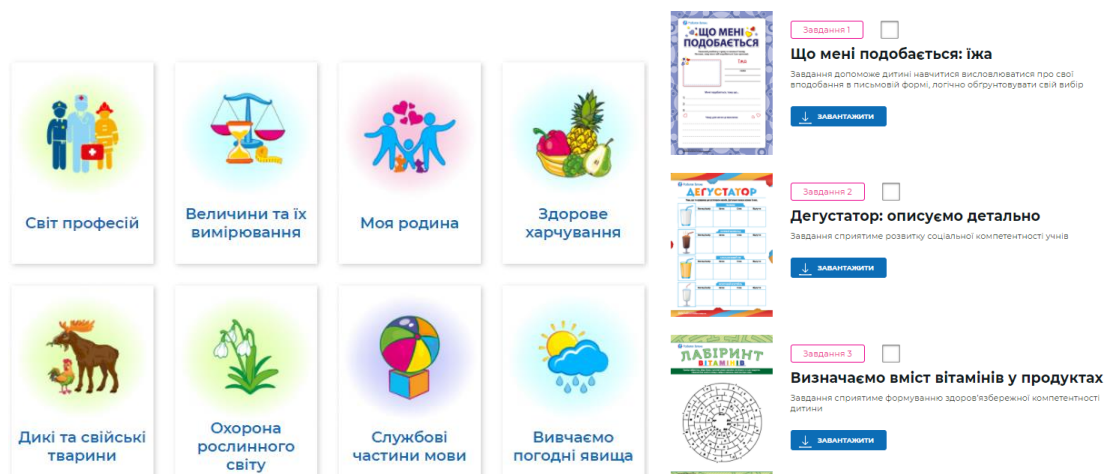


Рисунок 1.10 – Різноманіття навчальних програм

Далі розглядатимемо дуже насичений та функціонально переповнений веб-сайт logiclike.com. Одразу ж в очі кидають різноманіття кольорів, привабливий інтерфейс, та велика кількість можливостей з якими учень дошкільної підготовки самостійно не розбереться. Сайт призначений переважно для 2 класу та вище, но також знайдеться чимало матеріалів спеціально для дошкільнят і з допомогою вихователів чи батьків виникає можливість їх використовувати учнем дошкільної підготовки. Весь сайт це одна велика гра з різними міні-іграми, тестами та квестами. Для аналізування ресурсу будуть виділено та описано основний функціонал, що може бути використаний саме дошкільнятами.

Завдання для дошкільнят містять низку різнопланових тестувань. Застосовуються та покращуються такі навички: логіка, увага, математичні здібності, знання фігур цифр та букв, 3D мислення. Вигляд вибору тестування зображено на рисунку 1.11. [8]

					ДП.КН 22.481.32.000	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.11 – Вигляд вибору тестування

Наприклад, якщо обрати тестування логіки, буде відтворено голосове сповіщення про завдання, тобто що саме потрібно виконати. Вигляд сторінки тестування зображено на рисунку 1.12. Також на сторінці присутні елементи керування процесом тестування. Кожне завдання в тестуванні інтерактивне.



Рисунок 1.12 – Процес тестування

Кожен з тестів навчає різноманітним аспектам поступового розуміння складних для дошкільнят речей, як симетрія та асиметрія, вигляд 3D об'єктів з різних ракурсів та інше. Сайт запам'ятовує усі помилки, що здійснювались при тестуванні та пропонує повторно пройти саме той відрізок тесту, на

					ДП.КН 22.481.32.000	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

якому ця помилка відбулась. Вигляд реалізації цієї функції зображено на рисунку 1.13.

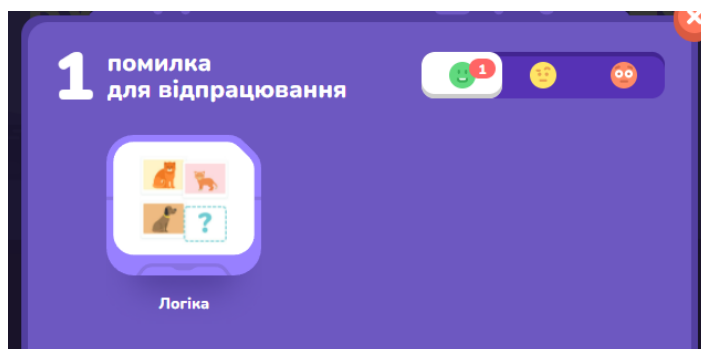


Рисунок 1.13 – Помилки для відпрацювання

За контроль прогресу та нагороди відповідає окрема сторінка, яка збирає статистику роботи з сайтом та при певній кількості виконаних тестувань та завдань видає нагороди, що дає додаткову мотивацію працювати і надалі. Вигляд цієї сторінки зображено на рисунку 1.14.

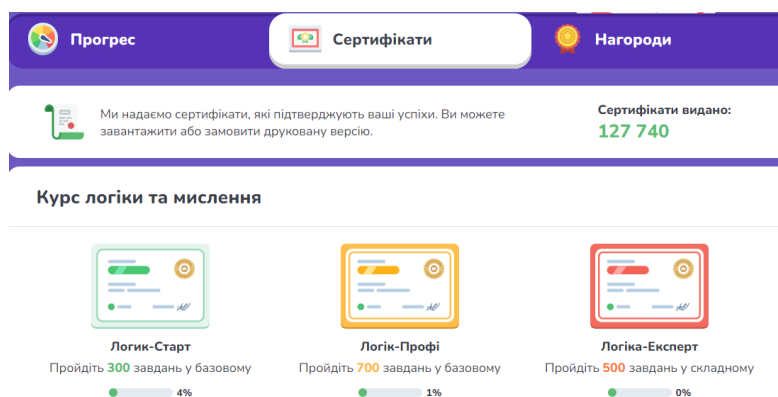


Рисунок 1.14 – Перегляд прогресу та винагороди сайту

Сайт містить ще велику кількість функціональності, як для учнів що будуть працювати та навчатись так і для батьків чи вихователів, що будуть здійснювати контроль прогресу. Ту частину, яка відноситься безпосередньо до предметної області даного дипломного проекту була досліджена та проаналізована.

					ДП.КН 22.481.32.000	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПЛАНУВАННЯ АРХІТЕКТУРНИХ СКЛАДОВИХ

2.1 Аналіз та визначення потрібних UML діаграм

Для проектування невеликого по вмісту та не масштабного за структурою додатка буде цілком достатньо скористатись створенням декількох простих UML діаграм. В особливості для попередньої структуризації основного функціоналу для користувача та програмних зав'язків написаного коду. Звичайно можна виношувати усю візуалізацію в голові та виконувати одразу створення програмного коду, но правильним в розробці програмного продукту завжди є мінімальний процес планування. Тому для підтвердження цього судження виконаємо аналіз цієї теми. Розглянемо більш органічне середовище – розробка програмного продукту командою спеціалістів. Тому проектування з застосуванням діаграм потрібна практика, якої краще притримуватись.

Мінімальним завданням буде побудова діаграми використання та діаграми класів. Розробка програмного забезпечення рідко відбувається ізольовано. Розробникам доводиться співпрацювати з іншими розробниками, дизайнерами, менеджерами проектів і замовниками, щоб створити продукти, якими всі задоволені.

З такою кількістю зацікавлених сторін легко помилитися – а в області, де важливі дрібні деталі, комунікація важлива для підтримки проектів. Навіть розробники іноді опиняються в довгих технічних розмовах, щоб потім зрозуміти, що насправді вони описували зовсім не те саме.

Тому і потрібні діаграми. Хоча дехто може подумати, що час, витрачений на навчання малювати діаграми, — це втрачений час, но на великій дистанції ці знання стають все потрібнішими. Діаграми фокусують розмови та полегшують спілкування, і вони можуть скоротити час, який витрачався б на пояснення та повторне пояснення.

					ДП.КН 22.481.32.000	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

Замість написання 200-сторінкового документу, що буде описувати деяку область функціонування додатка, можна намалювати кілька діаграм.

Уніфікована мова моделювання – це стандарт діаграм, створений для галузі розробки програмного забезпечення, який підтримується Групою управління об'єктами, і є затвердженим стандартом Міжнародної організації зі стандартизації. Дотримання стандарту для складання діаграм усуває непорозуміння щодо стилів діаграм різних людей.

Пояснення того, що всі різні фігури на діаграмі означають для нової аудиторії, може зайняти деякий час. Такі стандарти, як UML вимагають, щоб кожен вивчав лише один стандарт, а не щоразу адаптувався до індивідуального стилю кожного розробника, що буде створювати діаграми по власній схемі.

Є два фактори, які визначають, яку діаграму UML слід використовувати для певної ситуації: аудиторія, яка переглядає діаграму, і тип проекту, над яким працює команда.

Коли менеджери з продуктів представляють клієнту рішення, діаграма класів, ймовірно, не є правильним вибором, вона дуже детальна, і клієнти зазвичай хочуть бачити загальну картину. Замість цього краще використовувати діаграми компонентів, а також діаграми послідовності та варіантів використання для пояснення поведінки.

Діаграми варіантів використання чудово підходять для нетехнічної аудиторії. Особливо під час збору вимог проекту, вони можуть допомогти спланувати сценарії того, як різні користувачі будуть взаємодіяти з системою.

Галузь проекту також визначає, яка діаграма найкраще підходить. Наприклад, автомобільна промисловість не зосереджується на традиційних інтерфейсах користувача на екранах. Тому натомість розробка програмного забезпечення для подібних вбудованих систем зосереджується на стані автомобіля чи іншого потрібного об'єкту.

Різні галузі часто навіть мають підстандарти в UML. Існують стандарти для моделювання бізнес-процесів, моделювання систем і навіть налаштувань

					ДП.КН 22.481.32.000	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

для конкретних організацій, наприклад управління оборонними контрактами, що можна вважати досить специфічною галуззю.

Вибирати між різними діаграмами може бути непросто. Звичайно не можна очікувати, що будуть використовуватися всі діаграми одночасно.

Потрібно зазвичай обирати основне – діаграми класів і діаграми послідовності це найосновніші речі, які малював би будь-хто, хто використовує UML, а також намалювати діаграму класів для сторони розробки безпосередньо програмного коду.

При створенні діаграми вирішити, що залишити, може бути складно. Найкращий спосіб зберегти сферу дії вузькою – це враховувати мету діаграми та її аудиторію. Також рекомендується використовувати програмне забезпечення для моделювання, яке може відфільтрувати частини діаграм, які не стосуються різних сценаріїв.

Знати, яку діаграму використовувати та як нею користуватися, стає легше з більшим досвідом. Але це вміння, в яке варто інвестувати, оскільки створення хороших діаграм є важливим інструментом комунікації. Коли при роботі над програмним продуктом турбують всі деталі, добре пам'ятати, що діаграми існують, щоб допомогти спілкуватися чітко та коротко з командою розробників.

Проаналізувавши суть та важливість діаграм вдалось звернути увагу на ще одну якісну діаграму для візуалізації роботи майбутнього програмного продукту – діаграму послідовності.

2.2 Створення та опис вибраних UML діаграм

Малювання обраних видів діаграм буде проводитись у спеціалізованому для цієї цілі прикладному додатку PowerDesigner. Усі інструменти для створення діаграм відповідають загальноприйнятим стандартам побудови UML діаграм. [4].

					ДП.КН 22.481.32.000	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

Найпростішою для розуміння з запланованого переліку діаграм користувачем є діаграма варіантів використання, документується як UseCase (рисунок 2.1).

