

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ БІЗНЕСУ ТА СУЧАСНИХ
ТЕХНОЛОГІЙ

ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ ІНФОРМАТИКИ

Допускається до захисту

Завідувач кафедри _____ О.О. Ємець
(підпис)

« _____ » _____ 2021 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО БАКАЛАВРСЬКОЇ РОБОТИ

на тему

СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
З ТЕМИ «УМОВНІ ОПЕРАТОРИ МОВИ PHP»

зі спеціальності 122 «Комп'ютерні науки»

Виконавець роботи Клименко Микола Миколайович

_____ «__» _____ 2021р.
(підпис)

Науковий керівник доц., каф.-мат. наук, Чілікіна Т.В.

_____ «__» _____ 2021р.
(підпис)

ПОЛТАВА 2021 р.

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»**

ЗАТВЕРДЖУЮ

Завідувач кафедри _____ О.О. Ємець

« 8 » вересня 2020р.

**Завдання та календарний графік
виконання дипломної роботи**

**Студент(ка) спеціальності 122 «Комп'ютерні науки»
Прізвище, ім'я, по батькові Клименко Микола Миколайович**

1. Тема «Створення програмного забезпечення з теми «Умовні оператори мови PHP» затверджена наказом ректора № 121-Н від « 1 » вересня 2020 р.

Термін подання студентом бакалаврської роботи « 20 » травня 2021 р.

2. Вихідні дані до дипломної роботи: публікації з теми навчальні тренажери в дистанційних курсах з комп'ютерних наук.

Завданням розробки є програмна реалізація тренажеру для навчання темі «Умовні оператори мови PHP».

3. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Вступ

1 ПОСТАНОВКА ЗАДАЧІ

1.1. Постановка задачі розробки тренажера

2 ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1 Огляд робіт, де розглянуте аналогічне до теми роботи завдання

2.2 Позитивні аспекти оглянутих робіт

2.3 Вади розробок з оглянутих робіт

3 ТЕОРЕТИЧНА ЧАСТИНА

3.1 Загальні відомості

3.1.1 Оператор if

3.1.2 Оператор else

3.1.3 Оператор switch

3.2 Алгоритм роботи тренажера

3.3 Блок-схема програми-тренажера

3.4 Обґрунтування вибору програмних засобів для реалізації завдання роботи

4 Практична частина

4.1 Опис процесу програмної реалізації

4.2 Перевірка валідності (тестування всіх процедур)

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ
ДОДАТОК А

4. Перелік графічного матеріалу: 3-4 аркуші блок-схем, інші необхідні ілюстрації.

5. Консультанти розділів бакалаврської роботи

Розділ	Прізвище, ініціали, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1. Постанова задачі	Чілікіна Т.В.	8.09.20	8.09.20
2. Інформаційний огляд	Чілікіна Т.В.	8.09.20	8.09.20
3. Теоретична частина	Чілікіна Т.В.	8.09.20	8.09.20
4. Практична реалізація	Чілікіна Т.В.	8.09.20	8.09.20

6. Календарний графік виконання бакалаврської роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ	10.05.21	
2. Вивчення методичних рекомендацій та стандартів та звіт керівнику	15.09.20	
3. Постановка задачі	1.10.20	
4. Інформаційний огляд джерел бібліотек та інтернету	2.11.20	
5. Теоретична частина	1.02.21	
6. Практична частина	17.05.21	
7. Закінчення оформлення	21.05.21	
8. Доповідь студента на кафедрі	28.05.21	
9. Доробка (за необхідністю), рецензування	14.06.21	

Дата видачі завдання « 8 » вересня 2020 р.

Студент Клименко Микола Миколайович
Науковий керівник _____ доц., каф.-мат. наук, Чілікіна Т.В.
(підпис)

Результати захисту бакалаврської роботи

Дипломна робота оцінена на _____
(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № _____ від « _____ » _____ 2021 р.

Секретар ЕК _____
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Записка: 53 стор., основна частина 48 стор., джерел - 14.

Предмет розробки – тренажер для навчання темі «Умовні оператори мови PHP».

Мета роботи – програмно реалізувати тренажер для навчання темі «Умовні оператори мови PHP» для систем дистанційного навчання.

Методи, які були використані для розв’язування задачі – для створення програмного забезпечення була використана платформа Unity 2020 та мова програмування C# та IDE MS Visual Studio.

Розроблено алгоритм навчального тренажера та здійснена програмна реалізація тренажера «Умовні оператори PHP»

Ключові слова: УМОВНІ ОПЕРАТОРИ PHP, НАВЧАЛЬНИЙ ТРЕНАЖЕР, ДИСТАНЦІЙНЕ НАВЧАННЯ.

ЗМІСТ

ВСТУП.....	5
1 Постановка задачі.....	7
2 Інформаційний огляд	8
2.1 Огляд робіт зі схожим завданням.....	8
2.2 Позитивні аспекти оглянутих робіт	8
2.3 Вади розробок з оглянутих робіт.	9
3 Теоретична частина.....	11
3.1 Загальні відомості	11
3.1.1 Оператор if	11
3.1.2 Оператор else	12
3.1.3 Оператор switch	12
3.2 Алгоритм роботи тренажера	14
3.3 Блок-схема програми-тренажера.....	24
3.4 Обґрунтування вибору програмних засобів для реалізації завдання роботи	27
4 Практична частина	31
4.1 Опис процесу програмної реалізації	31
4.2 Перевірка валідності (тестування всіх процедур)	43
ВИСНОВКИ.....	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	49
ДОДАТОК А.....	51

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, символів, скорочень
<i>ДК</i>	Дистанційний курс.
<i>ДН</i>	Дистанційне навчання.
<i>PHP</i>	Мова програмування.
<i>С#</i>	Мова програмування.
<i>Unity</i>	Платформа для програмної реалізації.
<i>IDE MS Visual Studio</i>	Середовище для написання програмного коду.

ВСТУП

В сучасному світі велике значення має навчання за допомогою мережі інтернет. Завдяки своїм перевагам, а саме: одночасному навчанню великої кількості людей, доступу до навчальних матеріалів з будь-якої частини світу, навчання у будь-який зручний час, дистанційне навчання використовується для багатьох сфер людської діяльності.

Одним із основних засобів дистанційного навчання є «програми-вчителі», або «програми-тренажери». Саме вони за допомогою свого широкого функціоналу є одним із найшвидших способів дистанційного навчання.

До функціоналу даних програм відносяться:

1. Робота з навчальними матеріалами за допомогою комп'ютера, телефону, планшету тощо;
2. Отримання дистанційних консультацій згідно до матеріалу тренажеру;
3. Швидка підтримка новими навчальними матеріалами;
4. Навчання у будь-якому вигляді, навіть у вигляді гри;
5. Можливість розвивання навчальних інтернет-ресурсів;
6. Доступ до веб-бібліотек;
7. Доступність для навчання для осіб з фізичними вадами;
8. Можливість розвивати передові педагогічні технології та навчання їм викладачів;
9. Підвищення ефективності навчання за рахунок гнучного графіку;

Програма-тренажер може бути використана у будь-якому місці у будь-який час, а при умові наявності усіх навчальних матеріалів безпосередньо в самій програмі, то й без доступу до мережі Інтернет.

Мета роботи – програмно реалізувати тренажер для навчання темі «Умовні оператори мови РНР» для систем дистанційного навчання.

Об'єкт роботи – алгоритмізація та програмування елементів тренажера для систем дистанційного навчання.

Предмет роботи – тренажер для навчання темі «Умовні оператори мови PHP».

Методи роботи – для створення програмного забезпечення була використана платформа Unity 2020. Для створення програми було використано мову програмування C# та IDE MS Visual Studio.

Актуальність цієї бакалаврської роботи полягає в розробці відсутнього навчального тренажеру для дистанційного курсу з теми «Умовні оператори мови PHP».

Новизною бакалаврської роботи є реалізація навчального тренажеру з теми «Умовні оператори мови PHP» для дистанційного навчального курсу. Унікальність даної програмної реалізації у тому, що вона не містить аналогів з даної теми.

Практичною цінністю реалізованого навчального тренажеру є створення засобу для навчання студентів темі «Умовні оператори мови PHP». Реалізований навчальний тренажер рекомендовано до використання студентами спеціальності «Комп'ютерні науки» дистанційного навчального курсу в ПУЕТ.

Пояснювальна записка до бакалаврської роботи містить в собі: вступ, постановку задачі, інформаційний огляд, теоретичну та практичну частину, висновки, список літератури та додаток.

Обсяг пояснювальної записки: 53 стор., в т.ч. основна частина 48 стор., джерел - 14.

1 Постановка задачі

В бакалаврській роботі з фаху головною задачею є створення програмного забезпечення у вигляді тренажера з теми «Умовні оператори PHP».

Дане програмне забезпечення має бути створено у середовищі розробки MS Visual Studio, з використанням мови програмування C# на платформі Unity 2020.

Головною задачею програмного забезпечення у вигляді тренажеру має бути навчання студентів темі «Умовні оператори мови PHP» .

Вимогами до розробки програмного забезпечення у вигляді навчального тренажеру є:

1. Обґрунтувати вибір мови та середовища розробки, які сумісні з системою дистанційного навчання MOODLE;
2. Складання основного алгоритму роботи «програми-тренажера»;
3. Складання блок-схеми алгоритму роботи програми;
4. Програмна реалізація тренажеру;
5. Перевірка та тестування програми.

Основними вимогами до роботи та використання навчального тренажеру є:

1. Розробка простого та зрозумілого сучасного інтерфейсу.
2. Створення перевірки відповідей користувача та виведення результат на екран.
3. Реалізація кнопки, яка здійснює відкриття сторінки з довідкою, що допомагає у вирішенні поставлених запитань та завдань.
4. Обробка всіх виключених ситуацій та можливих збоїв у роботі навчального тренажеру.
5. Завантаження реалізованого навчального тренажеру в систему дистанційного навчання MOODLE.

2 Інформаційний огляд

2.1 Огляд робіт зі схожим завданням

Під час огляду робіт зі схожим до теми завдання даної бакалаврської роботи були розглянуті та проаналізовані найбільш схожі та актуальні навчальні тренажери інших студентів, які вже впровадженні в систему дистанційного навчання MOODLE ПУЕТ.

Під час пошуку та аналізу робіт зі схожим завданням було знайдено наступні тренажери: тренажер з теми «Моделювання булевих функцій за допомогою елементарного персептрону» дисципліни «Нейронно-мережеві технології в інформатиці» (Гусак, Ю. С.) [1], тренажер з теми «Магазинні автомати» дисципліни «Теорія програмування» (Рзаев, Э. Ф.) [2], тренажер з теми «Складання математичної моделі» (Мандя, О. О.) [3] та тренажер з теми «Контекстовільні граматики» дисципліни «Теорія програмування» (Скромінський, М. В.) [4].

2.2 Позитивні аспекти оглянутих робіт

Перевагами розглянутого тренажеру «Моделювання булевих функцій за допомогою елементарного персептрону» (Гусак, Ю. С.) [1] є:

1. Існує вибір декількох мов інтерфейсу, наприклад: українська, російська та англійська.
2. Існують підказки користувачеві та при повторному вводі неправильної відповіді користувачеві надається правильний варіант відповіді.
3. Робота з тренажером починається з теоретичної частини, що дозволяє зрозуміти тему та переходить до практичних завдань, що дозволяють закріпити дану тему.
4. Існує можливість повторити роботу з тренажером.

Перевагами розглянутого тренажеру «Магазинні автомати» (Рзаев, Э. Ф.) [2] є:

1. Робота з тренажером зводиться до роботи з конкретними його панелями, такими як: початкова сторінка, сторінка з питаннями, сторінка з результатами.

2. Існує виведення повідомлення про помилкову відповідь в теоретичній частині та підказка в практичній.

3. В кінці роботи з тренажером є можливість отримати результати про його проходження.

Перевагами розглянутого тренажеру з теми «Складання математичної моделі» (Мандя, О. О.) [3] є:

1. Існує можливість авторизації в тренажері як викладача так і студента, з особливостями роботи з кожним.

2. Існує можливість роздрукувати дані по кожному студенту.

3. Існує можливість переглянути відомості про результати студентів.

4. Існує можливість зв'язатися з викладачем через тренажер.

Перевагами розглянутого тренажеру з теми «Контекстовільні граматики» дисципліни «Теорія програмування» (Скромінський, М. В.) [4] є:

1. При роботі з тестами є доступ до теоретичного матеріалу на одній сторінці.

2. Існує перевірка на введену відповідь.

3. Існує можливість повторити роботу з тренажером.

2.3 Ваді розробок з оглянутих робіт.

Недоліками розглянутого тренажеру «Моделювання булевих функцій за допомогою елементарного персептрону» (Гусак, Ю. С.) [1] є:

1. Необхідність наявності доповнення .NET Framework 4.0 та операційної системи Windows.

Недоліками розглянутого тренажеру «Магазинні автомати» (Рзаєв, Э. Ф.) [2] є:

1. Практичні завдання в тренажері реалізовано лише у вигляді тестів.

2. Повідомлення про помилку займає все вікно програми.

3. Немає можливості повторити роботу з тренажером через меню тренажеру.

Недоліками розглянутого тренажеру з теми «Контекстовільні граматики» дисципліни «Теорія програмування» (Скромінський, М. В.) [4] є:

1. У повідомленні з помилкою не має пояснення чи підказки до завдання.

2. Не зрозуміло для чого кнопка Start, можливо краще було б замінити на «Практичні завдання» або «Тести».

3 Теоретична частина

3.1 Загальні відомості

Умовні оператори забезпечують виконання або невиконання деякого оператора, групи операторів або блока в залежності від заданих умов. Використовуються умовні оператори в таких алгоритмах, де можливі декілька варіантів розв'язання задачі в залежності від початкових умов.

Під час роботи з бакалаврською роботою було розглянуто основні умовні оператори РНР, а саме оператори “if-else”, “switch-case” та деякі особливості роботи з ними.

3.1.1 Оператор if

Оператор if один з основних умовних операторів багатьох мов програмування, включаючи РНР. За його допомогою можливо виконувати певні фрагменти коду при відповідності заданій умові.

Його структуру можна представити наступним чином :

```
if (вираз) {блок_виконання}
```

На місці виразу може бути будь-яка умова РНР. В процесі виконання введений вираз перетвориться до логічного типу. Якщо в результаті перетворення значення виразу буде істинне, тобто “True”, то виконається блок-виконання. У іншому випадку (якщо значення виразу “False”) блок-виконання ігнорується.

Правила перетворення виразу до логічного типу:

1. Правила перетворення виразу до логічного типу:

- логічне False
- цілий нуль (0)
- дійсний нуль (0.0)
- порожній рядок і рядок "0"
- масив без елементів
- об'єкт без змінних (детально про об'єкти буде розказано в одній з

наступних лекцій)

- спеціальний тип NULL
2. Всі інші значення перетворюються в TRUE.