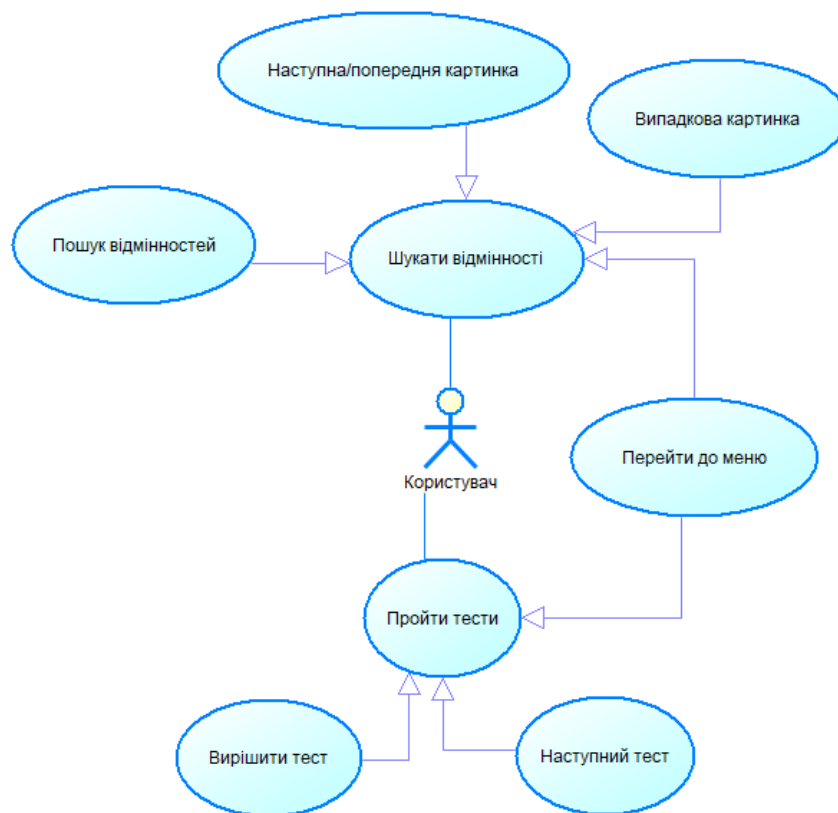


Рисунок 2.1 – Діаграма варіантів використання

По створеній діаграмі використання бачимо, що більшість варіантів є залежними від інших, тобто без виконання певних варіантів вони не можуть бути виконані ні в якому випадку. В діаграмі присутній лише один виконавець(актор) – Користувач, що працює з програмним функціоналом.

Варіанти використання діаграми:

– Шукати відмінності – відкриття одного з двох головних вікон окремої гри. В цьому випадку це початок гри Пошук відмінностей, що дає доступ до ряду інших варіантів використання.

– Пошук відмінностей – взаємодія з картинками для пошуку відмінностей, що при знаходженні відмінності запускає ряд програмних процесів для відображення результатів та майбутніх функцій.

– Наступна/попередня картинка – перемикання між існуючим переліком картинок з відмінностями, що будуть перемикатись на графічному інтерфейсі для користувача.

– Випадкова картинка – також перемикання пари поточних картинок по пошуку відмінностей на випадкову з існуючого списку, що застосовує випадкову незалежну генерацію.

– Пройти тести – відкриття другого основного вікна іншої розвиваючої гри у вигляді ряду тестів. Цей варіант відкриває можливість отримання доступу до інших варіантів використання зв'язаних з поточним.

– Вирішити тест – варіант використання, який фактично вміщає в собі три максимально схожих дії. Відбувається виконання тесту, що відображатиметься на користувацькому інтерфейсі та присутня можливість вибору однієї з трьох відповідей. Функція вибору відповіді ідентична, а от реакція системи може бути різною в залежності від правильності обраної відповіді. Також при правильному виборі запускає системні процеси для коректного функціонування додатка.

– Наступний тест – виконує перехід від поточного тесту, що відображається на наступний попередньо випадково визначений при запуску вікна. Ця дія повністю пропускає тест з втратою можливості на нього відповісти.

– Перейти до меню – повертає користувача на головне меню, що відкривається при запуску програми. Важливий елемент для маршрутизації між вікнами програми не закриваючи її.

Наступна діаграма класів представляє вигляд системи з програмної сторони і більш актуальна для команди розробників, документується як ClassDiagram (рисунок 2.2).

					ДП.КН 22.481.32.000	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

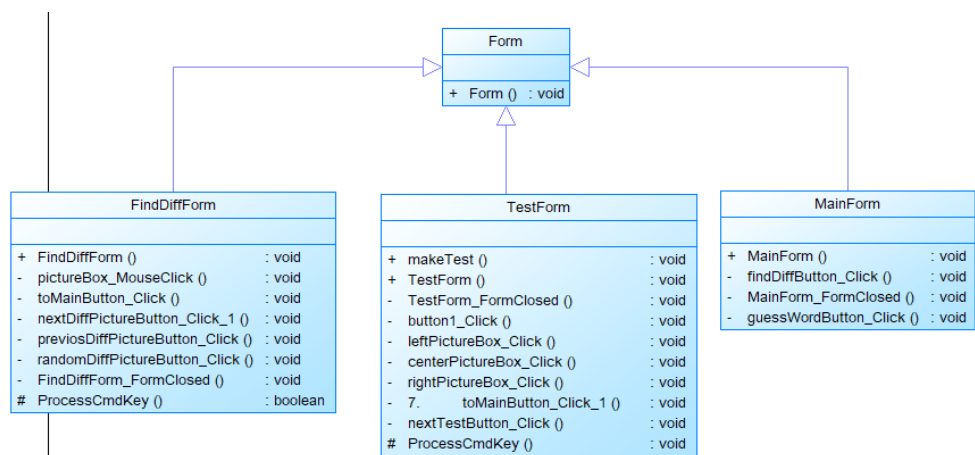


Рисунок 2.2 – Діаграма класів

Для запланованої програми проектування окремих важливих класів не проводилось. Усі класи представлені на діаграмі створенні автоматично засобами технології Windows Form. Дана технологія представляє собою графічний конструктор. При створенні форм за допомогою цього конструктора їх класи створюються самостійно, а розробник вже наповняє його потрібними функціями. Виходячи з цього функції класів є важливішими з сторони унікальної розробки [1].

Класи, їх призначення та потенційний функціонал:

– Form – попередньо існуючий клас Windows Form для успадкування іншими класами, виступають дійовими вікнами додатку. З функціоналу вказано очевидний конструктор Form().

– FindDiffForm – один з трьох класів, що відповідають за існування програмних вікон. Виконує відображення та функціонал форми Пошуку відмінностей. Функціонал класу:

1) FindDiffForm – конструктор класу, відповідає за створення класу та ініціалізацію потрібних параметрів для правильного створення цього класу;

2) pictureBox_MouseClick – обробник натиснення кнопки миші по картинці з відмінностями;

3) toMainButton_Click – повернення до головного меню.4;

4) nextDiffPictureButton_Click_1 – перемикання наступної картинки пошуку відмінностей;

5) previosDiffPictureButton_Click – перемикання попередньої картинки пошуку відмінностей;

6) randomDiffPictureButton_Click – перемикання випадкової картинкаи пошуку відмінностей;

7) indDiffForm_FormClosed – завершення роботи програми;

8) ProcessCmdKey – обробник натиснення клавіш для перемикання картинок пошуку відмінностей;

– TestForm – другий клас програмного вікна проведення тестування. Виконує функціонал форми тестування. Функціонал класу:

1) makeTest – функція, що відповідає за компонування контенту тесту. Викликається програмою у випадках, коли потрібно перейти на наступний тест;

2) TestForm – конструктор класу форми проведення тестування;

3) TestForm_FormClosed – завершення роботи програми;

4) leftPictureBox_Click – обробник вибору лівого варіанту відповіді тесту;

5) centerPictureBox_Click – обробник вибору центрального варіанту відповіді тесту;

6) rightPictureBox_Click – обробник вибору правого варіанту відповіді тесту;

7) toMainButton_Click_1 – вихід в головне меню;

8) nextTestButton_Click – створення наступного тесту;

9) ProcessCmdKey – обробник натиснення клавіш для відображення наступного тесту;

					ДП.КН 22.481.32.000	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

– MainForm – останній клас програмного вікна головного меню. З цього класу відбувається лише виклик інших двох вікон. Функціонал класу:

- 1) MainForm – конструктор класу головного меню.
- 2) findDiffButton_Click – запуск форми пошуку відмінностей.
- 3) MainForm_FormClosed – припинення роботи програми.
- 4) guessWordButton_Click – запуск форми проходження тестування.

Остання діаграма майбутньої послідовності роботи програми. Дана діаграма використовується для уточнення загального виду діаграми Використання. Документується як Sequence Diagram – послідовності зазвичай містять об’єкти, які вступають у взаємодію в рамках визначеного сценарію, повідомлення, якими вони обмінюються, і результати, пов’язані з повідомленнями. Результати, що повертаються до відповідної точки. Діаграма послідовностей зображена на рисунку 2.3. [2]

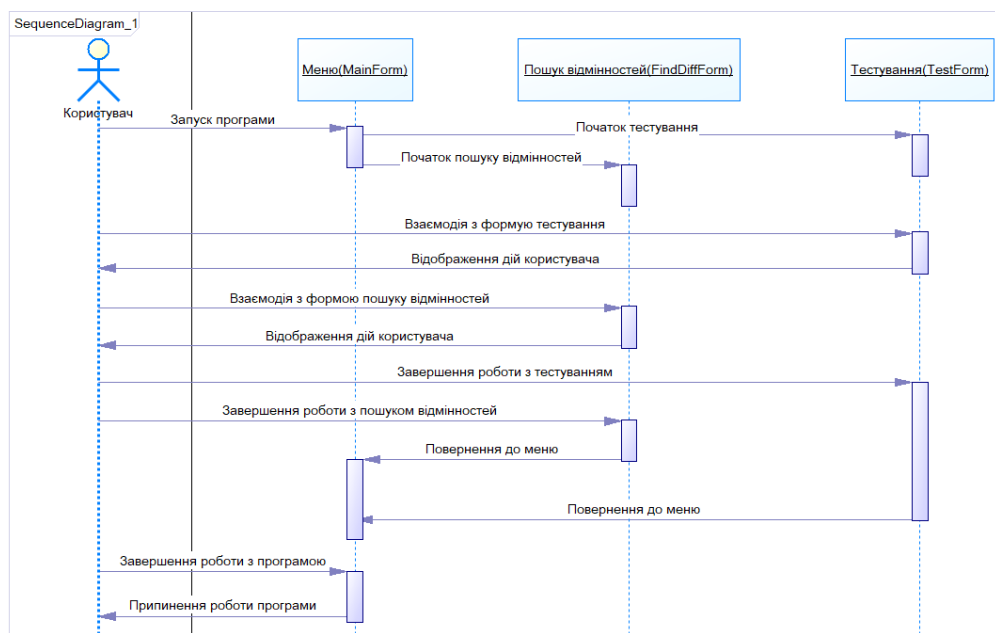


Рисунок 2.3 – Діаграма послідовностей

Діаграма містить чотири об’єкти(Користувач, Меню, пошук відмінностей, Тестування), які обмінюються повідомленнями, що будуть призводити до відповідних послідовних реакцій програми.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СПРОЕКТОВАНОЇ СИСТЕМИ

3.1 Опис середовища та технології написання програмного коду

Для створення коду додатка було обрано комплексне середовище Microsoft Visual Studio 2019. Інтегроване середовище розробки (IDE) – це багатофункціональна програма, яка підтримує багато аспектів розробки програмного забезпечення. Visual Studio – це креативна панель, яку можна використовувати для редагування, налагодження та створення коду, а потім для публікації програми. Крім стандартного редактора та налагоджувача, які надають більшість IDE, Visual Studio включає компілятори, інструменти завершення коду, графічні дизайнери та багато інших функцій для покращення процесу розробки програмного забезпечення.[3]

Внутрішній вигляд та деякі поверхневі функції зображені на рисунку 3.1.