Приклад виконання коду:

```
<?php $a = 10;
if ($a == 10)
{ echo 'true'; }
?>
```

3.1.2 Оператор else

Оператор if може існувати без оператора else, він використовується для розширення його можливостей. Таким чином для виконання певних дій на випадок якщо вираз описаний в if = “false” можна застосувати блок else.

Його структуру можна представити наступним чином:

```
if (вираз) {блок_виконання}
else {блок_виконання1}
```

Цю конструкцію можна описати так: якщо виконується умова записана у виразі то виконується блок-виконання, у випадку не виконання умови у виразі виконується блок-виконання1.

Приклад виконання коду:

```
<?php $a = 10;
if ($a == 10)
{ echo 'true'; }
else { echo 'false'; }
?>
```

3.1.3 Оператор switch

У випадку необхідності перевірки коду на багато різних умов зручно використовувати оператор switch. Він перевіряє всі умови та виконує в залежності від цього різноманітні дії.

Його структуру можна представити наступним чином:

```
<?php
$a = значення;
```

```

switch ($a)
{ case значення: блок_виконання;
break;
case значення1: блок_виконання1
break; }
default: блок_виконання2
?>

```

На відміну від конструкції `if` значення виразу не перетворюється до логічного типу, а просто порівнюється зі значеннями перерахованими після ключового слова `case`. У випадку співпадання значення виконується блок_виконання від двокрапки до першого оператора `break`. Якщо жодне значення в прописаних `case` не співпало з введеним значенням чи виразом, то виконується блок_виконання прописаний після ключового слова `default`. Наприклад можна записати повідомлення про помилку.

Приклад виконання коду:

```

<?php
$a = 1;
switch ($a)
{ case 1: echo $a;
break;
case 2: $a ++;
break; }
default: 'error';
?>

```

Тобто при вводі одиниці на екран виведеться одиниця, при вводі двійки в змінну `$a` запишеться значення збільшене на одиницю, якщо введеться значення іншого типу, чи значення введення якого не передбачене в програмі на виході отримаємо повідомлення про помилку. [5]

3.2 Алгоритм роботи тренажера

Під час роботи з тренажером використовуються питання з декількома варіантами відповідей, після вибору правильної відповіді буде можливість перейти далі, у разі вибору неправильної відповіді буде реалізована кнопка повернення до завдання теоретичного типу та кнопка підказки. Також в деяких практичних завданнях реалізовано можливість вводу свого варіанту відповіді.

Під час роботи з практичними завданнями у першому випадку користувач отримує певну задачу з трьома варіантами відповіді, при виборі правильного варіанту з'являється повідомлення про правильну відповідь у вигляді «✓» та з'являється кнопка переходу до наступного завдання, при виборі неправильної відповіді з'являється повідомлення про помилку у вигляді «X» та з'являється кнопка «Підказка».

Алгоритм роботи з навігаційними елементами тренажеру описано наступними кроками:

Крок 1: Запуск тренажеру та вибір оптимальних налаштувань для екрану.

Крок 2: Перехід до головного меню тренажеру. В головному меню описано тему та надано доступ до елементів тренажеру – «Довідка», «Вибір практичного завдання» та «Інформація про тренажер».

Крок 3: При переході до елемента «Інформація про тренажер» користувач отримує інформацію про виконавця та керівника роботи. Для повернення до головного меню необхідно натиснути кнопку «ОК».

Крок 4: При переході до елемента «Довідка» користувач отримує доступ до кнопок для переходу до певного довідкового матеріалу тренажеру. Існує можливість повернутися до меню «Довідка» натиснувши кнопку «Тема» в довідковому матеріалі.

Крок 5: При переході до елемента «Вибір практичної теми» користувач отримує доступ до вікна з трьома практичними темами. При натисненні на практичну тему користувач отримує доступ до практичних завдань з неї. Існує можливість повернутися до меню «Вибір практичної теми» натиснувши кнопку «До практики» під час роботи з практичним завданням.

Алгоритм роботи з навчальними матеріалами описано наступними кроками:

Крок 1: Після вибору довідкового матеріалу користувач переходить до вікна з довідковою інформацією по певній темі. Після ознайомлення з довідкою користувач переходить до практичних завдань по першій темі. Існує можливість обрати будь-яке практичне завдання через відповідне меню.

Крок 2: Користувач отримує практичне завдання з теми «IF-ELSE».

Задача 1. Якщо \$a дорівнює 10 то вивести "true" якщо ні - "false".

Заповніть пропуски ("...") в кодї:

```
<?php
$a = 10;
if ($a ... 10) {
    echo 'true';
} else {
    echo 'false';
}
?>
```

Варіанти (правильний 1) –

1. ==
2. !=
3. ===»

Крок 3: Користувач отримує практичне завдання:

Задача 2. Дана наступна програма, що є її результатом?

```
<?php
$a == 0;
if ($a < 0) {
    echo 'true';
} else {
    echo 'false';
}
```

?>

Варіанти (правильний 2) –

1. true
2. false
3. error»

Крок 4: Користувач отримує практичне завдання:

Задача 3. Дана наступна програма, що є її результатом?

```
<?php  
$a = -3;  
if ($a == 0) {  
echo 'true';  
} else {  
echo 'false';  
}  
?>
```

Варіанти (правильний 2) –

1. true
2. false
3. error

Крок 5: Користувач отримує практичне завдання:

Задача 4. Дана наступна програма, що є її результатом?

```
<?php  
$a = -12;  
if ($a != 0) {  
echo 'true';  
} else {  
echo 'false';  
}  
?>
```

Варіанти (правильний 1) –

1. true
2. !true
3. False

Крок 6: Користувач отримує практичне завдання:

Задача 5. Дана наступна програма, що є її результатом?

```
<?php
$a = "abc";
if ($a === "abc") {
    echo 'true';
} else {
    echo 'false';
}
?>
```

Варіанти (правильний 3) –

1. false
2. error
3. true

Крок 7: Користувач отримує практичне завдання з теми «ЛОГІЧНІ ЗМІННІ».

Задача 1. Дано наступний код, що є результатом його виконання?

```
<?php
$a = 4;
$b = 3;
$c = 66;
$var = true;
if ($var == true) {
    echo $a;
}
else {
    echo 'sample1';
}
```

```

}
if ($var == false) {
echo $b;
}
else {echo 'sample2';}
?>

```

Варіанти (правильний 1) –

1. 4sample2
2. 3sample2
3. 4sample1

Крок 8: Користувач отримує практичне завдання:

Задача 2. Дано наступний код, що є результатом його виконання?

```

<?php
$a = 4;
$b = 3;
$c = 66;
$var = !true;
if ($var == true) {echo $a;}
else {echo 'sample1';}
if ($var == false) {echo $b;}
else {echo 'sample2';}
?>

```

Варіанти (правильний 3) –

1. sample14
2. sample1sample2
3. sample13

Крок 9: Користувач отримує практичне завдання:

Задача 3. Дано наступний код, що є результатом його виконання?

```

<?php
$a = 4;

```

```
$b = 3;
$c = 66;
$var = false;
if ($var) {
    echo $a;
}
else {
    echo 'sample1';
}
if (!$var) {
    echo $b;
}
else {echo 'sample2';}
?>
```

Варіанти (правильний 2) –

1. sample14
2. sample13
3. sample1sample2

Крок 10: Користувач отримує практичне завдання:

Задача 4. Дано наступний код, що є результатом його виконання?

```
<?php
$a = 4;
$b = 3;
$c = 66;
$var = false;
if ($var == true) {
    echo $a;
}
else {
    echo $c;
```

```

}
if ($var == false) {
echo $b;
}
else {echo 'sample2';}
?>

```

Варіанти (правильний 2) –

1. 664
2. 663
3. 66sample2

Крок 11: Користувач отримує практичне завдання:

Задача 5. Дано наступний код, що є результатом його виконання?

Введіть правильну відповідь у відповідному полі (оберіть мову

"Англійська")

```

<?php
$a = 4;
$b = 3;
$c = 66;
$var = !false;
if ($var == !true) {
echo $a;
}
else {
echo $c;
}
if (!$var) {
echo $b;
}
else {echo 'sample2';}
?>

```

Після вводу правильна відповідь «66sample2»

Крок 12: Користувач отримує практичне завдання з теми «SWITCH-CASE».

Задача 1. Сайт підтримує 3 мови - російська, англійська, німецька. Змінна \$ lang може приймати 3 значення - 'ru', 'en' і 'de'. Залежно від значення змінної \$ lang слід вивести фразу на одній з мов.

Заповнити пропуски в коді.

```
<?php
switch ($lang) {
case '??':
echo 'Російська';
break;
case '??':
echo 'Англійська';
break;
case '??':
echo 'Німецька';
break;
default:
echo 'Ця мова не підтримується';
break;
}
?>
```

Варіанти відповіді (Правильна 1) :

1. ru en de
2. en ru de
3. ru de en

Крок 13: Користувач отримує практичне завдання:

Задача 2. Змінна \$1 може приймати 2 значення: 'ru' і 'en'. Якщо вона має значення 'ru', то в змінну \$a запишемо масив днів тижня російською мовою, а якщо має значення 'en' - то англійською.

Заповніть пропуски в коді.

```
<?
$a='ru';
switch($1){
case 'ru':
$a=['пн','вт','ср','чт','пт','сб','вс'];
break;
case 'en':
$a=['mn','ts','wd','th','fr','st','sn'];
break;
}
????
?>
```

Варіанти відповіді (Правильна 3) :

1. printf a
2. echo \$a
3. var_dump(\$a)

Крок 14: Користувач отримує практичне завдання:

Задача 3. Який результат коду, що наведено нижче?

```
<? Php
$ Speed = 55;
switch ($ speed)
{
case 30:
echo "Ваша швидкість 30 км / год";
break;
case 58:
```



```
echo "Ваша швидкість 50 км / год";  
break;  
case 70:  
echo "Перевищення швидкості!";  
break;  
default:  
echo "Швидкість в межах норми";  
break;  
}  
>
```

Варіанти відповіді (Правильна 2) :

1. Перевищення швидкості!
2. Швидкість в межах норми
3. Ваша швидкість 55 км / год

Крок 15: Після вибору правильного варіанту відповіді користувач отримує доступ до кнопки «Наступне завдання»

Крок 16: Після вибору неправильного варіанту відповіді користувач отримує доступ до кнопки «Підказка», що надає коротку підказку до завдання та доступ до кнопки, що дозволяє ознайомитися з теорією до завдання. Після переходу до теорії, з'являється кнопка повернення до практичного завдання.

Крок 17: Для завершення роботи з тренажером реалізовано кнопку Вихід, що дозволяє завершити роботу у будь-який момент роботи з тренажером.

3.3 Блок-схема програми-тренажера

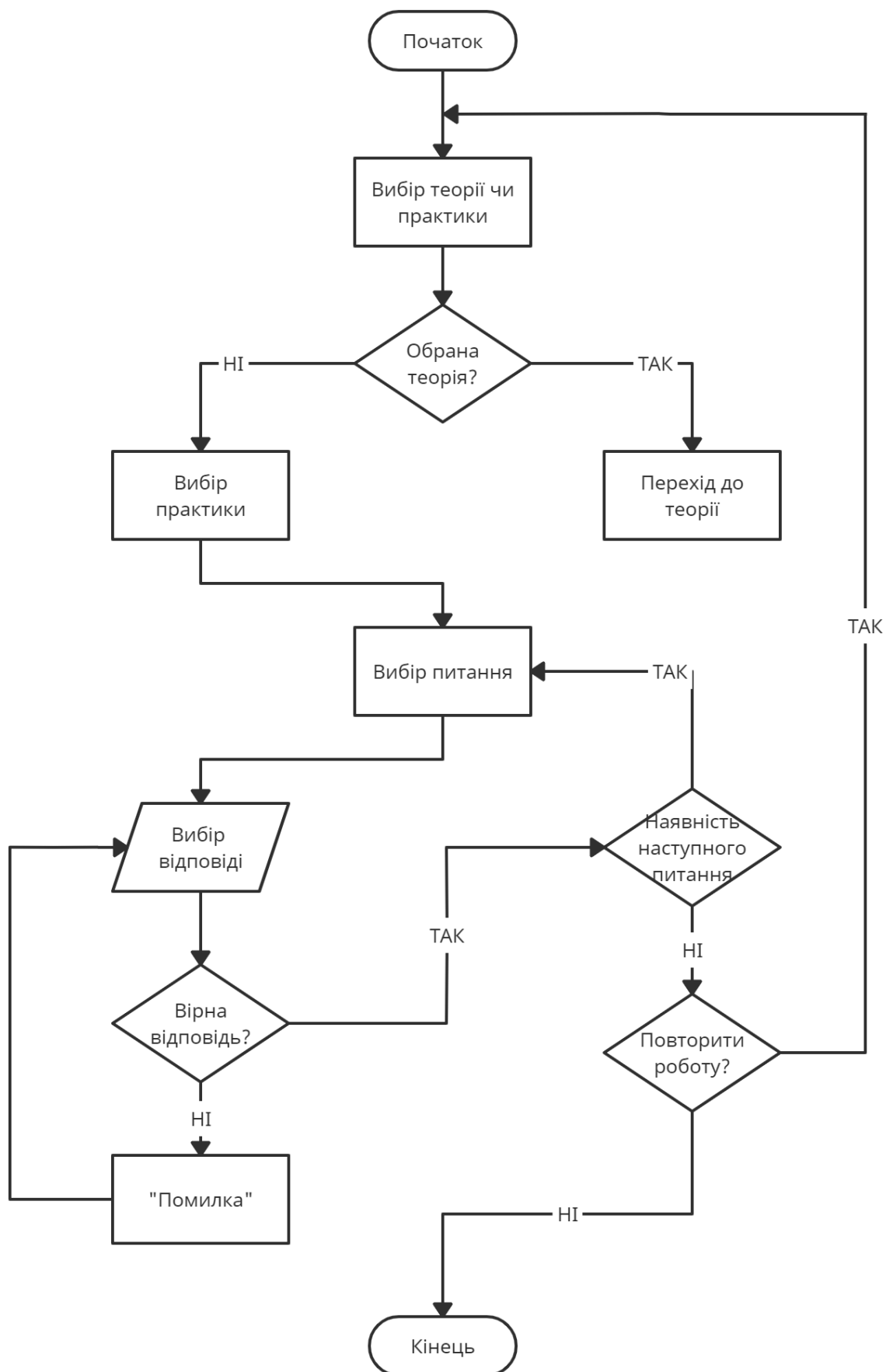


Рисунок 3.1 – Блок-схема програми-тренажера

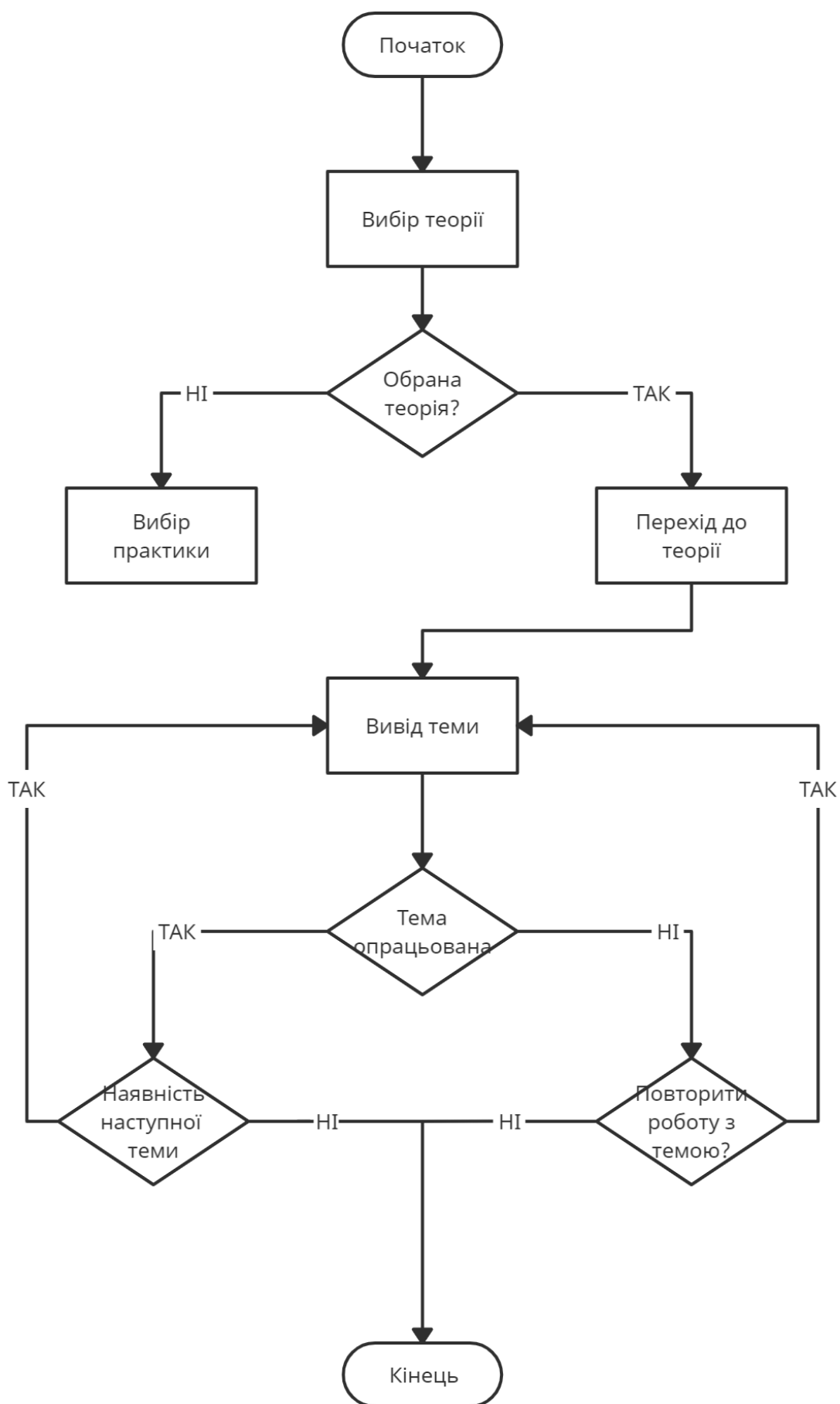


Рисунок 3.2 – Блок-схема роботи з теоретичним матеріалом

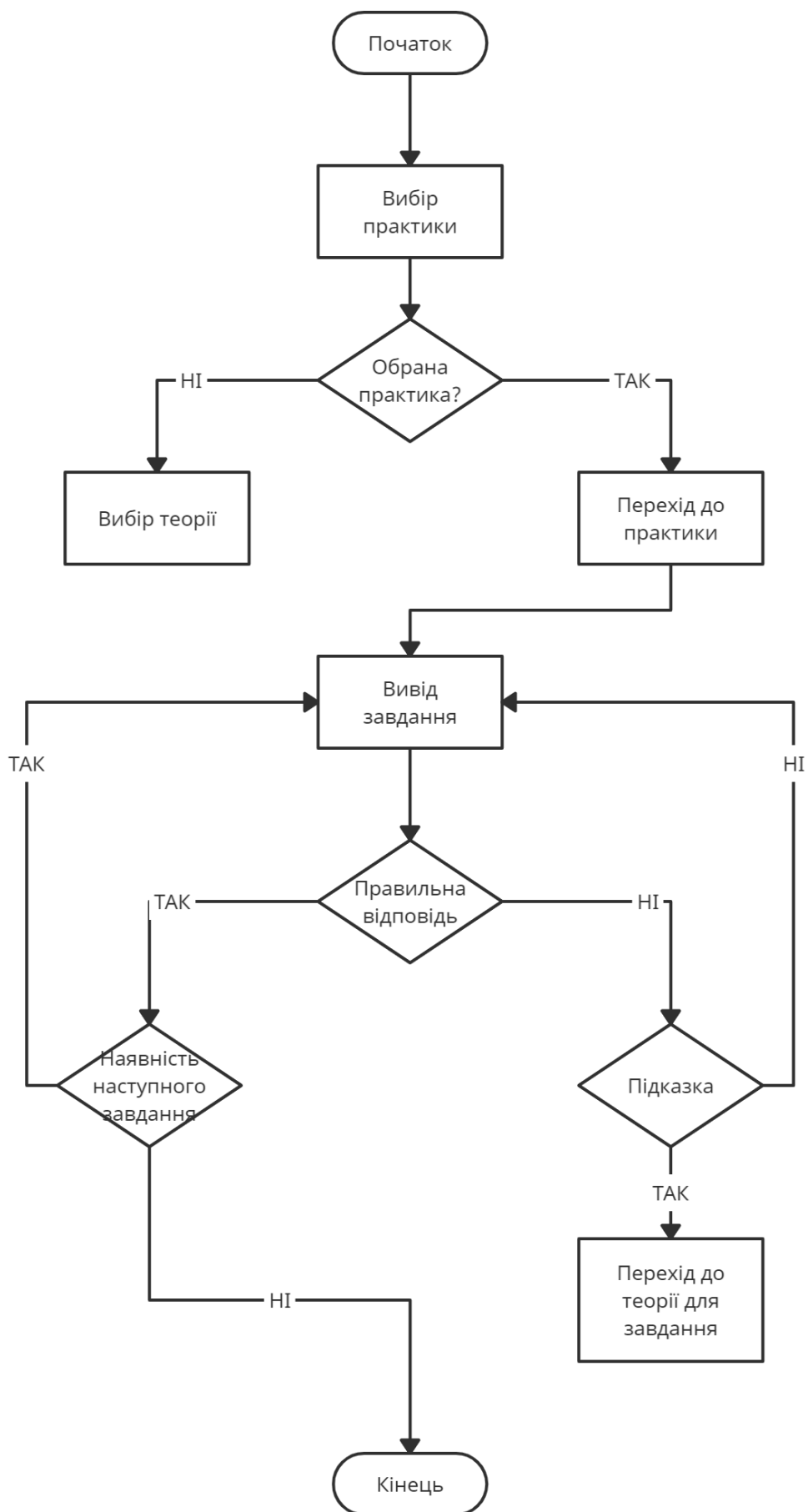


Рисунок 3.3 – Блок-схема роботи з практичним матеріалом

3.4 Обґрунтування вибору програмних засобів для реалізації завдання роботи

«Microsoft Visual Studio – це інтегроване середовище розробки (IDE) від Microsoft. Він використовується для розробки комп'ютерних програм, а також веб-сайтів, веб-програм, веб-сервісів та мобільних додатків. Visual Studio використовує платформи для розробки програмного забезпечення Microsoft, так як Windows API, Windows Forms, Windows Presentation Foundation, Windows Store і Microsoft Silverlight. Він може створювати як власний код, так і керований код.

Visual Studio дає змогу розробнику написати єдину програму для роботи на кількох платформах Windows, таких як мобільний, настільний і навіть експериментальний середовище Microsoft HoloLens. Він також забезпечує спосіб створення додатків, які взагалі не працюють на комп'ютерах Windows, але замість цього вони працюють на пристроях ISO або у веб-додатках у хмарі. Visual Studio включає в себе редактор коду, що підтримує IntelliSense (компонент завершення коду), а також рефакторинг коду. Вбудований відкладчик працює як відкладчик на рівні вихідного рівня, так і відкладчик на рівні машини. Інші вбудовані інструменти включають програму для кодування, конструктор форм для побудови графічних інтерфейсів, веб-дизайнерів, дизайнер класів та дизайнер схеми баз даних. Він приймає плагіни, як покращують функціональність практично на всіх рівнях, включаючи підтримку систем керування джерельними ресурсами та додавання нових наборів інструментів, таких як редактори та візуальні розробники для мов або наборів інструментів для інших аспектів розробки програмного забезпечення життєвий цикл.

Visual Studio підтримує 26 мов програмування і дозволяє редакторові коду та відкладчику підтримувати (в тій чи іншій мірі) майже будь-яку мову програмування, якщо існує певна мова-служба. Вбудовані мови включають C, C++, C++/CLI, Visual Basic, NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML та CSS. Підтримка інших мов таких як Python, Ruby, Node.js та M, серед

інших, доступна через плагіни. А також містить в собі сильний інструмент NuGet для завантаження різного роду плагінів та додатків.

Visual Studio дозволяє підключити функціональність, кодовану як VSPackage. Після встановлення функціональності доступна як сервіс. IDE надає три сервіси: SVsSolution, що забезпечує можливість переліку проектів та рішень; SVsUIShell, що забезпечує вікна та функціональність інтерфейсу користувача (включаючи вкладки, панелі інструментів та вікна інструментів); і SVsShell, що займається реєстрацією зв'язку між службами. Всі редактори, дизайнери, типи проектів та інші інструменти реалізовані як VSPackages. Visual Studio використовує COM для доступу до VSPackages.

Visual Studio не включає в себе будь-яку підтримку управління вихідним кодом, але вона визначає два альтернативні шляхи для інтеграції систем керування вихідними кодами з IDE. Visual Studio підтримує запуск декількох екземплярів середовища. У прикладах використовується різні вулики реєстру для зберігання їх конфігураційного стану і диференціюються за допомогою AppId (ідентифікатора програми). Нові версії програмного забезпечення організують свою колекцію інструментів для налагодження та профілювання за допомогою єдиного інтерфейсу інструментів діагностики.

Редактор коду Visual Studio також підтримує налаштування закладок у кодї для швидкої навігації. Інші навігаційні засоби включають згортання блоків коду та послідовного пошуку, крім звичайного текстового пошуку та пошуку регулярного пошуку. Редактор коду також включає в себе багатопозиційний буфер обміну та список завдань. Редактор коду підтримує фрагменти коду, які є збереженими шаблонами для повторювального коду, і можуть бути вставлені в код і налаштовані для роботи над проектом. Вбудований інструмент управління фрагментами коду. Редактор коду Visual Studio також підтримує рефакторинг коду, включаючи реорганізацію параметрів, зміну і перейменування методів, вилучення інтерфейсу та інкапсуляцію членів класу всередині властивостей, серед іншого.

Visual Studio має фонову компіляцію (також називається додатковою компіляцією). У міру написання коду Visual Studio компілює його у фоновому режимі, щоб надати відгук про помилки синтаксису та компіляції, позначені червоними хвилястими підкресленнями. Попередження позначені зеленим підкресленням. Тональна компіляція не генерує виконуваний код, оскільки для цього потрібен інший компілятор, ніж той, який використовується для створення виконуваного коду» [8-9]

Unity – багатоплатформовий інструмент для розробки відеоігор та застосунків, і рушій на якому вони працюють. Створені за допомогою Unity програми працюють на настільних комп'ютерних системах, мобільних пристроях та гральних консолях у дво- та тривимірній графіці, та на пристроях віртуальної чи доповненої реальності. Застосунки, створені за допомогою Unity, підтримують DirectX та OpenGL.

Редактор Unity має інтерфейс, що складається з різних вікон, які можна розташувати на свій розсуд. Завдяки цьому можна проводити налагодження гри чи застосунка прямо в редакторі. Головні вікна — це оглядач ресурсів проєкту, інспектор поточного об'єкта, вікно попереднього перегляду, оглядач сцени та оглядач ієрархії ресурсів[26].

Проєкт в Unity поділяється на сцени (рівні) — окремі файли, що містять свої ігрові світи зі своїм набором об'єктів, сценаріїв, і налаштувань. Сцени можуть містити в собі як об'єкти-моделі (ландшафт, персонажі, предмети довкілля тощо), так і порожні ігрові об'єкти — ті, що не мають моделі, проте задають поведінку інших об'єктів (тригери подій, точки збереження прогресу тощо). Їх дозволяється розташовувати, обертати, масштабувати, застосовувати до них скрипти. В них є назва (в Unity допускається наявність двох і більше об'єктів з однаковими назвами), може бути тег (мітка) і шар, на якому він повинен відображатися. Так, у будь-якого предмета на сцені обов'язково наявний компонент Transform — він зберігає в собі координати місця розташування, повороту і розмірів по всіх трьох осях. У об'єктів з видимою геометрією також за умовчанням присутній компонент Mesh Renderer, що робить модель видимою.

Різні моделі можуть об'єднуватися в набори (асети) для швидкого доступу до них. [7]

4 Практична частина

4.1 Опис процесу програмної реалізації

Даний програмний продукт розроблювався за допомогою платформи Unity 2020 за допомогою IDE MS Visual Studio. Тип роботи тренажера можна описати як інтерактивну презентацію в якій можливий вибір між теоретичними та практичними завданнями та зручним переходом між ними. Реалізована можливість вийти з тренажера та перейти в головне меню з будь-якого моменту роботи з ним за допомогою кнопок.