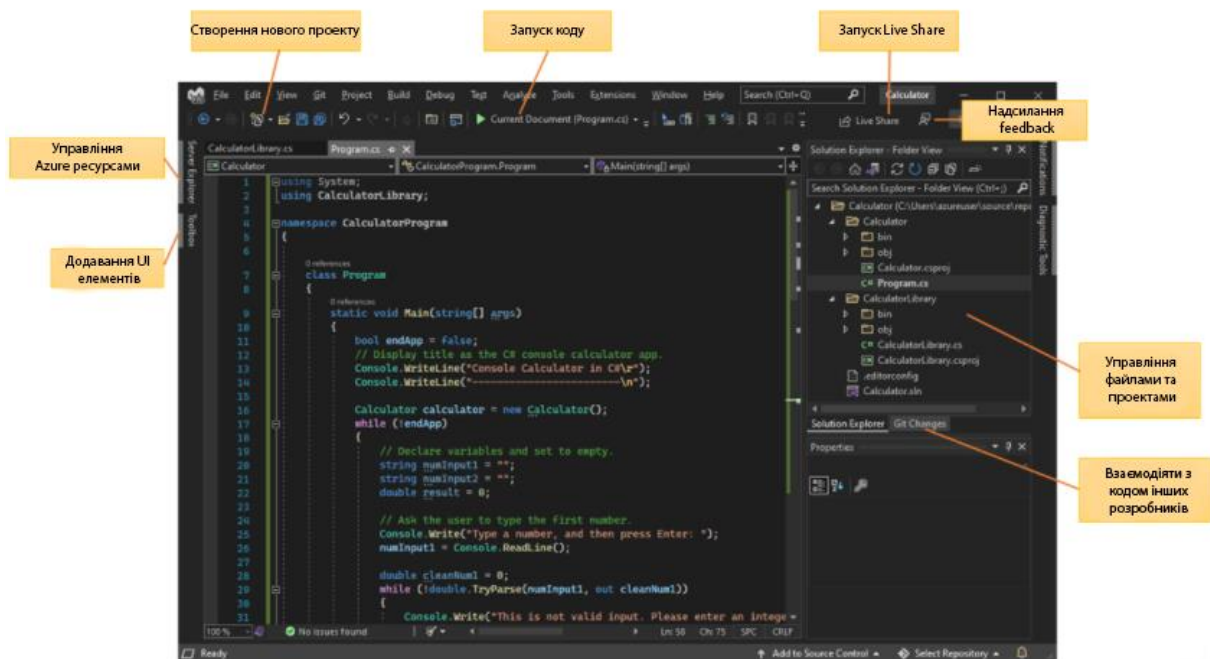


Рисунок 3.1 – Вигляд середовища Visual Studio 2019

Попереднє зображення показує Visual Studio з відкритим проектом, який показує ключові вікна та їх функціональність:

					ДП.КН 22.481.32.000	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

– У провіднику рішень у верхньому правому куті можна переглядати, переміщатися та керувати файлами коду. *Solution Explorer* може допомогти впорядкувати ваш код, згрупувавши файли в рішення та проекти.

– Центральне вікно редактора, де проводиться більшу частину часу роботи з середовищем, відображає вміст файлу. У вікні редактора можна редагувати код або створювати інтерфейс користувача, наприклад вікно з кнопками та текстовими полями, що потрібно для даного проекту.

Популярні функції, що найбільше впливають на продуктивність написання коду чи роботи з програмними файлами:

– Завитки та швидкі дії. Завитки – це хвилясті підкреслення, які сповіщають про помилки або потенційні проблеми у коді під час введення. Ці візуальні підказки допоможуть негайно виправити проблеми, не чекаючи виявлення помилок під час збірки або виконання. Якщо навести курсор на завитку, ви побачите більше інформації про помилку. На лівому полі також може з'явитися лампочка, яка показує *швидкі дії*, які можна виконати, щоб виправити помилку (рисунки 3.2).

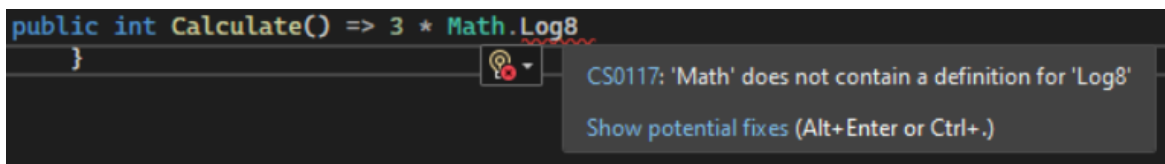


Рисунок 3.2 – Приклад завитків

– Очищення коду. Натиснувши кнопку, можна відформатувати свій код та застосувати будь-які виправлення коду, запропоновані налаштуваннями стилю коду, умовами *.editorconfig* та аналізаторами Roslyn. Очищення коду, наразі доступне лише для коду C#, що допомагає вирішити проблеми у коді, перш ніж він буде перевірений компілятором (рисунки 3.3).[5].

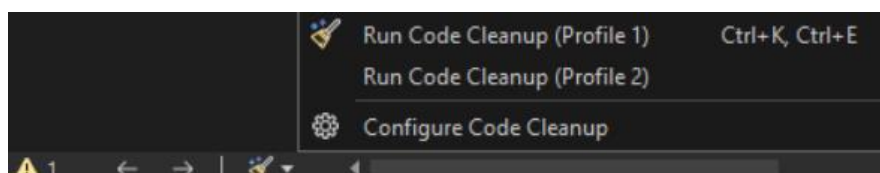


Рисунок 3.3 – Параметр Очищення коду

					ДП.КН 22.481.32.000	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

– Рефакторинг включає такі операції, як інтелектуальне перейменування змінних, видалення одного або кількох рядків коду в новий метод і зміна порядку параметрів методу. Фактично виконує багато роботи самостійно, якщо ти новачок у справі написання якісного та зрозумілого коду 5 (рисунок 3.4).

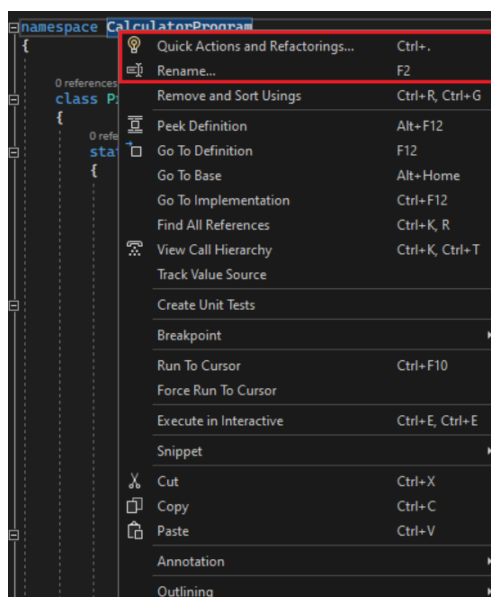


Рисунок 3.4 – Параметр Рефакторингу

– IntelliSense – це набір функцій, які відображають інформацію про код безпосередньо в редакторі і, в деяких випадках, записують невеликі фрагменти коду для розробника. Це як мати базову документацію в редакторі, тож не доведеться шукати інформацію про типи в іншому місці. На рисунку 3.5 показано, як IntelliSense відображає список можливих функцій для типу DateTime:

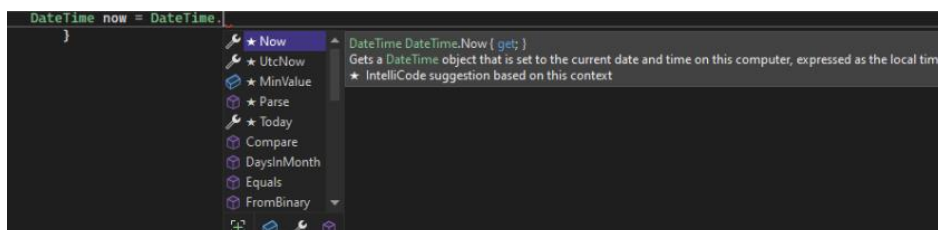


Рисунок 3.5 – IntelliSense на практиці

– Пошук у Visual Studio. Меню, параметри та властивості Visual Studio інколи можуть здаватися заплутаними. Пошук у Visual Studio або Ctrl + Q – це чудовий спосіб швидко знайти функції та код IDE в одному місці (рисунок 3.6).

					ДП.КН 22.481.32.000	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

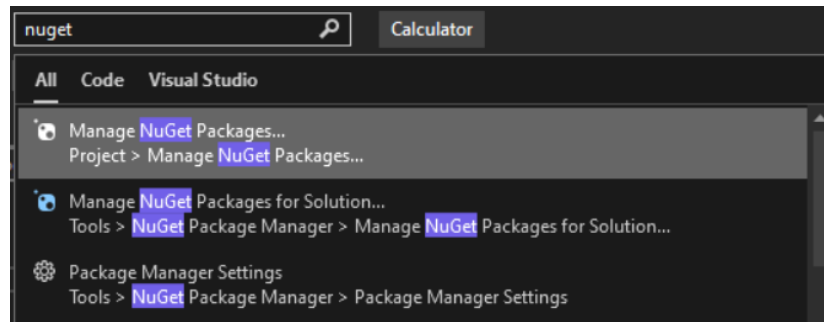


Рисунок 3.6 – Функція пошуку

– Live Share. Можливість спільного редагування та налагоджувати код з іншими в режимі реального часу, незалежно від типу програми чи мови програмування. Можливість миттєво та безпечно поділитися своїм проектом [6]. Також можна ділитися сеансами налагодження, екземплярами терміналів, веб-програмами localhost, голосовими дзвінками тощо

– Ієрархія викликів. У вікні «Ієрархія викликів» показано методи, які викликають вибраний метод. Ця інформація може бути корисною, коли з'являється потреба про зміну чи видалення методу, або коли потрібно відстежити помилку (рисунок 3.7).

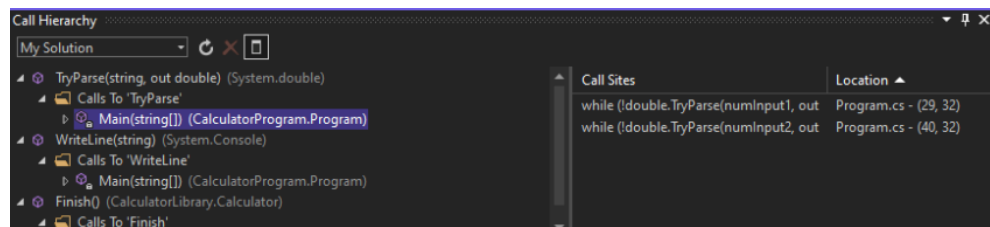


Рисунок 3.7 – Перегляд Ієрархії викликів методу

– CodeLens допомагає знайти посилання на код, зміни коду, пов'язані помилки, робочі елементи, огляди коду та модульні тести, не виходячи з редактора (рисунок 3.8).

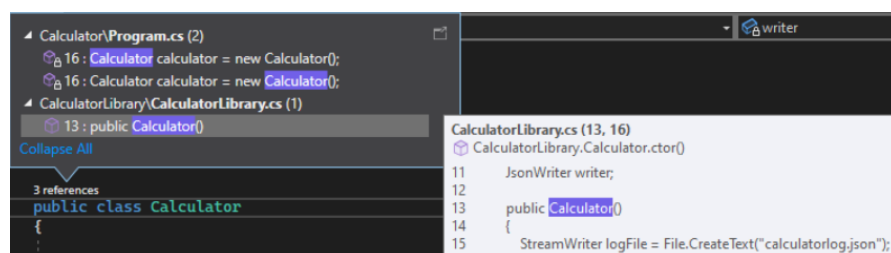


Рисунок 3.8 – Функція CodeLens

					ДП.КН 22.481.32.000	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

– Функція Перейти до визначення переносить безпосередньо до розташування визначення функції або типу (рисунок 3.9).

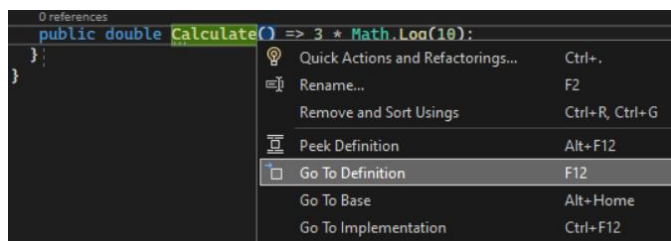


Рисунок 3.9 – Функція Перейти до визначення

– У вікні Peek Definition показано визначення методу або типу без відкриття окремого файлу (рисунок 3.10).