Код для кнопки виходу з тренажера:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class QuitButton : MonoBehaviour
6  {
7      public void QuitGame()
8      {
9          Debug.Log ("QUIT!!");
10         Application.Quit();
11     }
12 }
13
14
```

Рисунок 4.1 – Код кнопки виходу



Рисунок 4.2 – Зовнішній вигляд кнопки виходу

Перехід між слайдами тренажеру реалізовано за допомогою кнопок «Далі» та «Назад». Робота кнопки «Далі» базується на закритті поточного слайду та відкритті наступного. У разі закінчення роботи з завданнями теоретичного типу, в кінці кнопка «Далі» замінена на кнопку «Перейти до практики».

Код для роботи кнопки «Далі»:

```
Slide1.SetActive(false);
```

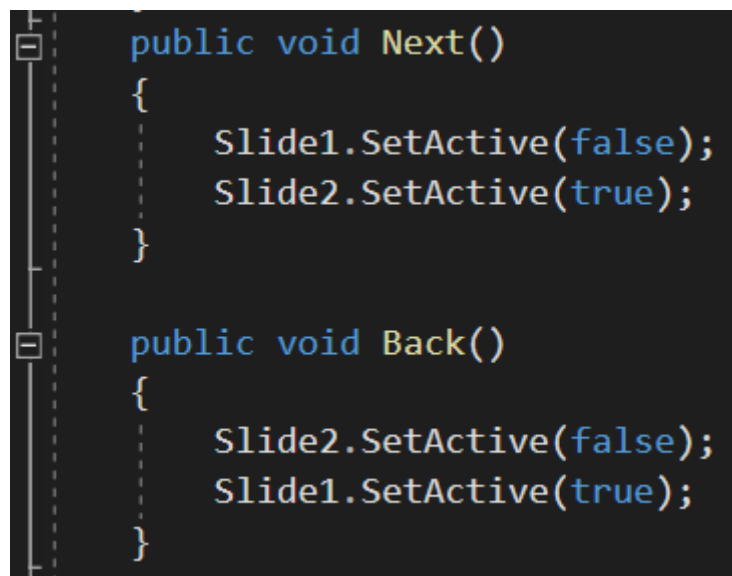
```
Slide2.SetActive(true);
```

Кнопка «Назад» працює за тим же принципом.

Код для роботи кнопки «Назад»:

```
Slide2.SetActive(false);
```

```
Slide1.SetActive(true);
```



```
public void Next()
{
    Slide1.SetActive(false);
    Slide2.SetActive(true);
}