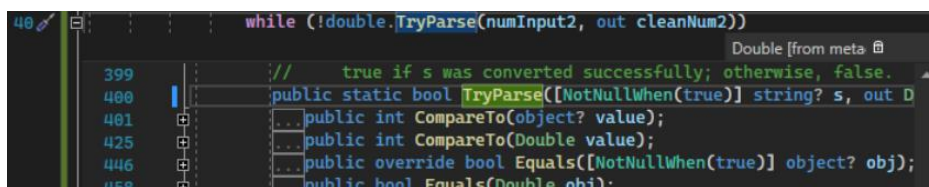


Рисунок 3.10 – Функція Peek Definition

В якості мови програмування обрано C# для реалізації технології Windows Form. Це бібліотека класів графічного інтерфейсу користувача (GUI), яка входить у .Net Framework. Її головна мета – забезпечити спрощений інтерфейс розробки додатків для настільних комп'ютерів, планшетів, ПК. Її також називають WinForms. Програми, розроблені за допомогою Windows Forms або WinForms, відомі як програми, які працюють на настільному комп'ютері. WinForms можна використовувати лише для розробки додатків Windows Forms, а не веб-додатків. Програми WinForms можуть містити різні типи елементів керування, наприклад мітки, списки, підказку тощо.

3.2 Створення програмних рішень проекту

Велика частина роботи це продумування внутрішньої бізнес-логіки додатка, що передбачає програмні рішення взаємодії усіх графічних об'єктів та обробки цифрових даних. В цьому підрозділі будуть зображені найбільш важливі написані програмні рішення, що були придумані з нуля і не являються

					ДП.КН 22.481.32.000	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

частино автоматичної генерації WinForms.

Усі екрани, що відображають графічні елементи з програмної сторони називаються формами. Найменш функціональна початкова форма додатка це форма головного меню. В ній існують лише обробники натискання кнопок без будь-якої особливої бізнес-логіки. У лістингу 3.1 зображено код однієї з двох кнопок, що виконують перехід до форми гри пошуку відмінностей.

Лістинг 3.1 – Код відкриття форми пошуку відмінностей

```
private void findDiffButton_Click(object sender, EventArgs e)
{
    using (FindDiffForm frm = new
FindDiffForm())//Відкриття форми гри пошуку відмінностей
    {
        this.Hide();
        frm.ShowDialog();
    }
}
```

Наступною формою для опису буде форма пошуку відмінностей. Для пошуку відмінностей кожної з списку картинок сформовано блок типу даних Point для повідомлення програмі, де знаходяться відмінності.[7] Кожне значення блоку точок містить два параметра: точку координат по осі X та точку координат по осі Y. Оголошення одного з списку об'єктів у лістингу 3.2.

Лістинг 3.2 – Код оголошення об'єкту Point

```
Point[] difs2 = new Point[] { new Point(636, 382)
                                , new Point(638, 659)
                                , new Point(775, 293)
                                , new Point(518, 88)
                                , new Point(280, 655)
                                , new Point(514, 429)
                                , new Point(78, 683)
                                , new Point(383, 257)
                                , new Point(229, 396)
                                , new Point(579, 587)
                                , new Point(470, 480) };
```

Для форми визначено ряд глобальних змінних для класу. Вони потрібні для динамічної роботи усіх елементів та інформації форми. Перелік цих змінних:

– projectDirectory – місце розміщення директорії проекту для завантаження з іншої директорії проекту пар зображень для пошуку відмінностей.

					ДП.КН 22.481.32.000	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

– `currentPicture` – змінна, що містить інформацію про поточну пару зображень, що знаходиться на формі та відображається для користувача. Потрібна для співставлення картинок та їх переліку відмінностей.

– `diffAmount` – змінна, що приймає значення кількості існуючих відмінностей, що потрібно знайти користувачеві. Містить значення кількості відмінностей відносно поточної пари зображень, що відображаються на формі.

– `currentDiffCount` – змінна, що відповідає за динамічний підрахунок вже знайдених користувачем відмінностей для відображення цього рахунку на графічному інтерфейсі користувача.[8] Допомагає тримати в курсі користувача про кількість відмінностей, які вже було виявлено та водночас дає зрозуміти скільки відмінностей ще потрібно знайти.

При створенні форми у конструкторі заповнюється пустуючі контейнери картинок. Тобто при відкритті форми для користувача вже мають бути встановлені картинки для пошуку відмінностей(лістинг 3.3).

Лістинг 3.3 – Код конструктора форми

```
public FindDiffForm()
{
    InitializeComponent();

    string pictureFilePath1 = projectDirectory +
    "differencePictures\\" + currentPicture.ToString() +
    "_1.png";
    string pictureFilePath2 = projectDirectory +
    "differencePictures\\" + currentPicture.ToString() +
    "_2.png";

    pictureBox1.Image =
    Image.FromFile(@pictureFilePath1);
    pictureBox2.Image =
    Image.FromFile(@pictureFilePath2);
}
```

Для прив'язання окремого об'єкту точок знаходження відмінностей на картинці до відповідної пари зображень при кліку по відображеній картинці відбувається прив'язання конкретного об'єкту точок. Цей функціонал реалізовано логічною структурою `switch case`(лістинг 3.4).

Лістинг 3.4 – Код визначення потрібного об'єкту точок

```
private void pictureBox_MouseClick(object sender,
```

					ДП.КН 22.481.32.000	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

```

MouseEventArgs e)
{
    Point[] difs;
    difs = difs1;
    switch (currentPicture)
    {
        case 1:
            difs = difs1;
            diffAmount = 4;
            break;
        case 2:
            difs = difs2;
            diffAmount = 10;
            break;
    }
}

```

Також у цій самій функції натиснення мишкою по контейнеру зображень присутня і логіка визначення правильності попадання по відмінності. Ця функція реалізована комбінацією логічних структур програмування циклу(for) та перевірки параметрів(if). Принцип роботи наступний – цикл проходить кількість повторень відносно раніше визначеної кількості точок з відмінностями, перевірка чи координати натискання мишки, в яких знаходиться курсор співпадає з один з записів точок відмінностей, в разі позитивного відклику змінюються значення знайдених відмінностей та оновлюються на графічному інтерфейсі.[9] Кінець циклу це малювання графічною функцією еліпса зеленого кольору в відповідній точці координат для візуального визначення знайденої відмінності (лістинг 3.5).

Лістинг 3.5 – Код визначення потрібного об'єкту точок

```

for (int i = 0; i < difs.Length; i++)
{
    if (Math.Abs(e.X - difs[i].X) <= 35 &&
        Math.Abs(e.Y - difs[i].Y) <= 35)
    {
        currentDiffCount++;
        countDiffTextBox.Text =
currentDiffCount.ToString() + "/" + diffAmount.ToString();
        Graphics g1 =
pictureBox1.CreateGraphics();
        g1.DrawEllipse(new Pen(Color.Green, 5),
difs[i].X - 25, difs[i].Y - 25, 50, 50);
        Graphics g2 =
pictureBox2.CreateGraphics();
        g2.DrawEllipse(new Pen(Color.Green, 5),
difs[i].X - 25, difs[i].Y - 25, 50, 50);
    }
}

```

					ДП.КН 22.481.32.000	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

Ряд інших функцій це змінна поточної пари зображень на графічному інтерфейсі для пошуку відмінностей та супровідних параметрів всередині програми. Такий приклад це встановлення випадкової пари зображень. За допомогою випадкової генерації чисел у проміжку кількості існуючих пар зображень у проекті (лістинг 3.6).

Лістинг 3.6 – Код встановлення випадкової пари зображень

```
private void randomDiffPictureButton_Click(object sender,
EventArgs e)
{
    var rand = new Random();

    currentPicture = rand.Next(1, 11);
    string pictureFilePath1 = projectDirectory +
"differencePictures\\" + currentPicture.ToString() +
"_1.png";
    string pictureFilePath2 = projectDirectory +
"differencePictures\\" + currentPicture.ToString() +
"_2.png";

    pictureBox1.Image =
Image.FromFile(@pictureFilePath1);
    pictureBox2.Image =
Image.FromFile(@pictureFilePath2);

    currentDiffCount = 0;
    countDiffTextBox.Text = "";
}
```

Наступна форма з цікавими програмними рішеннями – це форма проходження тестування. Конструктор містить код визначення усіх зображень для тестування(лістинг 3.7). Зображення розподілені на 8 груп п'ятьма парами в загальній кількості існує 40 тестів. Перше зображення з пари визначається цифрою групи та цифрою пари, а друге зображення однієї пари визначається додатковою цифрою пари. Наприклад перше зображення пари 4_2(4 група зображень 2 пара), а друге зображення пари буде визначатись, як 4_2_2.

Щоб не зчитувати назви усіх файлів в визначеній для них папці, вони створюються у вигляді двох списків.[10] Перший список містить усі перші зображення окремої пари, а другий список містить усі другі зображення цієї ж пари. Це все потрібно для майбутньої бізнес-логіки програми. Також виконується виклик функції make_test().

					ДП.КН 22.481.32.000	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг 3.7 – Код конструктора форми

```
public TestForm()
{
    InitializeComponent();

    for (int i = 1; i <= 8; i++)
    {
        for (int j = 1; j <= 5; j++)
            mainPictureList.Add(i.ToString() + "_" +
j.ToString() + ".png");
    }

    for (int i = 1; i <= 8; i++)
    {
        for (int j = 1; j <= 5; j++)
            choicePictureList.Add(i.ToString() + "_"
+ j.ToString() + "_" + j.ToString() + ".png");
    }

    mainPictureList.ToArray();

    makeTest();
}
```

Найбільше складної для розуміння роботи виконує функція `make_test()`. Ця функція містить усю бізнес-логіку створення тесту та зв'язування з правильною відповіддю. Принцип роботи наступний. Починається створення тесту з перевірки, чи ще не всі тести пройдені, про що свідчить кількість елементів в списку перших зображень кожної окремої пари та після проходження одного з тестів чи його пропуску він видаляється з списку. Головним в роботі цієї функції є випадкова генерація кожного тесту, що значить кожен запуск форми буде завжди з унікальним порядком тестів та варіантів відповідей. При випадковому визначені тесту та варіантів відповіді, одна з яких завжди правильна та інші дві випадкові. Також правильний варіант відповіді завжди займатиме випадкове положення з трьох доступних(праве, центральне, ліве). Коли визначені тест та варіанти відповіді, записується яка з них правильна для правильної роботи програми. Код масивний та буде зображений у додатку А.

При виборі одного з трьох варіантів відповіді, якщо вибір неправильний його зображення замінюється на червоний хрестик, що повідомляє користувача про неправильність вибору. Користувач може обирати відповіді до тих піп,

					ДП.КН 22.481.32.000	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

поки не буде вибрано правильну [11]. При виборі правильної відповіді вона програмно зараховується та відображається кількість правильних відповідей на графічному інтерфейсі користувача. Цей функціонал зображений у лістингу 3.8.

Лістинг 3.8 – Код вибору відповіді

```

if (whereAnswer == 1)
{
    points++;

    centerPictureBox.Hide();
    rightPictureBox.Hide();

    arrowCenter.Hide();
    arrowRight.Hide();

    pointsTextBox.Text = points.ToString() + "/"
+ testCount.ToString();
}
else
{
    leftPictureBox.Image =
Image.FromFile(projectDirectory + "wrongAnswer.png");
}

```

Перемикання наступного тесту не містить особливої бізнес-логіки, як і усі інші неописані функції. Більшість незгаданого коду буде відображено у додатку А.

3.3 Створення графічного інтерфейсу програми

Графічна складова створюється за допомогою вбудованого конструктора, що містить усі необхідні контейнери та види елементів інтерфейсу користувача. Прикладом послужить готовий графічний інтерфейс форми тестування (рисунок 3.11).