public void Back()
{
    Slide2.SetActive(false);
    Slide1.SetActive(true);
}
```

Рисунок 4.3 – Скрипт для кнопок навігації

Робота з логічними змінними в Unity 2020 оформлена у вигляді приєднаного скрипту для кнопок (див. Рисунок 4.4).

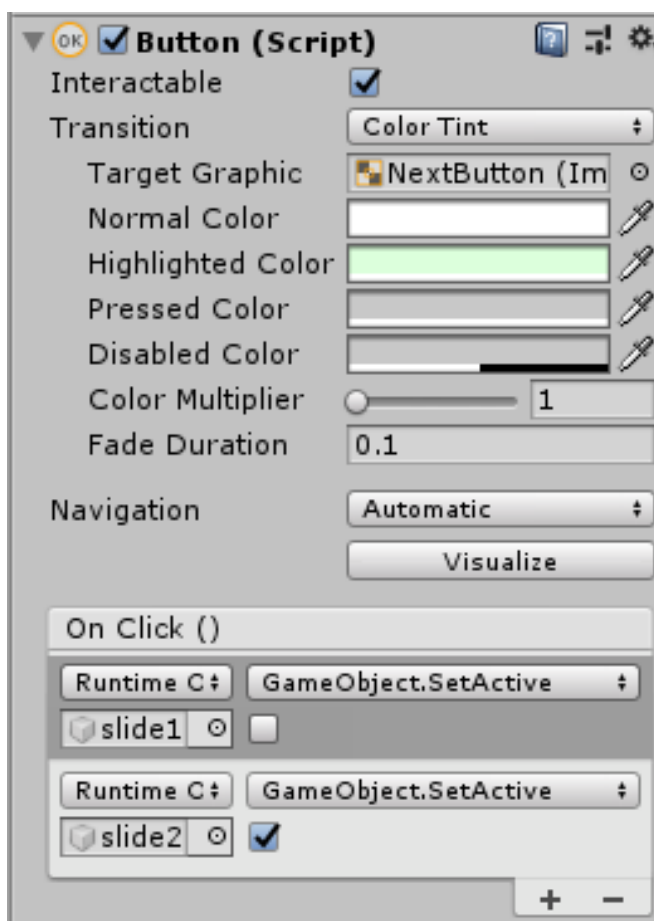


Рисунок 4.4 – Скрипт для виконання роботи кнопок

Всі кнопки в програмному забезпеченні виконані за даним принципом.

Головне меню тренажера (див. Рисунок 4.5) дозволяє здійснити вибір між завданнями теоретичного та практичного вигляду та вивести інформацію про тренажер.

Кнопка «Вибір теми» перенаправляє користувача на слайд з усіма доступними теоретичними темами (див. Рисунок 4.6).

Кнопка «Вибір практичного завдання» перенаправляє користувача на слайд з вибором теми та подальшим вибором завдань (див. Рисунок 4.7, Рисунок 4.8).

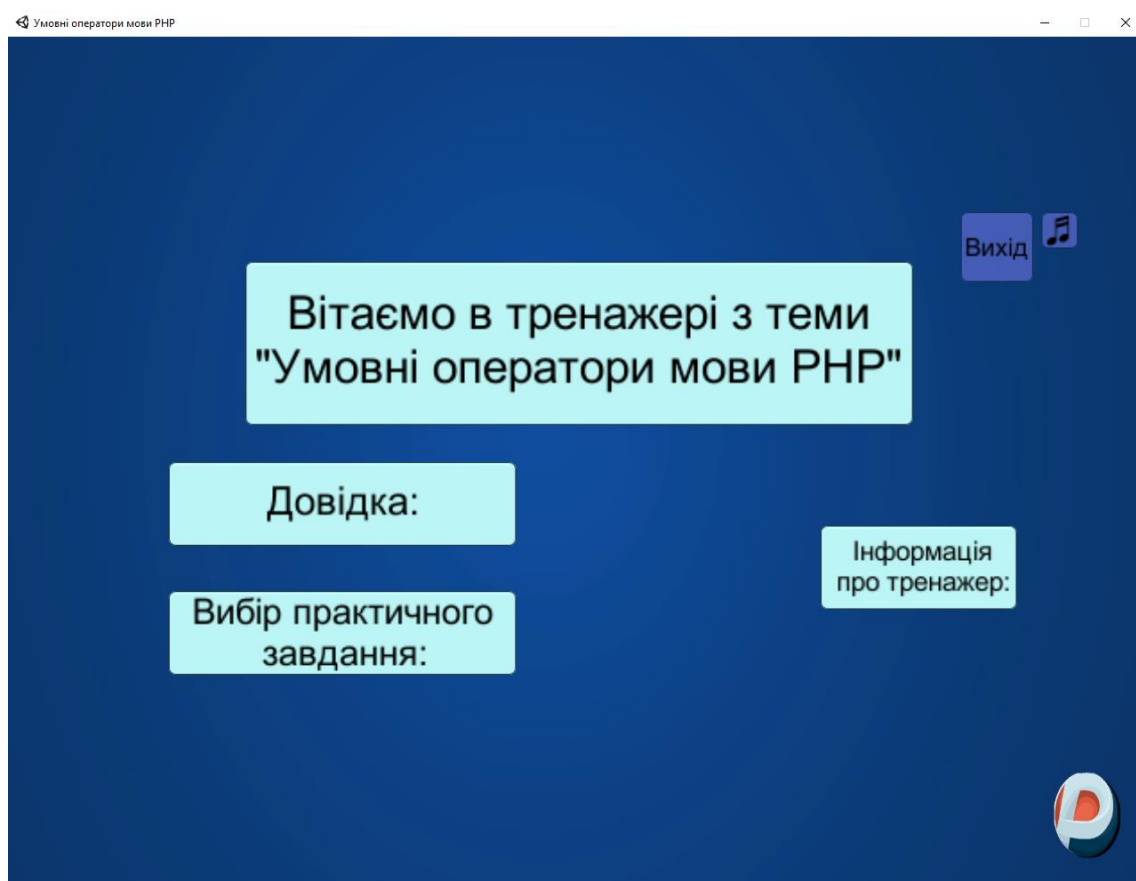


Рисунок 4.5 – Головне меню тренажера

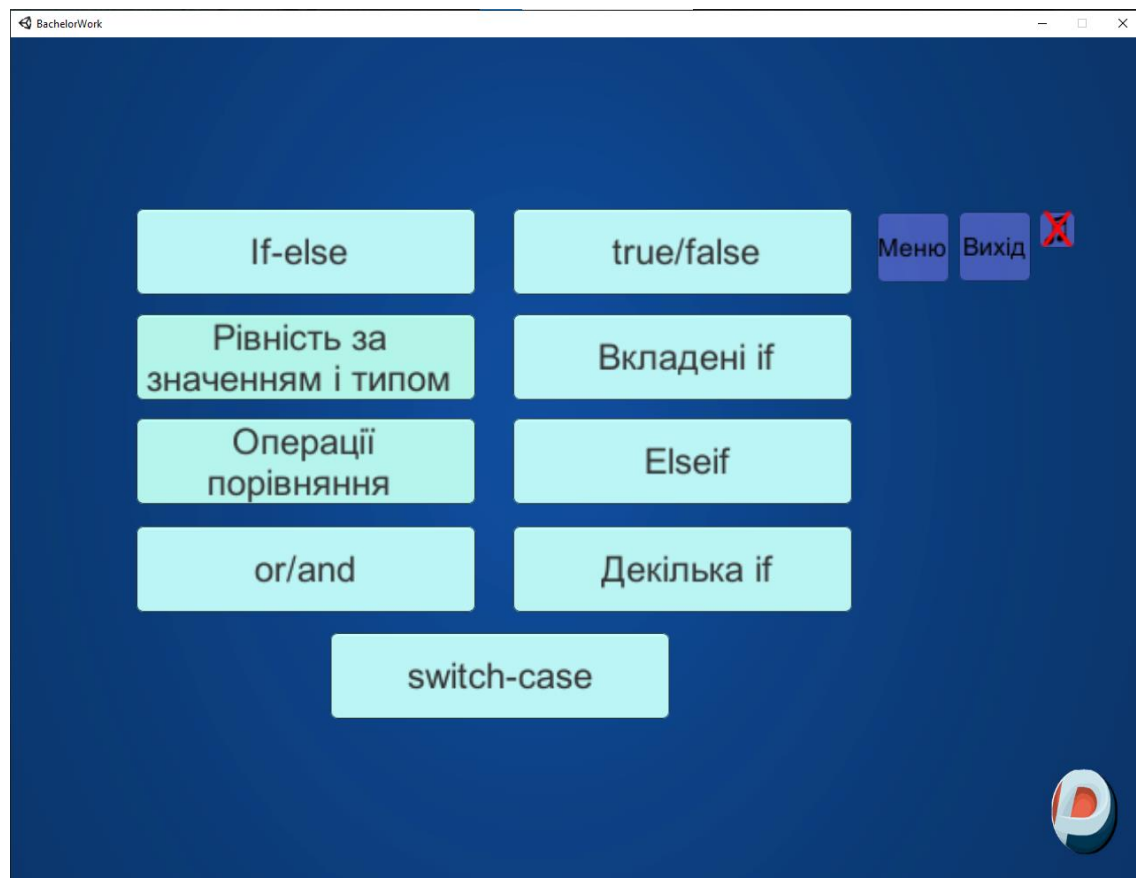


Рисунок 4.6 – Вибір теоретичних тем

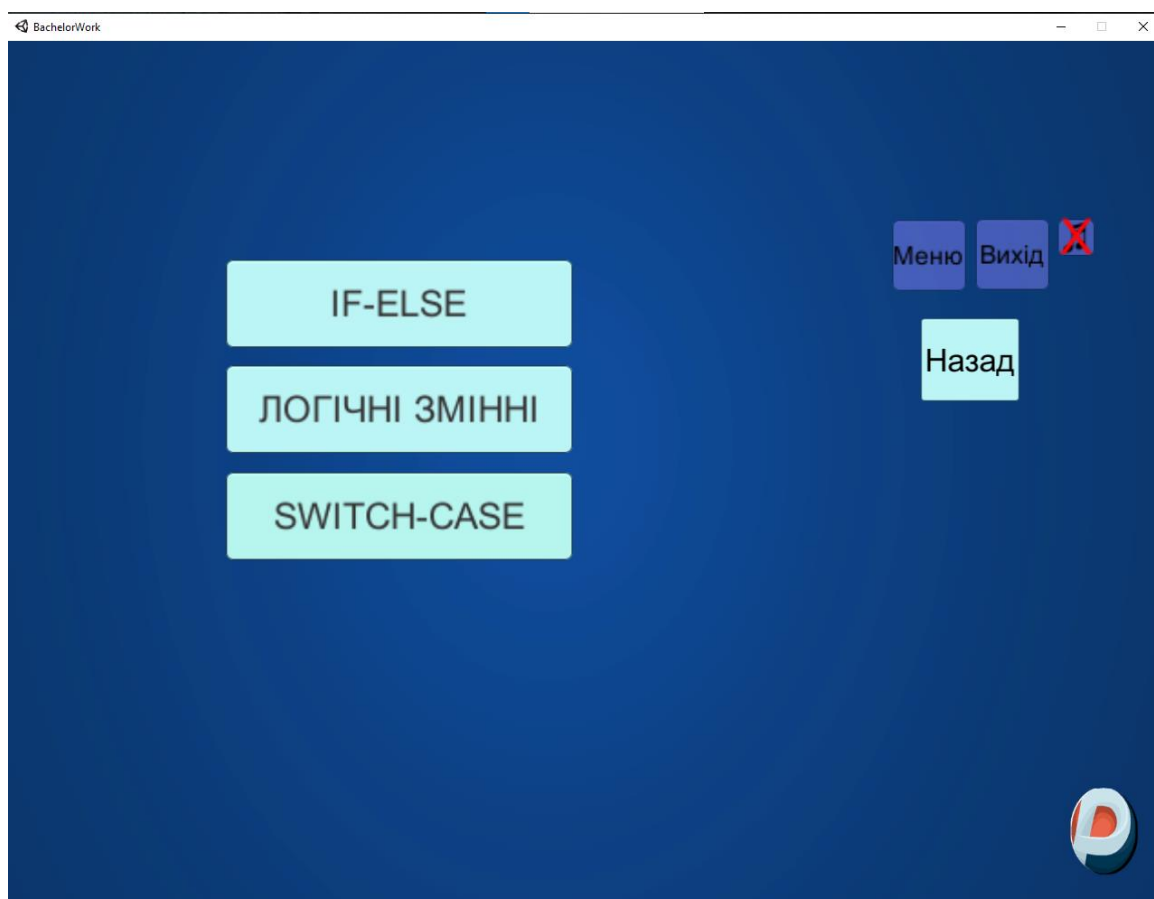


Рисунок 4.7 – Вибір теми для практичних завдань

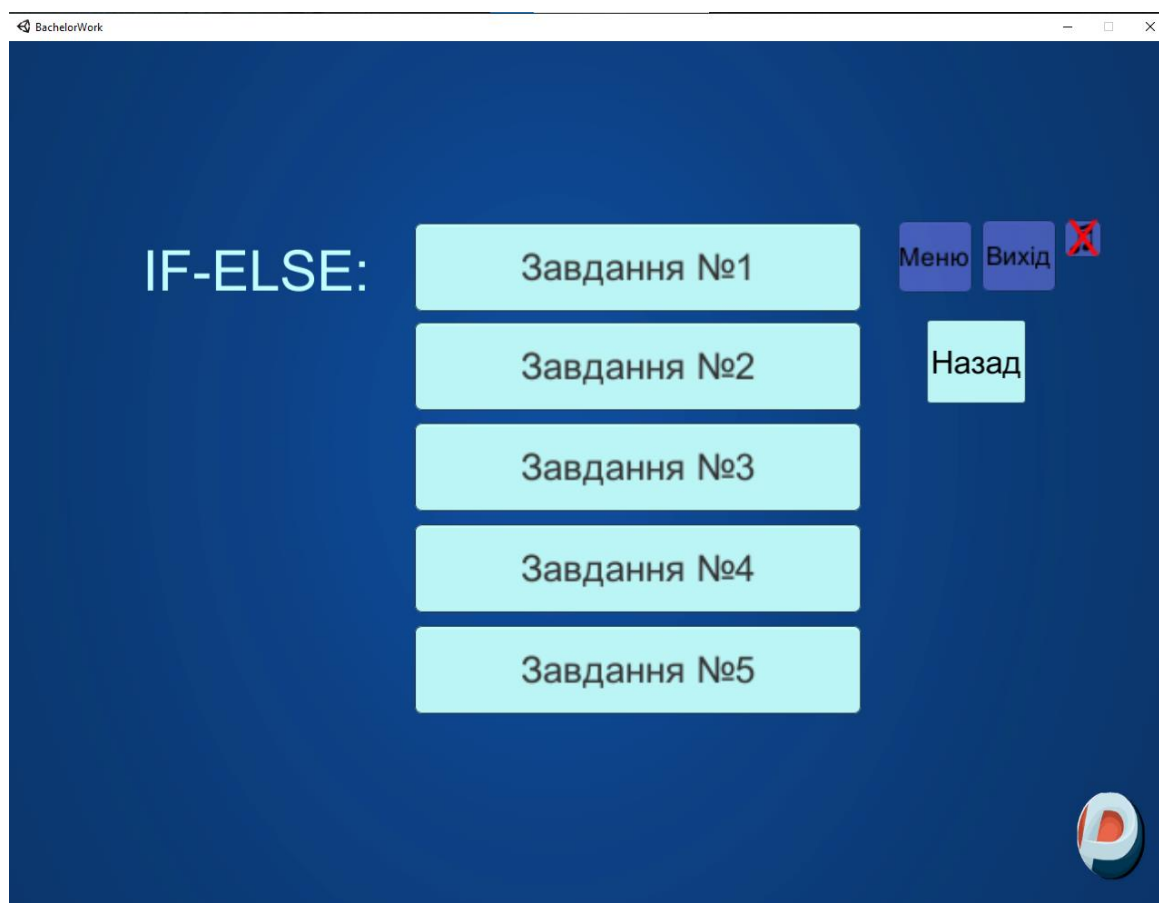


Рисунок 4.8 – Вибір практичного завдання

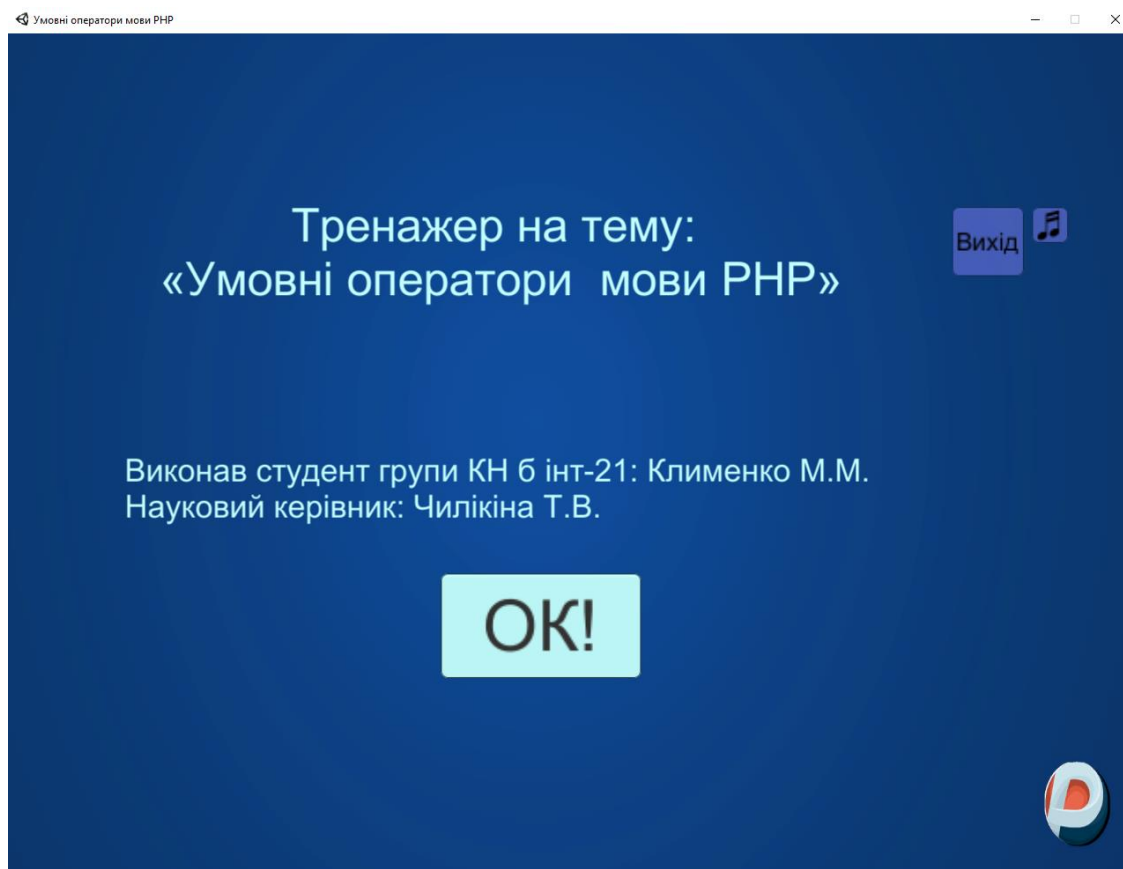
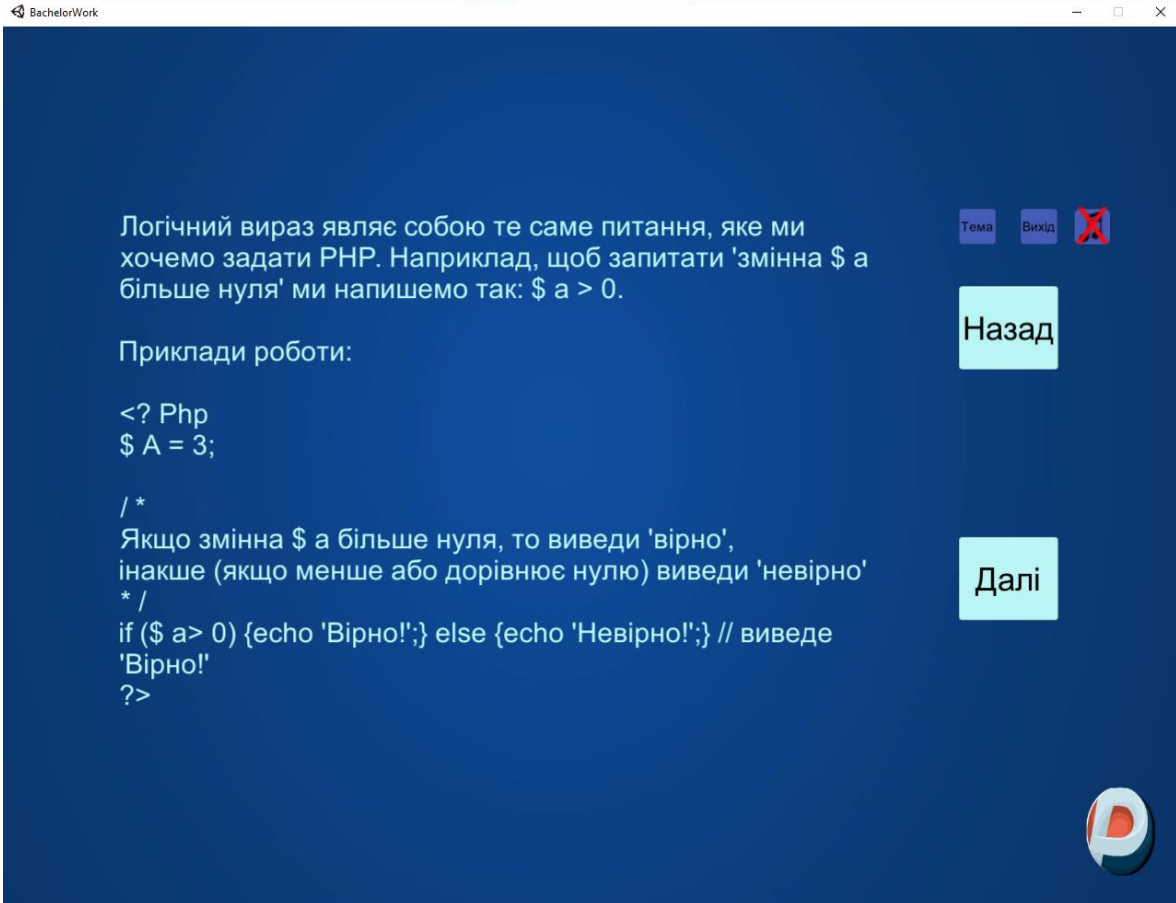


Рисунок 4.9 – Вікно з інформацією про тренажер

Приклад подачі теоретичної інформації:



The screenshot shows a window titled "BachelorWork" with a dark blue background. The main text explains that a logical expression is a question about whether a variable is greater than zero, and provides a PHP example: `$ a > 0`. Below this, it says "Приклади роботи:" and shows a PHP code snippet: `<? Php $ A = 3; /* Якщо змінна $ a більше нуля, то виведи 'вірно', інакше (якщо менше або дорівнює нулю) виведи 'невірно' */ if ($ a > 0) {echo 'Вірно!';} else {echo 'Невірно!';} // виведе 'Вірно!' ?>`. On the right side, there are three small buttons: "Тема", "Вихід", and a red "X" icon. Below these are two larger buttons: "Назад" and "Далі". In the bottom right corner, there is a circular logo with a stylized "P" and "D".

Рисунок 4.10 – Подача тренажером завдань теоретичного типу

Після закінчення роботи з завданням теоретичного типу для пришвидшення роботи реалізовано кнопку переходу до практики що перенаправляє користувача на слайд з вибором тем для практичного опрацювання (див. Рисунок 4.7)

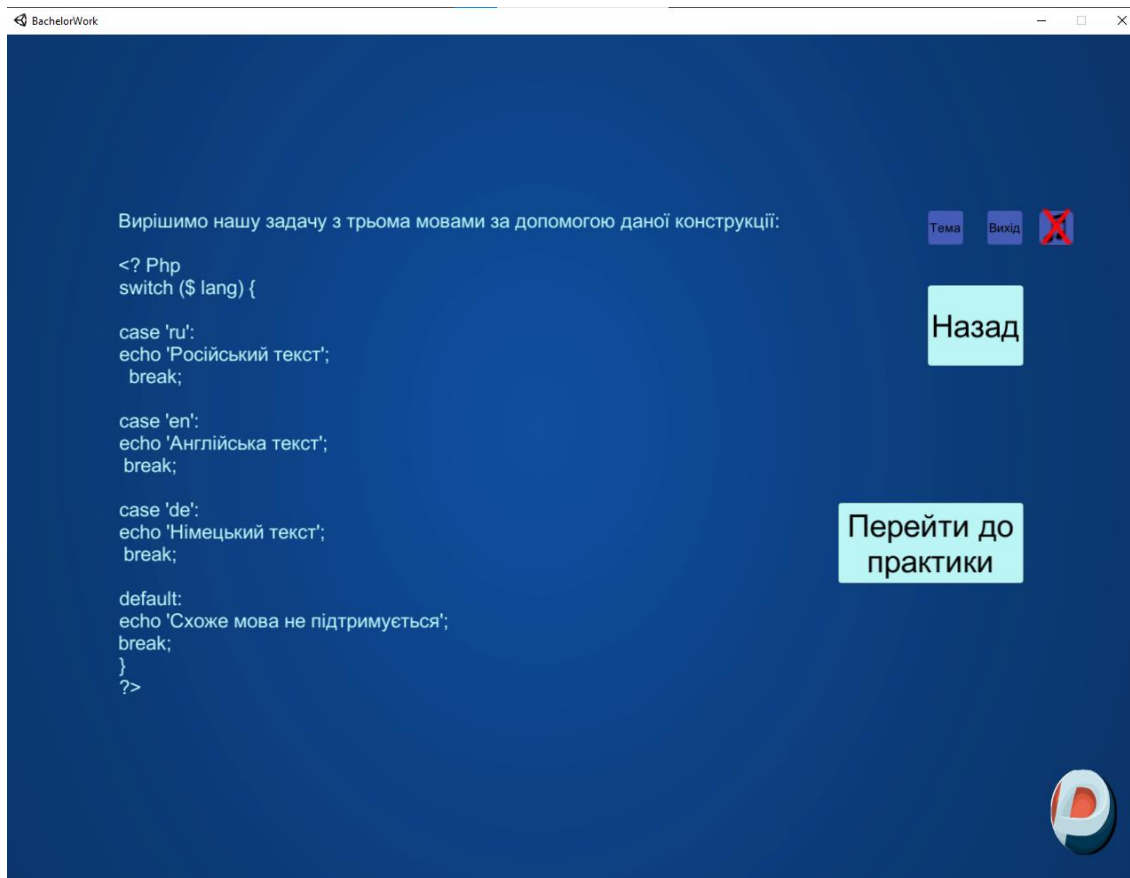


Рисунок 4.11 – Кнопка переходу до практики

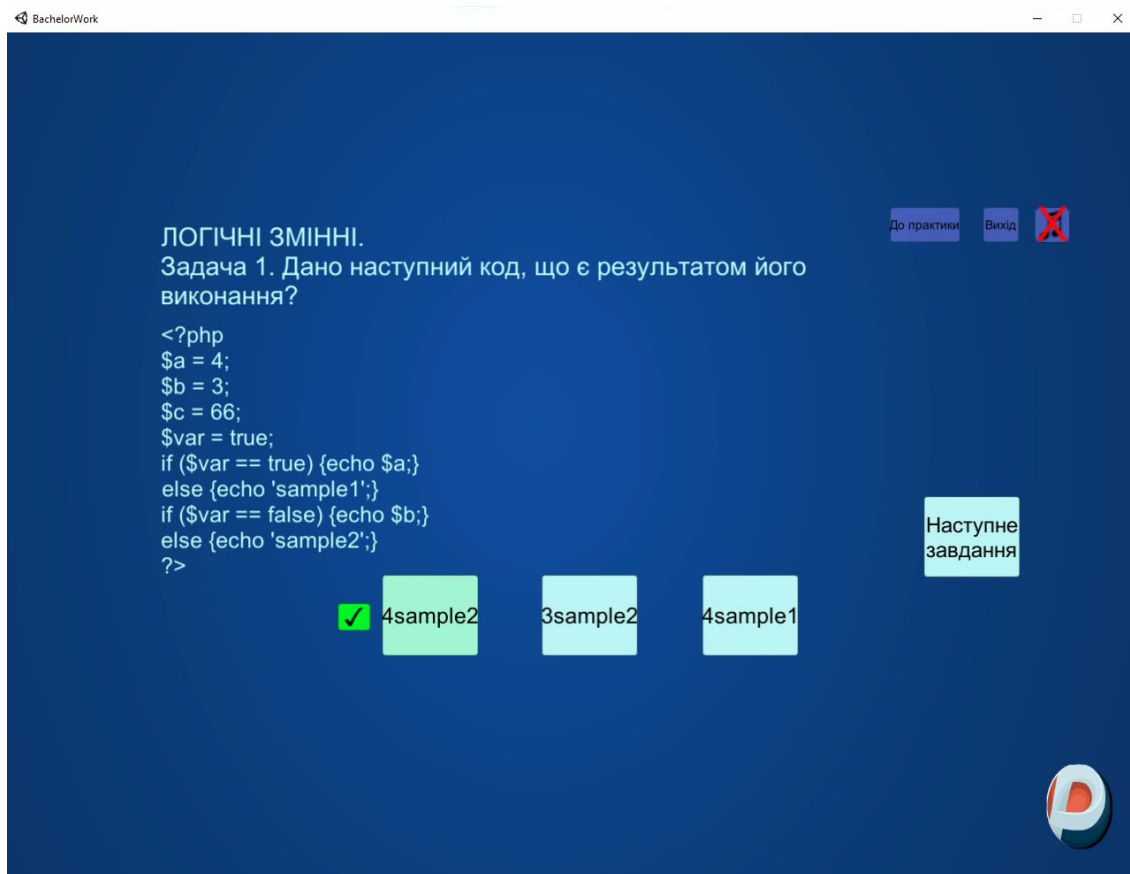


Рисунок 4.12 – Вікно після вибору правильної відповіді

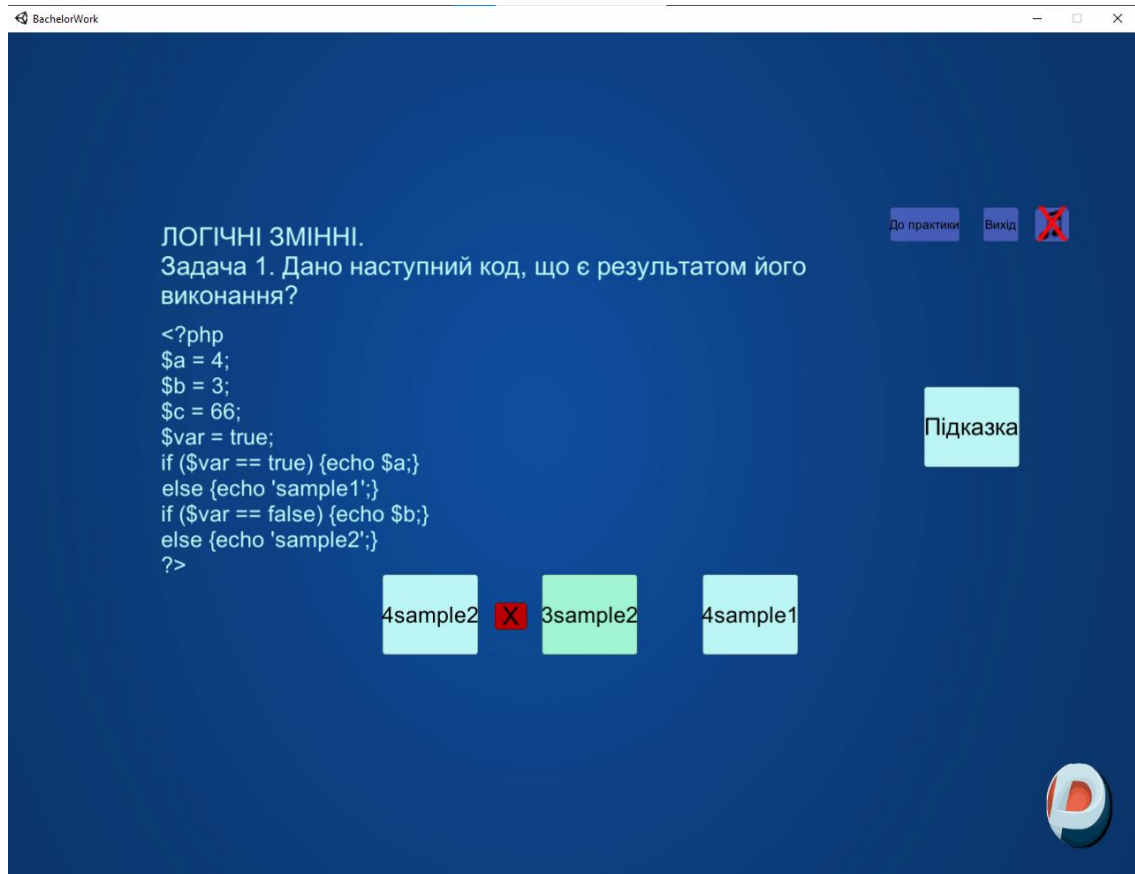


Рисунок 4.13 – Вікно після вибору неправильної відповіді

При другому варіанті роботи з тренажером користувач отримує завдання та поле для вводу інформації (див. Рисунок 4.14). Після вводу певної відповіді та натиснення на кнопку «Ввести дані» користувач отримує правильну відповідь, свою відповідь та доступ до кнопки «Підказка» (див. Рисунок 4.15).

Для роботи завдань з полем для вводу було використано скрипт з наступним кодом:

```
public class Inputs : MonoBehaviour
{
    public GameObject Input;
    public GameObject AfterInput;
    public InputField inputedTxt;
    public Text ShowInpTxt;
    public void Next()
    {
        Input.SetActive(false);
```

```
AfterInput.SetActive(true);  
ShowInpTxt.text = inputedTxt.text;  
}  
}
```

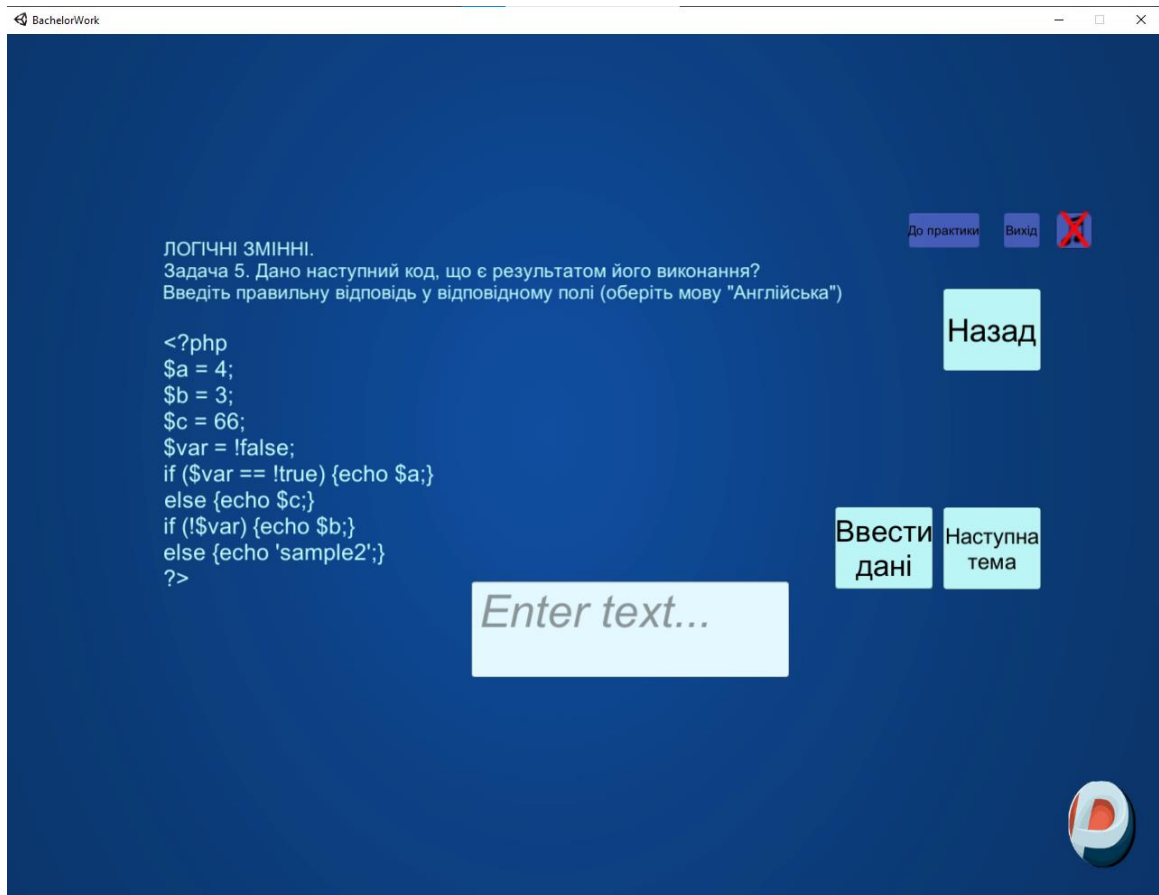


Рисунок 4.14 – Зовнішній вигляд практичного завдання з вводом інформації

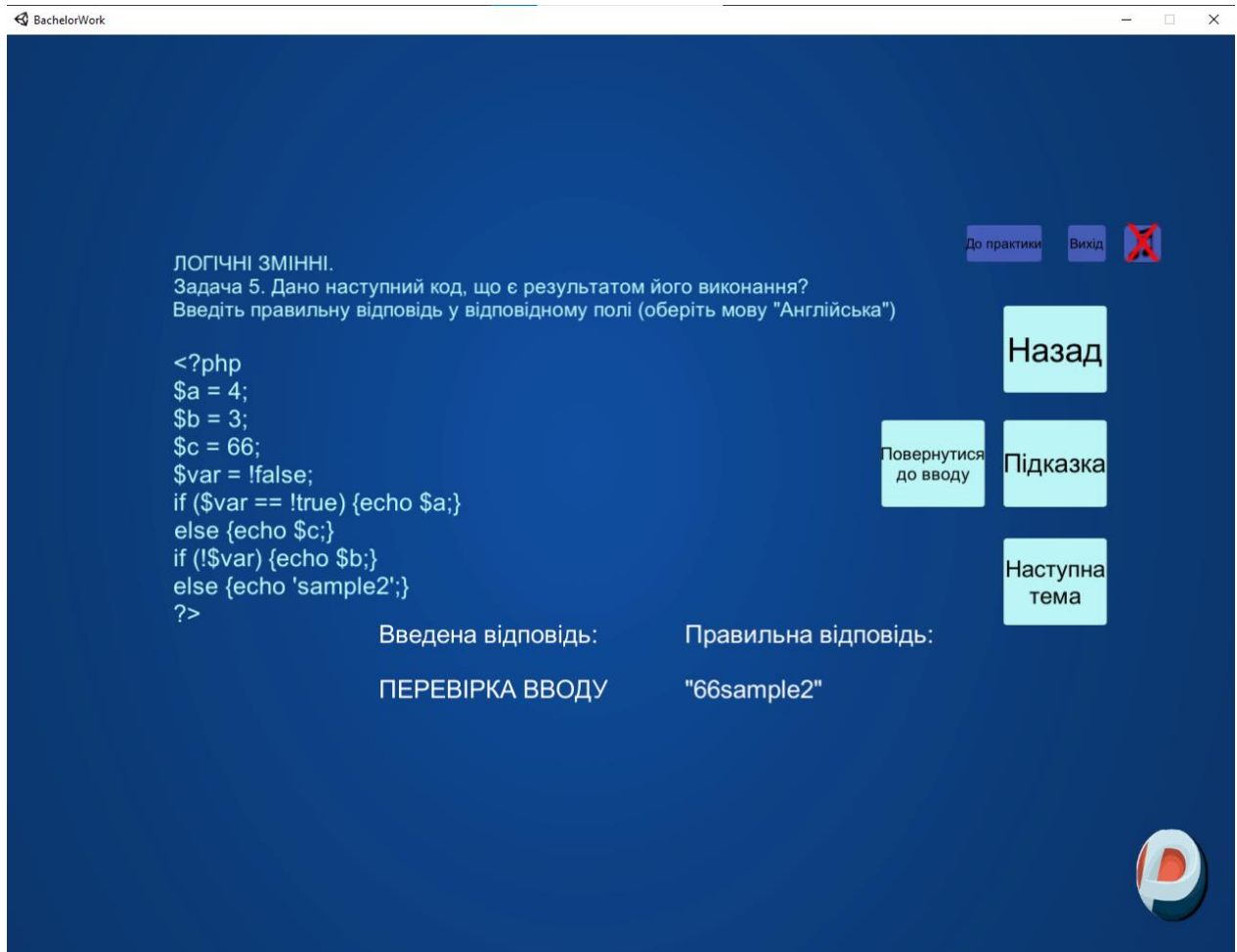


Рисунок 4.15 – Вікно після вводу відповіді

Кнопка «Підказка» відкриває вікно що накладається на завдання та виводить певну корисну та коротку інформацію про завдання (див. Рисунок 4.14). Також реалізована кнопка переходу до завдань теоретичного типу та кнопка повернення до практичного завдання (див. Рисунок 4.16).

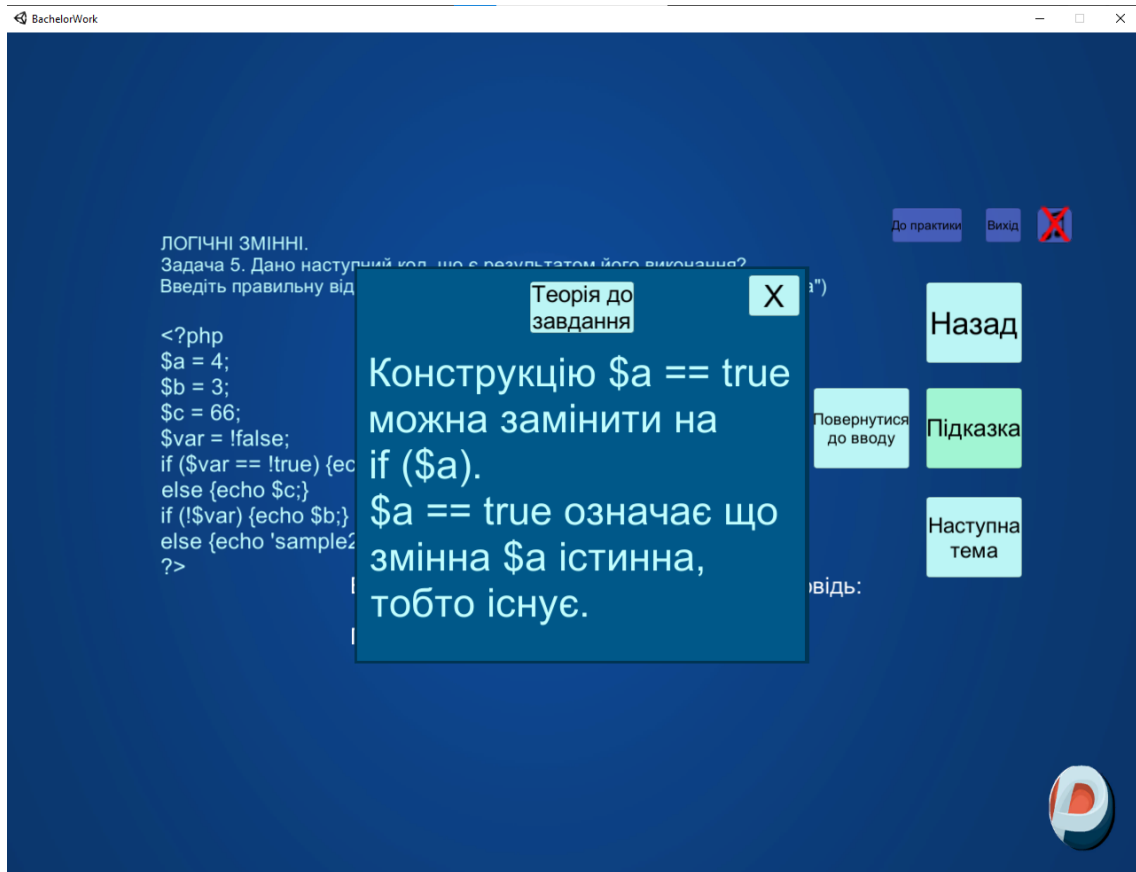


Рисунок 4.16 – Вікно «Підказка»

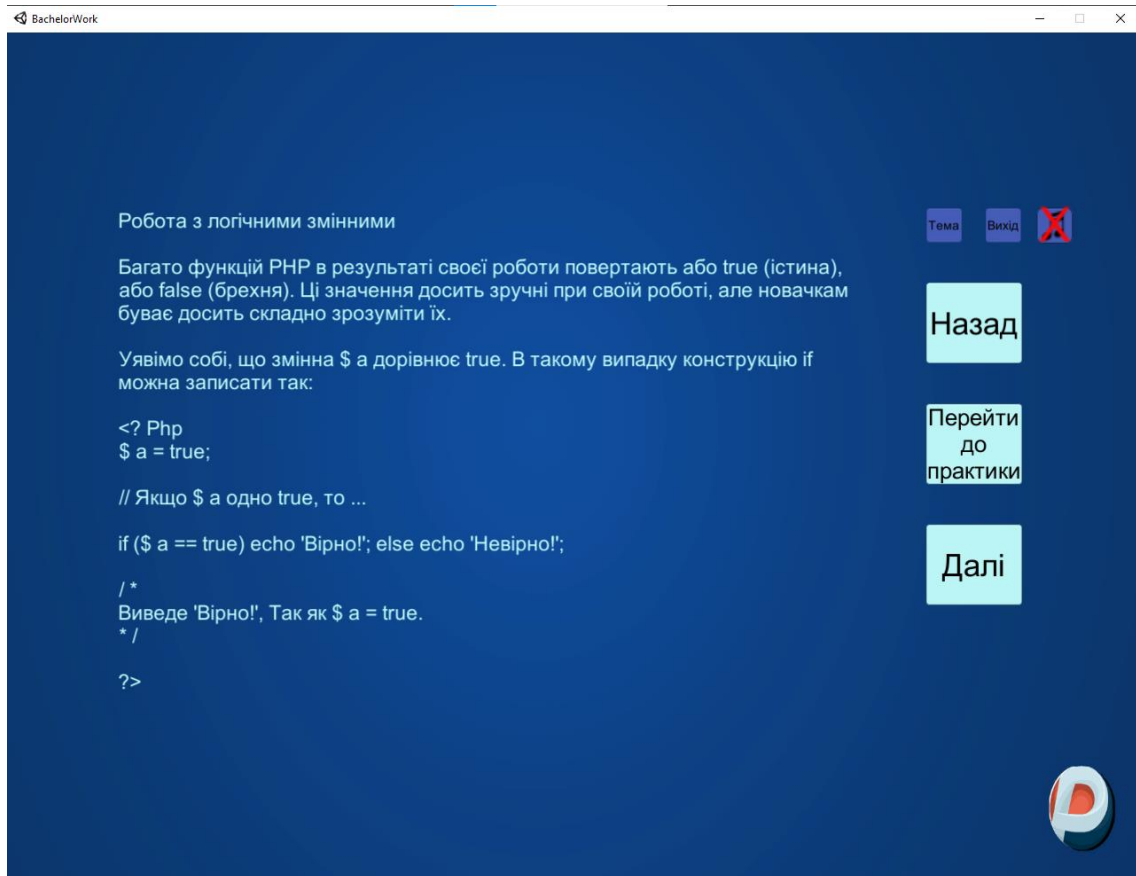


Рисунок 4.17 – Відповідний слайд з завданням теоретичного типу з кнопкою повернення до практичного завдання

4.2 Перевірка валідності (тестування всіх процедур)

Запуск тренажера відбувається без помилок (див. Рисунок 4.18).

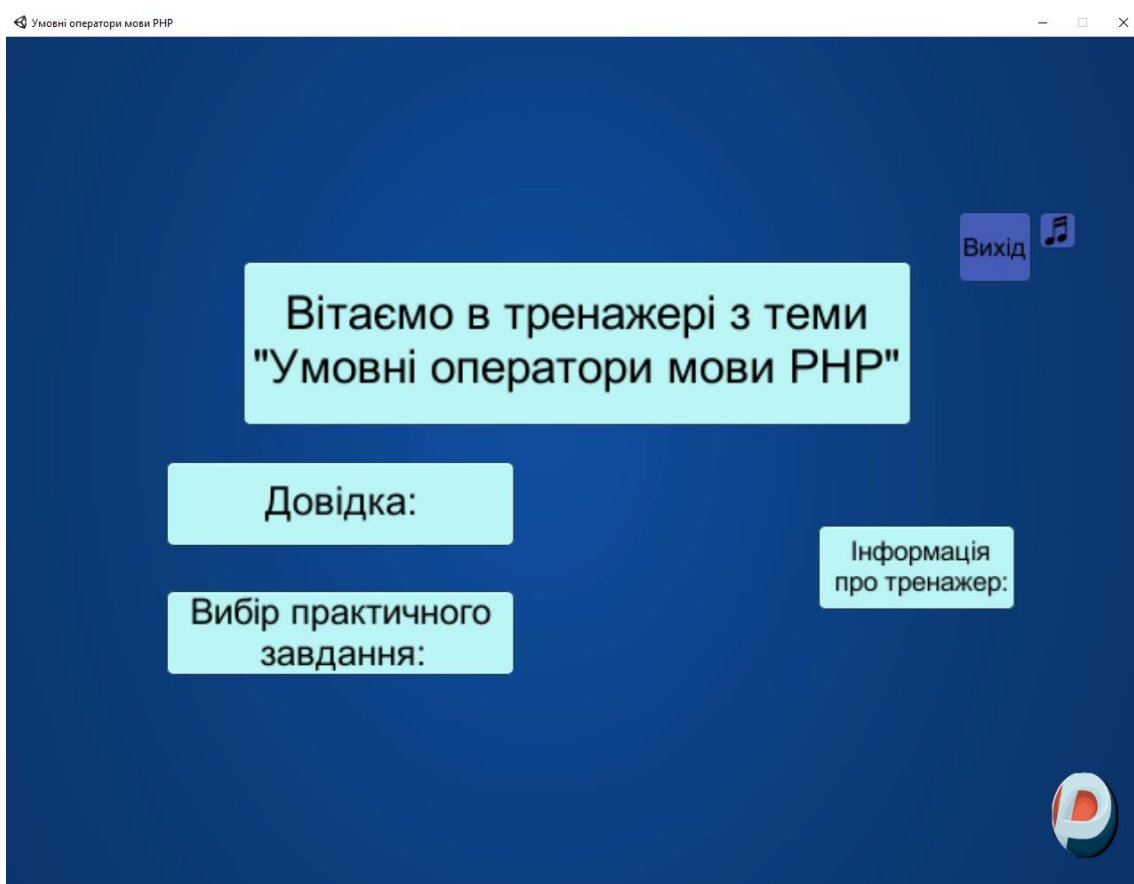


Рисунок 4.18 – Запуск тренажера

Після натиснення на кнопку «Вибір теми» та подальшого вибору теми (див. Рисунок 4.19) отримуємо доступ до теоретичного матеріалу (див. Рисунок 4.20).

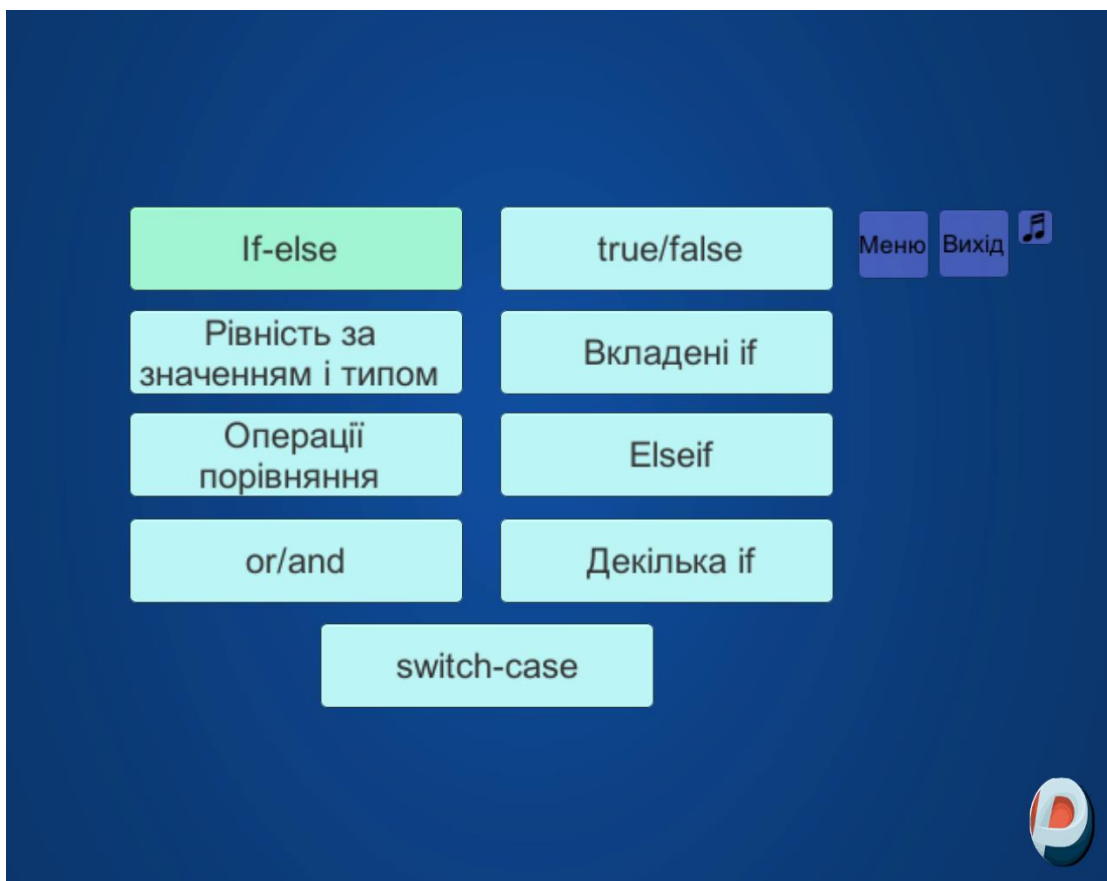


Рисунок 4.19 – Вибір теоретичної теми

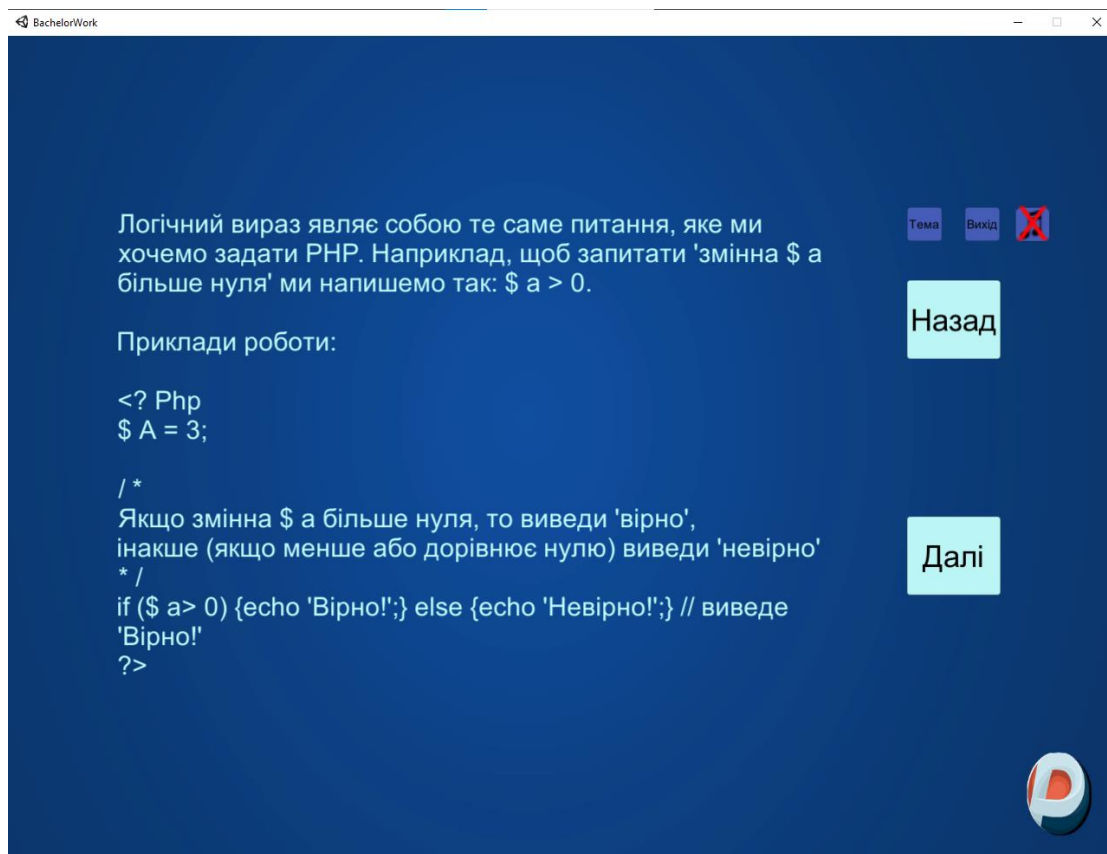


Рисунок 4.20 – Перший слайд з завданням теоретичного типу

Після натиснення на кнопку «Тема» у верхньому правому куті, отримуємо доступ до всіх теоретичних тем (див. Рисунок 4.21).

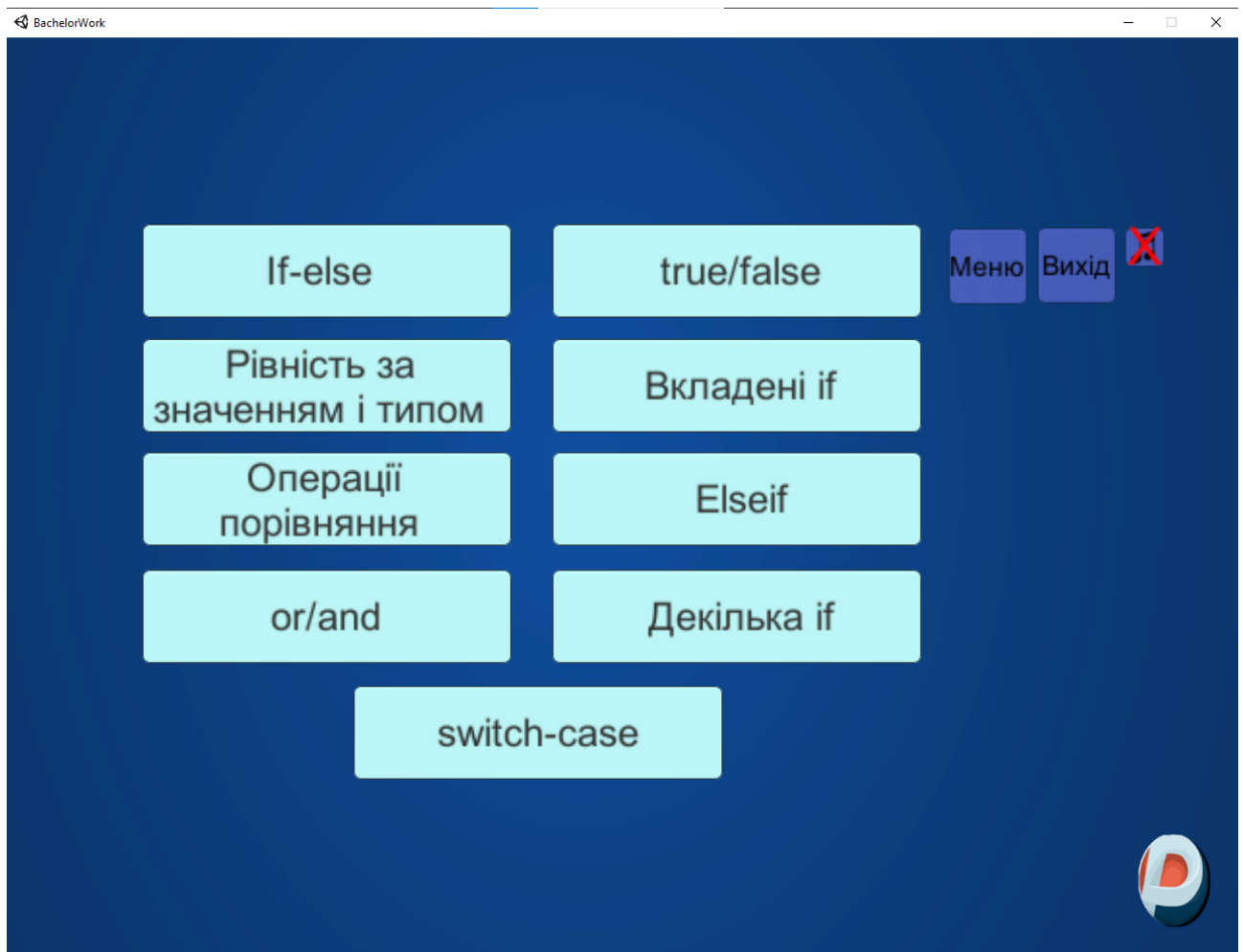


Рисунок 4.21 – Повернення до вибору теми під час роботи з завданням теоретичного типу

Для доступу до практичних завдань під час роботи з тренажером достатньо натиснути кнопку «Меню», перейти до вибору практичного завдання та обрати відповідну тему. При виборі завдання з інпутом, виводиться нагадування про необхідність змінити мову на англійську.

Робота з тренажером типова на всіх пристроях на платформі Windows.

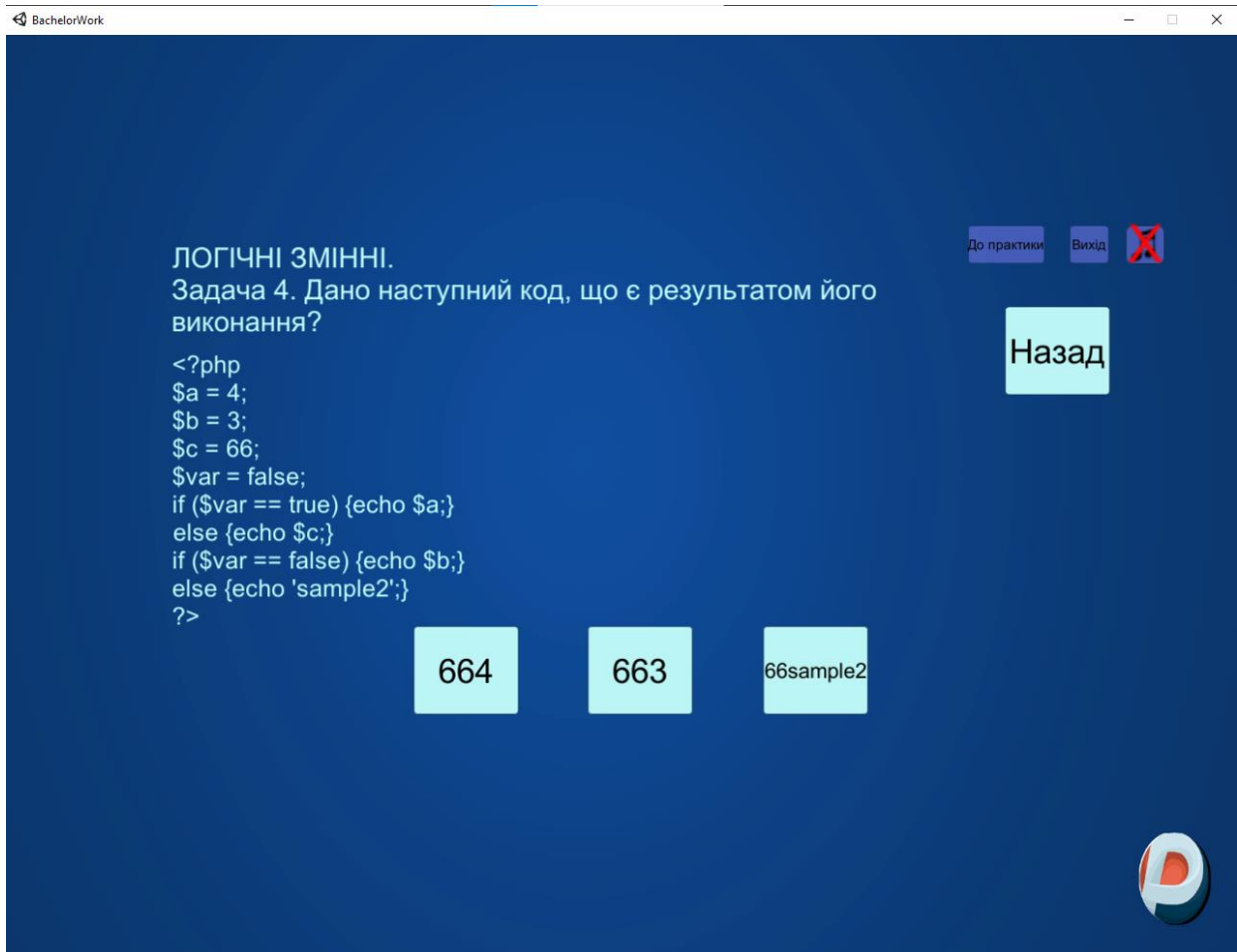


Рисунок 4.22 – Вибір практичного завдання з меню

З будь-якого практичного завдання є можливість перейти до списку всіх практичних завдань, або повернутись в головне меню.

При виборі неправильної відповіді зникає кнопка «Далі», а при виборі правильної зникає кнопка «Підказка».

Під час роботи з тренажером існує можливість включити аудіо-супровід через відповідну кнопку (див. Рисунок 4.23)



Рисунок 4.23 – Кнопка для ввімкнення аудіо-супроводу в двох положеннях

ВИСНОВКИ

Створена на основі сучасних комп'ютерних технологій програма-тренажер, навчає студентів темі «Умовні оператори мови PHP», має теоретичну та практичну частину та зручний перехід між ними.

При виконанні даної бакалаврської роботи на основі поставленої задачі було реалізовано та алгоритмізовано тренажер для дистанційного навчального курсу з теми «Умовні оператори мови PHP».

Під час виконання бакалаврської роботи з фаху було розроблено тренажер з теми «Умовні оператори мови PHP» яке може бути використано для дистанційного курсу.

Результатами бакалаврської роботи є:

1. Обґрунтовано вибір мови та середовища розробки, яке сумісне з системою дистанційного навчання MOODLE;
2. Обрано програмне середовище – MS Visual Studio та мову програмування C#;
3. Обрано платформу для розробки програмного забезпечення – Unity 2020;
4. Розроблено алгоритм роботи тренажера з теми «Умовні оператори мови PHP» для дистанційного курсу;
5. Складено блок-схеми до алгоритму з врахуванням всіх можливих варіантів роботи;
6. Програмно реалізовано програму-тренажер на платформі Unity 2020 у середовищі IDE MS Visual Studio за допомогою мови програмування C#;
7. Виконано перевірку валідності програми. Помилки в роботі тренажера не знайдено.

Позитивними сторонами розробленого програмного забезпечення є:

1. Сучасний дизайн та інтуїтивно зрозумілий інтерфейс програми;

2. Передбачено перевірку на правильність вводу даних та вивід відповідного повідомлення. Також можливою є функція повернення до теоретичного матеріалу з типовим практичним завданням;

3. Покращена навігація при роботі з тренажером, що дозволяє зручно та швидко орієнтуватися в навчальних матеріалах;

4. Додано аудіо-супровід до роботи з тренажером, також додано можливість вимкнути його через відповідну кнопку;

Мету та завдання бакалаврської роботи виконано, розроблено програмне забезпечення у вигляді тренажеру з теми «Умовні оператори мови PHP».