					ДП.КН 22.481.32.000	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

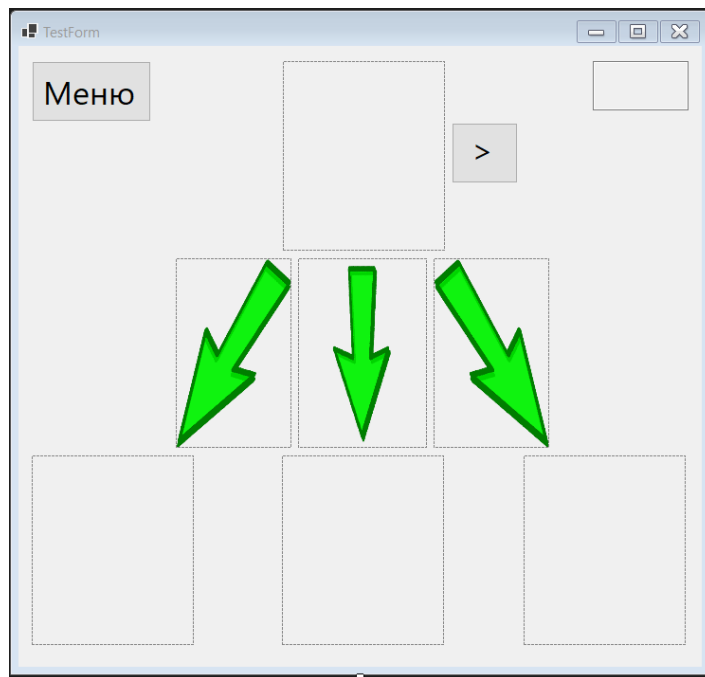


Рисунок 3.11 – Графічне представлення форми тестування

Для побудови дизайну форм використовувались наступні елементи:

- Button – елемент кнопка, відповідає за виконання функціоналу лише по натисканні цієї кнопки.

- TextBox – елемент відображення тексту, використано для цієї цілі саме цей елемент, тому що має властивість multiline(відображення багаторядкового тексту). Виконує лише відображення тексту тому встановлено значення лише для читання(ReadOnly).

- PictureBox – елемент контейнер, що містить відповідне зображення, яке було програмно встановлено. Зручний елемент з властивістю стиснення зображення відносно параметрів ширини та висоти контейнера.

При роботі з конструктором проводилась робота з інструментарієм(переліком всіх можливих елементів). Їх вигляд зображено на рисунку 3.12.

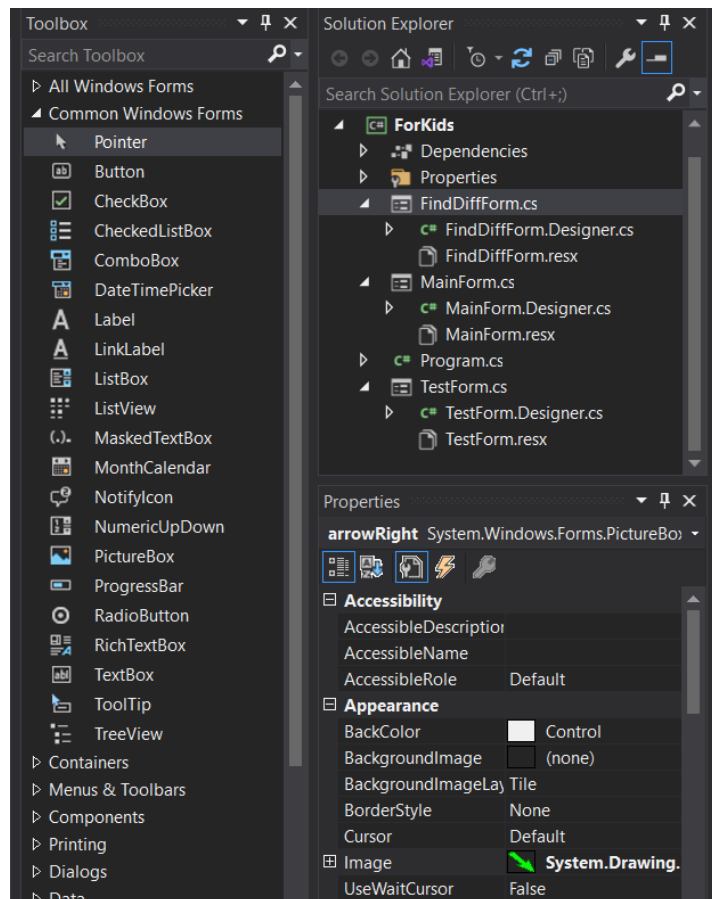


Рисунок 3.12 – Інтерфейс роботи з властивостями графічних елементів

На рисунку 3.12 також зображено ієрархію файлів проекту. Форми, їх програмні та графічні складові.

					ДП.КН 22.481.32.000	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ТЕСТУВАННЯ ДЕСКТОПНОГО ДОДАТКА

Перевірка функціональних складових графічного інтерфейсу та динамічне відображення інформації буде проводитись ручним тестуванням. Запуск програми супроводжується відкриттям першого початкового вікна – вікна головного меню (рисунок 4.1).

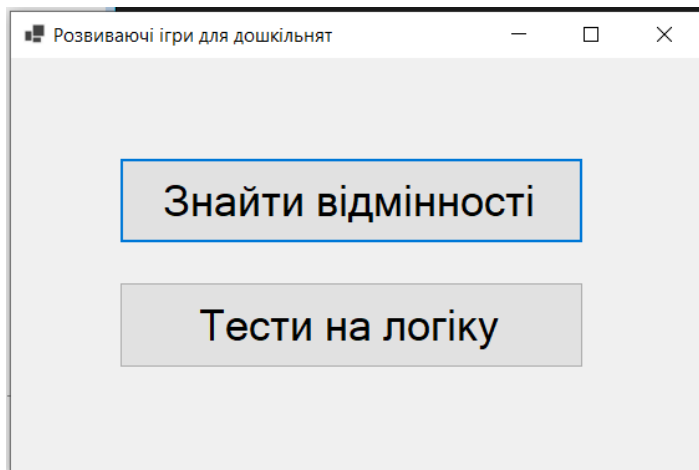


Рисунок 4.1 – Головне меню програми

На вікні відображені дві кнопки, функціональність яких потрібно перевірити. Після натискання кнопки Знайти відмінності буде відкрите вікно розвиваючої гри для дошкільнят, що тренує уважність – пошук відмінностей на двох зображеннях (рисунок 4.2).

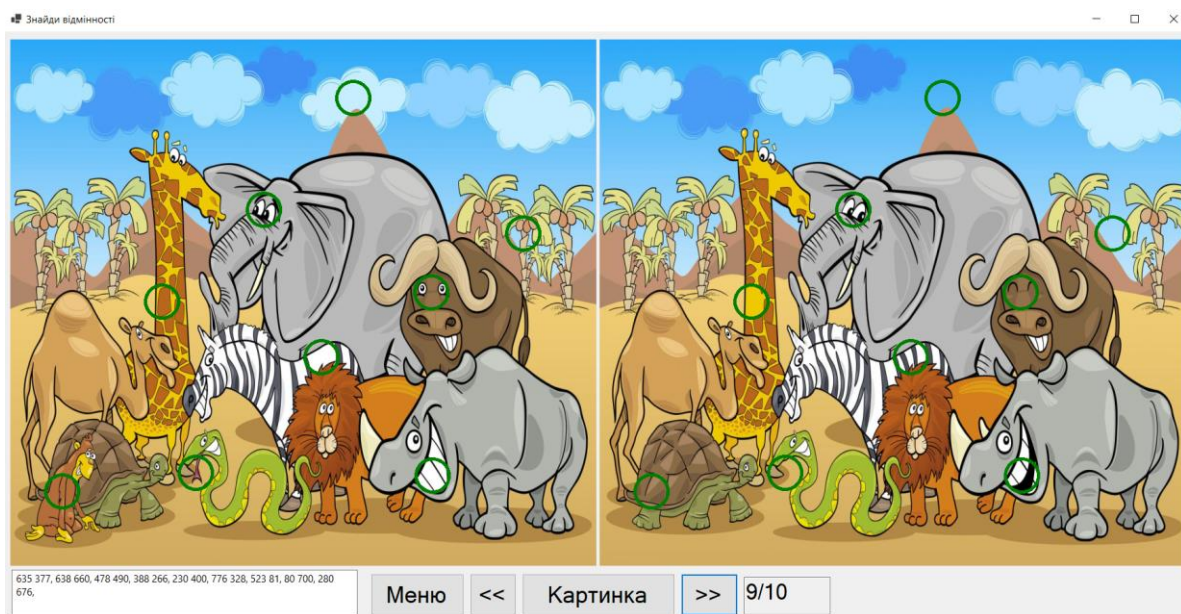


Рисунок 4.2 – Вікно гри пошуку відмінностей

					ДП.КН 22.481.32.000	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

Після відкриття одразу перевірено функціональність гри. При правильному виборі області на одній з зображень, де присутня відмінність відбуваються наступні функції: область позначається зеленим еліпсом; змінюється кількість знайдених відмінностей. Також на час тестування на вікні присутнє текстове поле, що записує координати вибраної області зображення.

Також перевірено маршрутизацію зображень(перехід між зображеннями). Тут перевірено чи відображаються минулі та наступні зображення по натисненні кнопок з стрілочками, ці операції можна виконувати і стрілочками клавіатури. По натисненні кнопки з написом Картинка відображається випадкова пара зображень. Кнопка Меню закриває вікно пошуку відмінностей та відкриває вікно головного меню.

З головного меню (рисунок 4.1) при натисненні на кнопку Тести на логіку відкривається вікно гри проходження тестування для тренування логіки (рисунок 4.3).

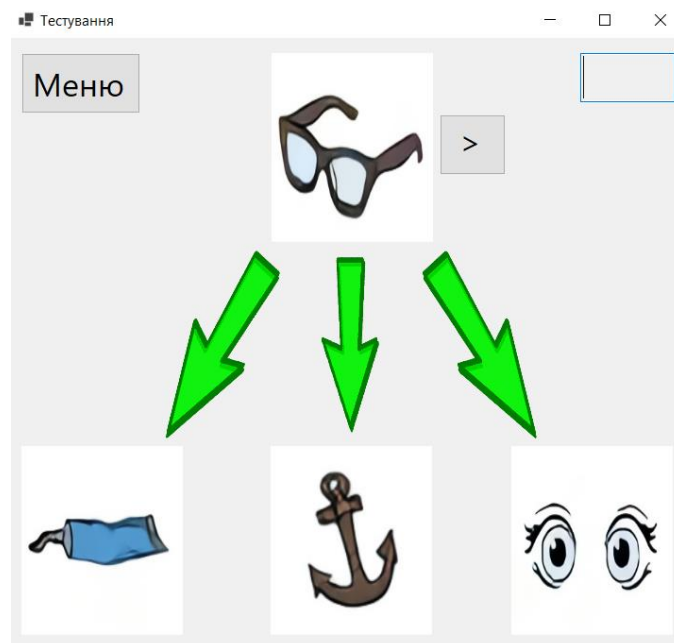


Рисунок 4.3 – Вікно гри тестування

Операції, які було протестовано на вікні тестування:

Правильність роботи функціоналу при виборі неправильного варіанту відповіді, що має відображати замість цієї відповіді червоний хрест (рисунок 4.4).

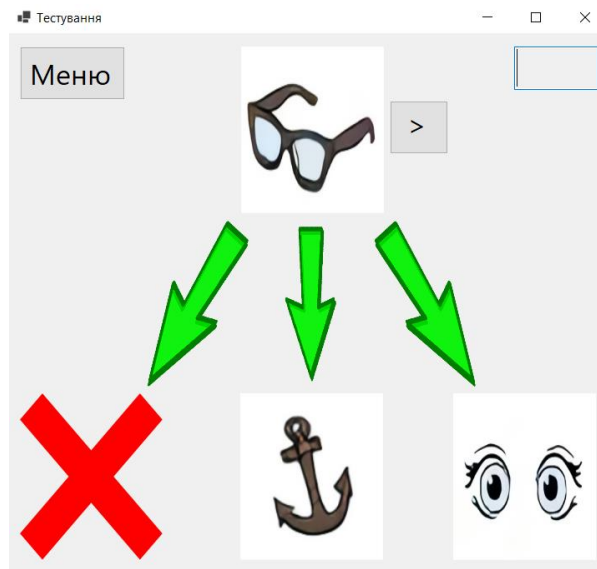


Рисунок 4.4 – Обрання неправильної відповіді

– Правильність відображення правильного варіанту відповіді, при такому випадку інші дві невірні варіанти відповіді зникають (рисунок 4.5).