Результати бакалаврської роботи опубліковані в матеріалах XLIV Міжнародної наукової конференції «АКТУАЛЬНІ ПИТАННЯ РОЗВИТКУ НАУКИ ТА ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ОСВІТИ У XXI СТОЛІТТІ».

Спроектований тренажер може бути впроваджено в систему дистанційного навчання ПУЕТ в дистанційний курс «Проектування та підтримка веб-застосувань», про що є акт про впровадження.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гусак, Ю. С. Пояснювальна записка до дипломної роботи на тему Розробка тренажеру з теми «Моделювання булевих функцій за допомогою елементарного перцептронну» дисципліни «Нейронно-мережеві технології в інформатиці»/ Гусак Ю.С., [Електронний ресурс].– Режим доступу до ресурсу: <http://dspace.puet.edu.ua/handle/123456789/10043>
2. Рзаев, Э. Ф., Пояснительная записка к бакалаврской работе на тему Разработка программного обеспечения для тренажера по теме «Магазинные автоматы» дистанционного учебного курса «Теория программирования»/ Рзаев, Э.Ф., [Електронний ресурс].– Режим доступу до ресурсу: <http://dspace.puet.edu.ua/handle/123456789/9021>
3. Мандя, О. О., Тренажер з теми «Складання математичної моделі»/ Мандя О.О., [Електронний ресурс].– Режим доступу до ресурсу: <http://dspace.puet.edu.ua/handle/123456789/1718>
4. Скромінський, М. В., Пояснювальна записка до бакалаврської роботи на тему Розробка програмного забезпечення для тренажеру з теми «Контекстовільні граматики» дистанційного навчального курсу «Теорія програмування»/ Скромінський, М. В., [Електронний ресурс].– Режим доступу до ресурсу: <http://dspace.puet.edu.ua/handle/123456789/9025>
5. Трепачёв Д. Учебник PHP для новичков от Трепачёва Дмитрия / Дмитрий Трепачёв. – Режим доступу до ресурсу: <http://old.code.mu/books/php/base/rabota-s-konstrukciyami-if-else-switch-case-v-php.html>
6. Трепачёв Д. Дмитрий Трепачев підручник PHP/SQL для новачків, практичні завдання / Дмитрий Трепачёв. – Режим доступу до ресурсу: <http://old.code.mu/tasks/php/base/rabota-s-konstrukciyami-if-else-switch-case-v-php.html>

7. Unity (Рушій гри) [Електронний ресурс].– Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/Unity_\(%D1%80%D1%83%D1%88%D1%96%D0%B9_%D0%B3%D1%80%D0%B8\)](https://uk.wikipedia.org/wiki/Unity_(%D1%80%D1%83%D1%88%D1%96%D0%B9_%D0%B3%D1%80%D0%B8))
8. IDE Visual Studio [Електронний ресурс].– Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio
9. Уроки по роботі з IDE Visual Studio [Електронний ресурс].– Режим доступу до ресурсу: <http://www.codenet.ru/progr/vbasic/Visual-Basic-NET/>
10. Working with scripts [Електронний ресурс].– Режим доступу до ресурсу: <https://learn.unity.com/tutorial/working-with-scripts?language=en#>
11. Гоша Дударь, Відео-уроки по Unity, С# / Гоша Дударь [Електронний ресурс].–Режим доступу до ресурсу: <https://www.youtube.com/watch?v=GGsOU7sP0r4>
12. Уроки по роботі з PHP [Електронний ресурс].– Режим доступу до ресурсу: <http://javaphp.ptngu.com/phplessons/phplesson4>
13. Уроки по роботі з PHP [Електронний ресурс].– Режим доступу до ресурсу: <http://www.e-helper.com.ua/node/107>
14. Уроки по роботі з PHP [Електронний ресурс].– Режим доступу до ресурсу: https://addphp.ru/materials/base/1_6.php

ДОДАТОК А

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine.UI;
using UnityEngine;

public class Inputs : MonoBehaviour
{
    public GameObject Input;
    public GameObject AfterInput;

    public InputField inputedTxt;
    public Text ShowInpTxt;

    public void Next()
    {
        Input.SetActive(false);
        AfterInput.SetActive(true);

        ShowInpTxt.text = inputedTxt.text;
    }
}

public class QuitButton : MonoBehaviour
{
    public void QuitGame()
    {
```

```
    Debug.Log ("QUIT!!");  
    Application.Quit();  
}
```

```
void Start()  
{  
    Button btn = nxtButton.GetComponent<Button>();  
    btn.onClick.AddListener(TaskOnClick);  
}
```

```
void TaskOnClick()  
{  
    Theme1.SetActive(false);  
    Theme2.SetActive(true);  
  
    Debug.Log("You touched this button.");  
}
```

```
void Start()  
{  
    Button btn = menuButton.GetComponent<Button>();  
    btn.onClick.AddListener(TaskOnClick);  
}
```

```
void TaskOnClick()  
{  
    Theme6.SetActive(false);  
    MainMenu.SetActive(true);  
}
```

```
        Debug.Log("You touched this button.");  
    }  
  
    public void Next()  
    {  
        Input.SetActive(false);  
        AfterInput.SetActive(true);  
  
        ShowInpTxt.text = inputdTxt.text;  
    }  
}
```