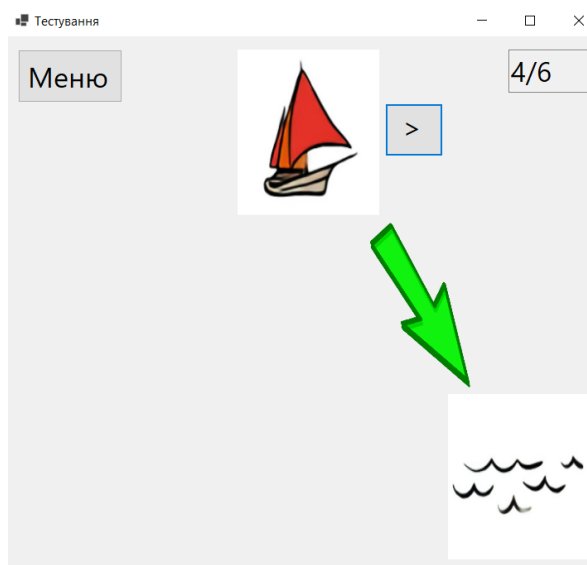


Рисунок 4.5 – Обрання правильної відповіді

- Перемикання наступного тесту.
- Відображення інформації про кількість пройдених тестів та кількість поточних правильних відповідей (рисунок 4.5).
- Робота кнопки повернення до головного меню.

Усі перелічені функції вікна працюють належним чином.

5 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ

5.1 Аналіз ринку

Мета цього техніко-економічного розділу дипломної роботи є здійснення економічних обрахунків, що вказують на деяку економічну ефективність десктопного додатка, прийняття рішення про подальший його розвиток і впровадження або ж недоцільність проведення відповідної розробки. Для проведення даного дослідження необхідно провести перелік розрахунків.

Характеристика програмного продукту:

Десктопний додаток розроблений за допомогою популярних та актуальних технологій WinForms та мови програмування C#. Додаток розроблено для операційної системи Windows 7 та вище, яка буде коректно відображати графічний інтерфейс та функціональність додатка.

Ринок збуту такого специфічного додатка не такий вже й великий та перспективний. Кожен користувач відповідної операційної системи, зможе скористатись розробленим продуктом. Ринок збуту зменшується та складається з тих користувачів, які зацікавлені конкретно в отриманні простого програмного продукту розвиваючих ігор для дошкільнят. Фактично це будуть батьки чи вихователі, які зможуть скачати програму з інтернету.

Характеристики експлуатації програмного продукту:

– Для користування продуктом потрібна операційна система Windows 7 та вище.

– Розроблений продукт є подібною системою вже наявних рішень поставленого завдання і різних її інтерпретаціях(веб-сайти). Розробити та реалізувати абсолютно новий та унікальний продукт на теперішній час не так просто, тому що сучасний ринок десктопних додатків досить великий та різноманітний. Но завдання є більше дослідженням та створенням власного унікально додатка. Реалізація функціоналу та наповнення інтерфейсу відрізняється від усіх аналогій, які зустрічались під час проведення дослідження.

					ДП.КН 22.481.32.000	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

- Можливості для користувача продуктом є можливість вибору однієї з двох розвиваючих ігор з кардинально різними принципами роботи.
- Потенційним замовником продукту вважається користувач операційної системи Windows.
- Основним ринком реалізації продукту будуть різні тематичні інтернет ресурси.
- Продукт створено для некомерційного призначення та більше локального застосування. Також буде знаходитись у вільному доступі абсолютно безкоштовно для усіх бажаючих.
- Головними конкурентами на ринку є усі аналогічні десктопні додатки.
- Продукція конкурентів являє собою краще оформлений дизайн та більш інтерактивний користувацький інтерфейс. За кожним популярним подібним ресурсом закріплена ціла команда і навіть організація, яка займається його підтримкою, дослідженням вимог та побажань цільової аудиторії.

Все таки для даного розділу важливо розглядати продукт, як повноцінний елемент бізнесу.

Розробка настільних додатків – це гарне рішення для бізнес-партнерів, які працюють стабільно. Вони можуть покращити свою ефективність в декілька разів при правильному дослідженню ринку та його можливостей.

Однією з головних вимог для успішного ведення бізнесу в сьогоденні є створення єдиного робочого середовища для співробітників(локального додатка) або створення додатка для просування власних послуг. Безсумнівно, слід приділяти більше уваги управлінню динамічним середовищем десктопних додатків. Якщо компанія знаходиться на стадії затишшя у виробництві, то настав час призначити експертних послуг з розробки десктопних додатків. Компанія з розробки додатків з готовністю обслуговує потреби клієнтів відповідно до їхніх бізнес-потреб.

На ранніх етапах розробки визначення проблем і проблем на високому рівні допоможе сформулювати продукт так, щоб він справді відповідав на загальні проблеми, з якими частіше всього стикаються продавці. Якщо певна функція не

					ДП.КН 22.481.32.000	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

вирішує її, не потрібно вкладати в неї час або розробку, навіть якщо це роблять конкуренти.

Однак у міру того, як програма розвивається з часом, дослідження також має розвиватися.

Замість того, щоб досліджувати лише бізнес проблеми, варто вивчати операційні проблеми, з якими стикаються зацікавлені сторони. Наприклад, відсутність API знижує ефективність. Чи брак документації ускладнює впровадження нових функцій. Чи занадто багато покладається на розробника для налаштування.

Звернення уваги до вирішення функціональних проблем збільшить тривалість зацікавленості саме у поточному програмному продукті користувачів, особливо якщо вони бачать, що ви помітно розвиваєтеся відповідно до їхніх потреб.

Варто привести ще важливі фактори при проведенні дослідження ринку, якими необхідно користуватись при розробці програмних продуктів.

Коли створюється додаток, ведеться вибір з десятків інших проектів та додатків, які потенційно можна створити. Багато ідей додатків ще не знайдені; багато ринків залишаються невикористаними. Далі результати аналізу того, як вибрати правильний проект для розробки.

Найкращі програми зазвичай вирішують проблему. Додаток завжди створюється для певної аудиторії, а це означає, що можна вирішити ту саму проблему для різних аудиторій.

Коли починаєте з великою кількістю ідей додатків, проходить процес усунення. Ось тут на допомогу приходить популярне тлумачення Kill Your Darlings. Замість того, щоб дотримуватися однієї нібито чудової ідеї програми, починаєте дослідження ринку з 10 гідних ідей додатків і дізнаєтеся, яка з них найкраща. Таким чином іноді «вбивається» першочергово обрана ідея.

Важливо те, що буде далі: замість того, щоб шукати докази того, що одна з ідей була найкращою, можна спробувати шукати докази того, що кожна ідея була найгіршою.

					ДП.КН 22.481.32.000	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

Замість того, щоб сподіватися, що обрана найкраща ідея програми переможе, потрібно уважно розглянути кожну з ідей програми. Чи життєздатні вони, технологічно можливі, чи отримають користь від цієї програми визначена аудиторія.

Якщо ні, вбивайте ідею і рухайтесь далі. Як можна самостійно з'ясувати різницю між займатись розробкою ідеї чи продовжити пошук та аналіз ринку.

Порівняння ідей з Google Trends. Коли виконується дослідження ідеї, першим пунктом в цьому завжди є Google Trends. З його допомогою можна побачити відносну популярність пошуків у пошуковій системі Google. Це чудовий інструмент дослідження ринку.

За допомогою цього інструмента можна знайти такі тенденції, як:

- Пошук актуальних тем, пов'язаних із новинами, наприклад, про популярність моделей iPhone.

- Порівняння зростання та занепаду технологій, як-от SMS із VR.

Коротка примітка. Неможливо в такий спосіб знайти, скільки людей переглянуло пошуковий термін у Google Трендах. Натомість це показує, який відсоток людей шукали термін щодо того, коли цей пошуковий термін був найпопулярнішим (тобто від 0% до 100%).

Щоб перевірити свої ідеї цим способом потрібно ввести пошукові запити в Google Trends і виміряти їх відносну популярність.

Після аналізу отриманих в такий спосіб даних, можна, наприклад, побачити наступне:

- Програми-планувальники є найпопулярнішими та мають сезонний характер з серпня по січень.

- Додатки для ведення журналу є найменш популярними, але їх популярність неухильно зростає.

- Програми To-do були найпопулярнішими в 2011-2012 роках, але зараз їх популярність падає.

Інтерпретуючи цей графік, легко помилитися. Працюючи з тенденціями, важливо враховувати напрямки тенденцій, а не дрібні точки даних. Чи варто

					ДП.КН 22.481.32.000	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

вибрати ідею програми-планувальника, оскільки вона популярніша, ніж ідея програми для ведення журналу.

Кілька прикладів:

– Не варто вибирати ідею програми зі списком справ, оскільки її популярність зменшується після свого піку. Це корабель, що тоне

– Також не треба вибирати додаток для цитат, серед інших причин, оскільки він, схоже, вирівнюється. Не сказати точно, можливо, це тимчасово, але не варто на це робити ставку.

– Можна вибрати ідею програми-планувальника, тому що в ній є шаблон: у січні та серпні планувальники популярніші. Skorистатися цією тенденцією, давши аудиторії альтернативну покупку у вигляді нового додатку.

Дослідження ринку з Планувальником ключових слів.

Так, наступний крок дослідження ринку включає інший інструмент Google: Google Keyword Planner.

За допомогою цього інструмента можна знайти детальні дані про обсяг пошукових фраз. Скільки людей виконали певний пошуковий запит. Зазвичай він використовується для реклами в пошукових системах і пошукової оптимізації (SEO), але як інструмент дослідження він настільки ж цінний.

Використовують Планувальник ключових слів Google для двох речей:

– Дізнайтеся, що люди шукають.

– Дізнайтеся, скільки з них шукають конкретний запит.

Чудова перевага Планувальника ключових слів полягає в тому, що не потрібно самостійно придумувати ідеї для пошукових фраз. Він може пропонувати пошукові фрази на основі деяких введених даних.

Перш ніж почати дослідження за допомогою Планувальника ключових слів, потрібно визначити, що саме треба дізнатися дізнатись. Перш ніж створювати додаток, зазвичай потрібно дізнатися наступні пункти:

– Які конкретні пошукові фрази використовують люди, які шукають інформацію, пов'язану з предметною областю проекту.

– Коли вони шукають ці речі, що саме вони знаходять.

– Скільки з них шукають ці фрази.

Коли проведено дослідження за допомогою Планувальника ключових слів, отримано список пошукових фраз і їх середній обсяг пошуку. Можна використовувати це для оптимізації платформи реалізації додатка, а також як список тем для статей, які можуть бути написаними.

Пошукові фрази, які будуть знайдені за допомогою Планувальника ключових слів, – це слова, які використовуються потенційними клієнтами та аудиторією, щоб шукати заплановану програму.

На жаль, Google надає лише діапазони для роботи. Це не абсолютний обсяг пошуку. Натомість Google просто дає базові цифри, з якими можна працювати. На щастя, є хитрість, за допомогою якої можна отримати фактичний обсяг пошуку.

Далі потрібно отримати точніші дані про обсяг пошуку. Ці діапазони нормальні, але треба знати, скільки людей шукає ці терміни. Це також допоможе порівняти обсяг пошукових фраз для різних ідей додатків.

Це можна зробити наступним чином:

- Спочатку в огляді варіантів ключових слів натисніть синю стрілку для кількох ключових слів. Стрілка знаходиться в останньому стовпці з написом Додати до плану. Це помістить ці ключові слова до плану ключових слів.

- Потім натисніть Переглянути план. Це перенесе користувача на нову сторінку. Там буде графік і електронна таблиця. Обов'язково потрібно перейти на вкладку Ключове слово.

- Потім виберемо у меню Тип відповідності -> Усі -> Точна відповідність.

- Нарешті, у верхній частині сторінки встановлюється 1 долар або 1 євро для введення ставки та натисніть Enter.

Такі операції будуть коштувати незначних витрат, но результати можуть бути набагато дорожчими, якщо йдеться про комерційну розробку.

Тепер електронна таблиця оновитися, показуючи точніші дані про кліки та покази порівняно з діапазонами, які були отримані раніше.

Далі ці дані можна використовувати для ранжування ідей додатків за популярністю. Популярна пошукова фраза не обов'язково означає, що це

хороша ідея програми, але вона дає приблизну оцінку попиту програму на основі кількості людей, які шукають певну фразу.

Переходимо до найважливішого етапу дослідження.

Пошук історій, проблем і запитань. Наступний інструмент дасть можливість відійти від холодного інструменту дослідження і зайнятися історіями, проблемами та питаннями.

Емпатія є найціннішим інструментом дослідження ринку.

Завдяки Google Trends і Планувальнику ключових слів тепер відомо, де шукати. Це не означає, що вже є ідея програми-переможця.

Було виключено деякі ідеї, але справжній переможець приходить, коли представляєте свою програму комусь, у кого є проблема, яку вирішує додаток. Потім можна перевірити ідею програми та перевірити, чи є на неї реальний попит.

Наступне, що потрібно зробити, досліджуючи ідею програми – це взяти усі досліджені ключові слова та фрази та почати їх використовувати наступним чином:

- Пошук у Google фрази, що дасть масу досліджень про конкурентів, дописи в блозі, альтернативні продукти тощо.
- Використати ці фрази в соціальних мережах та на форумах, таких як Twitter, Facebook, Reddit, Hacker News, LinkedIn, Product Hunt тощо.
- Використати ці фрази на різних платформах реалізації майбутнього продукту, а також шукав альтернативні фізичні продукти в Інтернеті.

5.2 Розрахунок витрат на проектування

Витрати на створення продукту $K_{ПО}$ включають витрати на заробітну плату виконавця програми $З_{ЗП}$ і витрат машинного часу, необхідного на налагодження програми на ЕОМ

$$K_{ПО} = З_{ЗП} + З_{МВ} \quad (5.1)$$

Заробітна плата виконавців визначається за формулою:

					ДП.КН 22.481.32.000	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

$$З_{зп} = t \times C_{пп} \quad (5.2)$$

де: t – загальна трудомісткість, людино-годин;

$C_{пп}$ – середня годинна заробітна плата програміста, грн/година;

$$C_{пп} = 35,0 \text{ грн/год.}$$

$$\text{Отже: } З_{зп} = 1205.87 \times 35 = 42175 \text{ грн.}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$З_{мв} = t_{отл} \times C_{мч}, \quad (5.3)$$

де $t_{отл}$ – трудомісткість налагодження програми на ЕОМ, год.

$C_{мч}$ – вартість машино-години ЕОМ, грн/год (приблизно 20грн/год).

Звідси:

$$З_{мв} = 728,25 \times 20 = 14565 \text{ грн.}$$

З отриманих даних витрати на створення програмного забезпечення складатимуть:

$$K_{по} = 42175 + 14565 = 56740 \text{ грн.}$$

Визначені в такий спосіб витрати на створення програмного забезпечення є частиною одноразових капітальних витрат на створення АСУП.

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \times F_p} \text{ міс.}, \quad (5.4)$$

де B_k – число виконавців;

F_p – місячний фонд робочого часу (при 40 годинному робочому тижні $F_p = 176$ годин).

Звідси:

$$T = \frac{1205.87}{(1 \times 176)} \approx 6 \text{ міс.}, \quad (5.5)$$

Таким чином, період розробки програми складає приблизно 6 місяців.

					ДП.КН 22.481.32.000	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

5.3 Обґрунтування необхідності розробки

Основна мета економічної ефективності – дати фінансову оцінку передбачуваних витрат та одержуваного корисного результату, а також оцінити прибутковість проекту і, в кінцевому підсумку, економічну доцільність його розробки та впровадження.

Нова техніка, технологія, засоби автоматизації, що розробляються і впроваджуються у виробництво, повинні приносити певний корисний результат – ефект. Ефект може проявлятися у поліпшенні умов праці працюючих (соціальний), в зниженні шкідливого впливу виробництва на навколишнє середовище (екологічний), та в економії витрат підприємства на виробництво продукції та збільшенні його прибутку (економічний).

Абсолютна величина економічного ефекту без співставлення його з витратами підприємства не дозволяє однозначно оцінити, наскільки вдалим виявився відповідний проект. Таку оцінку дають показники економічної ефективності (прибутковості) проекту.

При впровадженні інвестиційного проекту підприємство несе разові витрати, пов'язані з розробкою проекту, а також з придбанням і налагодженням необхідного обладнання, засобів програмного забезпечення і таке інше.

Такі разові витрати називають капітальними витратами або інвестиціями. При використанні інновацій підприємство отримує певний ефект, що зазвичай виражається приростом прибутку. Величина щорічного прибутку, додатково одержуваного підприємством за рахунок впровадження інвестиційного проекту, повинна бути достатньо високою у порівнянні з капітальними витратами та у порівнянні з іншими можливими варіантами вкладення коштів у розвиток виробництва.

Головною метою проекту є отримання деякої кількості аудиторії та числа користувачів, тому економічна ефективність у продукту може враховуватись лише у випадку монетизації (поширення реклами), а на передній план виходить соціальна ефективність від впровадження, що характеризує покращення навичок у користувачів та розробників програмного забезпечення.

					ДП.КН 22.481.32.000	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

Витрати на створення проекту для статистичного аналізу та дослідження ефективності обробки складають 56740 грн. Період розробки програми складає приблизно 6 місяців.

Розробка десктопного додатка повинна починатися з визначення моделі його створення. В результаті проведеного дослідження було систематизовано основні типові етапи створення розважальної гри для дошкільнят. У великих фірмах для створення та підтримки функціонування сайту створюється ціла команда розробників, дизайнерів, копірайтерів та інших спеціалістів, які працюють над даним проектом місяцями, а іноді навіть і більше. Проте не завжди у компанії є ресурси для цього, особливо якщо вона є досить молодою.

					ДП.КН 22.481.32.000	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

При виконанні комплексних робіт під час створення дипломного проекту було проведено дослідження та вивчення в різних підрозділах розробки повноцінного десктопного додатка розвиваючої гри для дошкільнят. Поставлені початкові вимоги та завдання по програмному функціоналу та графічному інтерфейсу були збережені та успішно реалізовані. Створений програмний додаток повністю реалізований та готовий до експлуатації.

При проведенні перших етапів проектування системи для подальшого його просування було проведено вивчення важливості побудови UML діаграм. Вибрано необхідні для майбутньої системи типи діаграм та успішно побудовано з супроводжуючою їх документацією та поясненням.

Десктопний додаток створений згідно опису у етапі проектування засобами, що були дослідженні під час початку етапу реалізації. Для розробки програмного коду додатка використано об'єктно-орієнтовану поширену мову програмування C# у середовищі IDE Microsoft Visual Studio. Програма виконана в рамках технології розробки десктопних додатків WindowsForms.

На фінальному етапі розробки будь-якого програмного продукту виконується тестування системи. При проведенні ручного тестування виявлені дефекти були успішно виправлені та затверджено правильність роботи додатка. Завершивши фінальний етап тестування представлено повноцінний готовий до експлуатації десктопний додаток розвиваючу гру для дошкільнят.

					ДП.КН 22.481.32.000	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Чак Сфер. Head First. Вивчаємо C#. Діалектика, 2015. 510 с.
2. Джон Пол Мюллер. C# для чайників. М.: Пітер, 2019. 608 с.
3. Білл Вагнер. Мова програмування C# 7 і платформи .NET і .NET Core. Вільямс, 2016. 240 с.
4. Джозеф Албахари. C# 9.0. Справочник. Повний опис мови. Вільямс, 2016. 1056 с.
5. WindowsForms. *Вікіпедія - вільна енциклопедія*: веб-сайт. URL: https://ru.wikipedia.org/wiki/Windows_Forms (дата звернення: 10.03.2022).
6. Microsoft_Visual_Studio. *Вікіпедія - вільна енциклопедія*: веб-сайт. https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio (дата звернення: 11.03.2022).
7. Дизайнерский подход к WinForms. *Habrahabr*: веб-сайт. URL: <https://habr.com/ru/sandbox/71682/> (дата звернення: 15.03.2022).
8. C Sharp. *Вікіпедія - вільна енциклопедія*: веб-сайт. URL: https://uk.wikipedia.org/wiki/C_Sharp (дата звернення: 17.03.2022).
9. Learning UML Diagrams. *Builton*: веб-сайт. URL: <https://gitscm.com> (дата звернення: 17.03.2022).
10. Introduction to C# Windows Forms Applications. *GeeksforGeeks*: веб-сайт. URL: <https://www.geeksforgeeks.org/introduction-to-c-sharp-windows-forms-applications/> (дата звернення: 20.03.2022).
11. Windows Form Application C#. *C-sharpcorner*: веб-сайт. URL: <https://www.c-sharpcorner.com/article/windows-form-application-c-sharp/> (дата звернення: 25.03.2022).

					ДП.КН 22.481.32.000	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

Додаток А

Програмний код

Лістинг А1 – Код форми FindDiff

```
public partial class FindDiffForm : Form
{
    Point[] difs1 = new Point[] { new Point(68, 293)
                                   , new Point(90, 514)
                                   , new Point(685, 541)
                                   , new Point(718, 149) };
    Point[] difs2 = new Point[] { new Point(636, 382)
                                   , new Point(638, 659)
                                   , new Point(775, 293)
                                   , new Point(518, 88)
                                   , new Point(280, 655)
                                   , new Point(514, 429)
                                   , new Point(78, 683)
                                   , new Point(383, 257)
                                   , new Point(229, 396)
                                   , new Point(579, 587)
                                   , new Point(470, 480) };

    string projectDirectory =
AppDomain.CurrentDomain.BaseDirectory;
    int currentPicture = 1;
    int diffAmount;
    int currentDiffCount = 0;
    public FindDiffForm()
    {
        InitializeComponent();
        string pictureFilePath1 = projectDirectory +
"differencePictures\\" + currentPicture.ToString() + "_1.png";
        string pictureFilePath2 = projectDirectory +
"differencePictures\\" + currentPicture.ToString() + "_2.png";
        pictureBox1.Image = Image.FromFile(@pictureFilePath1);
        pictureBox2.Image = Image.FromFile(@pictureFilePath2);
    }
    private void pictureBox_MouseClick(object sender,
MouseEventArgs e)
    {
        Point[] difs;
        difs = difs1;
        switch (currentPicture)
        {
            case 1:
                difs = difs1;
                diffAmount = 4;
                break;
            case 2:
                difs = difs2;
```

					ДП.КН 22.481.32.000	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        diffAmount = 10;
        break;
    case 3:
        difs = difs3;
        diffAmount = 10;
        break;
    case 4:
        difs = difs4;
        diffAmount = 10;
        break;
    case 5:
        difs = difs5;
        diffAmount = 10;
        break;
    case 6:
        difs = difs6;
        diffAmount = 10;
        break;
    case 7:
        difs = difs7;
        diffAmount = 10;
        break;
    case 8:
        difs = difs8;
        diffAmount = 5;
        break;
    case 9:
        difs = difs9;
        diffAmount = 10;
        break;
    case 10:
        difs = difs10;
        diffAmount = 10;
        break;
    default:
        difs = difs1;
        diffAmount = 4;
        break;
}
for (int i = 0; i < difs.Length; i++)
{
    if (Math.Abs(e.X - difs[i].X) <= 35 &&
Math.Abs(e.Y - difs[i].Y) <= 35)
    {
        currentDiffCount++;
        countDiffTextBox.Text =
currentDiffCount.ToString() + "/" + diffAmount.ToString();
        Graphics g1 = pictureBox1.CreateGraphics();
        g1.DrawEllipse(new Pen(Color.Green, 5),
difs[i].X - 25, difs[i].Y - 25, 50, 50);
        Graphics g2 = pictureBox2.CreateGraphics();
        g2.DrawEllipse(new Pen(Color.Green, 5),
difs[i].X - 25, difs[i].Y - 25, 50, 50);
    }
}

```

					ДП.КН 22.481.32.000	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        }
    }
    private void toMainButton_Click(object sender, EventArgs
e)
    {
        using (MainForm frm = new MainForm())
        {
            this.Hide();
            frm.ShowDialog();
        }
    }
    private void nextDiffPictureButton_Click_1(object sender,
EventArgs e)
    {
        if (currentPicture < 10 && currentPicture > 0)
        {
            currentPicture += 1;
            string pictureFilePath1 = projectDirectory +
"differencePictures\\" + currentPicture.ToString() + "_1.png";
            string pictureFilePath2 = projectDirectory +
"differencePictures\\" + currentPicture.ToString() + "_2.png";
            pictureBox1.Image =
Image.FromFile(@pictureFilePath1);
            pictureBox2.Image =
Image.FromFile(@pictureFilePath2);
        }
        currentDiffCount = 0;
        countDiffTextBox.Text = "";
    }
    private void previosDiffPictureButton_Click(object sender,
EventArgs e)
    {
        if (currentPicture <= 10 && currentPicture > 1)
        {
            currentPicture -= 1;
            string pictureFilePath1 = projectDirectory +
"differencePictures\\" + currentPicture.ToString() + "_1.png";
            string pictureFilePath2 = projectDirectory +
"differencePictures\\" + currentPicture.ToString() + "_2.png";
            pictureBox1.Image =
Image.FromFile(@pictureFilePath1);
            pictureBox2.Image =
Image.FromFile(@pictureFilePath2);
        }
        currentDiffCount = 0;
        countDiffTextBox.Text = "";
    }
    private void randomDiffPictureButton_Click(object sender,
EventArgs e)
    {
        var rand = new Random();
        currentPicture = rand.Next(1, 11);
        string pictureFilePath1 = projectDirectory +
"differencePictures\\" + currentPicture.ToString() + "_1.png";

```

					ДП.КН 22.481.32.000	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        string pictureFilePath2 = projectDirectory +
"differencePictures\\" + currentPicture.ToString() + "_2.png";
        pictureBox1.Image = Image.FromFile(@pictureFilePath1);
        pictureBox2.Image = Image.FromFile(@pictureFilePath2);
        currentDiffCount = 0;
        countDiffTextBox.Text = "";
    }
    private void FindDiffForm_FormClosed(object sender,
FormClosedEventArgs
    {
        Application.Exit();
    }
    protected override bool ProcessCmdKey(ref Message msg,
Keys keyData)
    {
        //capture right arrow key
        if (keyData == Keys.Right)
        {
            if (currentPicture < 10 && currentPicture > 0)
            {
                currentPicture += 1;
                string pictureFilePath1 = projectDirectory +
"differencePictures\\" + currentPicture.ToString() + "_1.png";
                string pictureFilePath2 = projectDirectory +
"differencePictures\\" + currentPicture.ToString() + "_2.png";
                pictureBox1.Image =
Image.FromFile(@pictureFilePath1);
                pictureBox2.Image =
Image.FromFile(@pictureFilePath2);
            }
            currentDiffCount = 0;
            countDiffTextBox.Text = "";
        }
        //capture left arrow key
        if (keyData == Keys.Left)
        {
            if (currentPicture <= 10 && currentPicture > 1)
            {
                currentPicture -= 1;
                string pictureFilePath1 = projectDirectory +
"differencePictures\\" + currentPicture.ToString() + "_1.png";
                string pictureFilePath2 = projectDirectory +
"differencePictures\\" + currentPicture.ToString() + "_2.png";

                pictureBox1.Image =
Image.FromFile(@pictureFilePath1);
                pictureBox2.Image =
Image.FromFile(@pictureFilePath2);
            }
            currentDiffCount = 0;
            countDiffTextBox.Text = "";
        }
        //capture space arrow key
        if (keyData == Keys.Space)

```

					ДП.КН 22.481.32.000	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        {
            var rand = new Random();
            currentPicture = rand.Next(1, 11);
            string pictureFilePath1 = projectDirectory +
            "differencePictures\\" + currentPicture.ToString() + "_1.png";
            string pictureFilePath2 = projectDirectory +
            "differencePictures\\" + currentPicture.ToString() + "_2.png";
            pictureBox1.Image =
            Image.FromFile(@pictureFilePath1);
            pictureBox2.Image =
            Image.FromFile(@pictureFilePath2);
            currentDiffCount = 0;
            countDiffTextBox.Text = "";
        }
        return base.ProcessCmdKey(ref msg, keyData);
    }
}

```

Лістинг А2 – Код форми TestForm

```

public partial class TestForm : Form
{
    string projectDirectory =
    AppDomain.CurrentDomain.BaseDirectory;
    List<string> mainPictureList = new List<string>();
    List<string> choicePictureList = new List<string>();
    int whereAnswer;
    int points = 0;
    int testCount = 1;
    void makeTest()
    {
        if (mainPictureList.Count() != 0)
        {
            var rand = new Random();
            //MessageBox.Show(mainPictureList.Count().ToString());
            int currentPicturePoss = rand.Next(0,
            mainPictureList.Count());
            int firstPicturePoss = rand.Next(0, 40);
            int secondPicturePoss = rand.Next(0, 40);
            //усунення випадків повторення картинок у
            варіантах відповідей, коли масив головних картинок заповнений
            while (firstPicturePoss == currentPicturePoss)
            {
                firstPicturePoss = rand.Next(0, 40);
            }
            while (secondPicturePoss == currentPicturePoss ||
            secondPicturePoss == firstPicturePoss)
            {
                secondPicturePoss = rand.Next(0, 40);
            }
            string mainPictureFilePath = projectDirectory +
            "testPictures\\" + mainPictureList[currentPicturePoss].ToString();
            string answerPictureFilePath = projectDirectory +
            "testPictures\\" +
            mainPictureList[currentPicturePoss].ToString().Substring(0, 3)

```

					ДП.КН 22.481.32.000	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

```

+
mainPictureList[currentPicturePoss].ToString().Substring(1);
    string firstPictureFilePath = projectDirectory +
"testPictures\\" + choicePictureList[firstPicturePoss].ToString();
    string secondPictureFilePath = projectDirectory +
"testPictures\\" +
choicePictureList[secondPicturePoss].ToString();
    //усунення випадків повторення картинок у
    //варіантах відповідей, коли масив головних картинок неповний
    if (firstPictureFilePath == answerPictureFilePath)
    {
        firstPicturePoss = rand.Next(0, 40);
        firstPictureFilePath = projectDirectory +
"testPictures\\" + choicePictureList[firstPicturePoss].ToString();
    }
    if (secondPictureFilePath == answerPictureFilePath
|| secondPictureFilePath == firstPictureFilePath)
    {
        secondPicturePoss = rand.Next(0, 40);
        secondPictureFilePath = projectDirectory +
"testPictures\\" + choicePictureList[firstPicturePoss].ToString();
    }
    mainPictureBox.Image =
Image.FromFile(@mainPictureFilePath);
    switch (rand.Next(1, 4))
    {
        case 1:
            leftPictureBox.Image =
Image.FromFile(@answerPictureFilePath);
            whereAnswer = 1;
            centerPictureBox.Image =
Image.FromFile(@firstPictureFilePath);
            rightPictureBox.Image =
Image.FromFile(@secondPictureFilePath);
            break;
        case 2:
            centerPictureBox.Image =
Image.FromFile(@answerPictureFilePath);
            whereAnswer = 2;
            leftPictureBox.Image =
Image.FromFile(@firstPictureFilePath);
            rightPictureBox.Image =
Image.FromFile(@secondPictureFilePath);
            break;
        case 3:
            rightPictureBox.Image =
Image.FromFile(@answerPictureFilePath);
            whereAnswer = 3;
            leftPictureBox.Image =
Image.FromFile(@firstPictureFilePath);
            centerPictureBox.Image =
Image.FromFile(@secondPictureFilePath);
            break;
    }

```

					ДП.КН 22.481.32.000	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

```

mainPictureList.Remove(mainPictureList[currentPicturePoss]);
//MessageBox.Show(mainPictureList.Count().ToString());
    }
    else
    {
        using (MainForm frm = new MainForm())
        {
            this.Hide();
            frm.ShowDialog();
        }
    }
}
public TestForm()
{
    InitializeComponent();
    for (int i = 1; i <= 8; i++)
    {
        for (int j = 1; j <= 5; j++)
            mainPictureList.Add(i.ToString() + "_" +
j.ToString() + ".png");
    }
    for (int i = 1; i <= 8; i++)
    {
        for (int j = 1; j <= 5; j++)
            choicePictureList.Add(i.ToString() + "_" +
j.ToString() + "_" + j.ToString() + ".png");
        mainPictureList.ToArray();
        makeTest();
    }
    private void TestForm_FormClosed(object sender,
FormClosedEventArgs e)
    {
        Application.Exit();
    }
    private void button1_Click(object sender, EventArgs e)
    {
        makeTest();
    }
    private void leftPictureBox_Click(object sender, EventArgs
e)
    {
        if (whereAnswer == 1)
        {
            points++;

            centerPictureBox.Hide();
            rightPictureBox.Hide();

            arrowCenter.Hide();
            arrowRight.Hide();
            pointsTextBox.Text = points.ToString() + "/" +
testCount.ToString();
        }
        Else

```

					ДП.КН 22.481.32.000	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        {
            leftPictureBox.Image =
Image.FromFile(projectDirectory + "wrongAnswer.png");
        }
    }

    private void centerPictureBox_Click(object sender,
EventArgs e)
    {
        if (whereAnswer == 2)
        {
            points++;

            leftPictureBox.Hide();
            rightPictureBox.Hide();

            arrowLeft.Hide();
            arrowRight.Hide();

            pointsTextBox.Text = points.ToString() + "/" +
testCount.ToString();
        }
        else
        {
            centerPictureBox.Image =
Image.FromFile(projectDirectory + "wrongAnswer.png");
        }
    }

    private void rightPictureBox_Click(object sender,
EventArgs e)
    {
        if (whereAnswer == 3)
        {
            points++;
            centerPictureBox.Hide();
            leftPictureBox.Hide();

            arrowCenter.Hide();
            arrowLeft.Hide();

            pointsTextBox.Text = points.ToString() + "/" +
testCount.ToString();
        }
        else
        {
            rightPictureBox.Image =
Image.FromFile(projectDirectory + "wrongAnswer.png");
        }
    }

    private void button1_Click_1(object sender, EventArgs e)
    {
        using (MainForm frm = new MainForm())

```

					ДП.КН 22.481.32.000	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		


```

        {
            this.Hide();
            frm.ShowDialog();
        }
    }

e) private void nextTestButton_Click(object sender, EventArgs
    {
        makeTest();

        leftPictureBox.Show();
        rightPictureBox.Show();
        centerPictureBox.Show();

        arrowCenter.Show();
        arrowLeft.Show();

        arrowRight.Show();

        testCount++;
        pointsTextBox.Text = points.ToString() + "/" +
testCount.ToString();
    }

    protected override bool ProcessCmdKey(ref Message msg,
Keys keyData)
    {
        //capture right arrow key
        if (keyData == Keys.Right)
        {
            makeTest();

            leftPictureBox.Show();
            rightPictureBox.Show();
            centerPictureBox.Show();

            arrowCenter.Show();
            arrowLeft.Show();
            arrowRight.Show();

            testCount++;
            pointsTextBox.Text = points.ToString() + "/" +
testCount.ToString();
            return true;
        }

        return base.ProcessCmdKey(ref msg, keyData);
    }
}

```

					ДП.КН 22.481.32.000	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		