ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ
«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

ТРЕНАЖЕР З ТЕМИ «ПОБУДОВА БЛОК-СХЕМ АЛГОРИТМІВ ЦИКЛІЧНОЇ СТРУКТУРИ НА ПРИКЛАДІ ЦИКЛУ REPEAT…UNTIL» ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО КУРСУ «ІНФОРМАТИКА» ТА РОЗРОБКА ЙОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

зі спеціальності 122 «Комп'ютерні науки»

Виконавець роботи Рижков Артем Віталійович _____ «___»_____2021 р.
(підпис)

Науковий керівник к.ф.-м.н., проф., Ємець Єлизавета Михайлівна
_____ «___»_____ 2021 р.
(підпис)

ПОЛТАВА 2021 р.

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСПІЛКИ
«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»

ЗАТВЕРДЖУЮ

Завідувач кафедри _____ О.О. Ємець
(підпис)

«_____» _____20__ р.

ЗАВДАННЯ ТА КАЛЕНДАРНИЙ ГРАФІК

ВИКОНАННЯ БАКАЛАВРСЬКОЇ РОБОТИ

Студента зі спеціальності 122 «Комп'ютерні науки»

Прізвище, ім'я, по батькові Рижков Артем Віталійович

1. Тема «Тренажер з теми «Побудова блок-схем алгоритмів циклічної структури на прикладі циклу repeat…until» дистанційного навчального курсу «Інформатика» та розробка його програмного забезпечення», затверджена наказом ректора № 121-Н від «1» вересня 2020 р.

Термін подання студентом бакалаврської роботи «24» травня 2021 р.

2.Вихідні дані до бакалаврської роботи: публікації за темою роботи; методичні рекомендації; стандарти.

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

Вступ.
1. Постановка задачі.
2. Інформаційний огляд.
   2.1. Огляд програм, аналогічних поставленій темі.
   2.2. Позитивні риси переглянутих програм.
   2.3. Негативні риси переглянутих програм.
   2.4. Необхідність та актуальність теми.
3. Теоретична частина.
   3.1. Алгоритм тренажеру.
      3.1.1. Приклад № 1.
      3.1.2 Приклад № 2.
   3.2. Блок-схема алгоритму.
4. Практична частина.
   4.1. Опис програмної реалізації тренажеру.
   4.2. Інструкція по роботі з програмою.

Висновки.

4. Перелік графічного матеріалу (з точним визначенням кількості блок-схем, іншого графічного матеріалу) Блок-схема алгоритму (4-6 листів).

5. Консультанти розділів бакалаврської роботи

Розділ	П.І.Б., посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Вступ	Ємець Є. М.	05.09.2020	05.09.2020
1. Постановка задачі	Ємець Є. М.	05.09.2020	05.09.2020
2. Інформаційний огляд	Ємець Є. М.	05.09.2020	05.09.2020
3. Теоретична частина	Ємець Є. М.	05.09.2020	05.09.2020
4. Практична частина	Ємець Є. М.	05.09.2020	05.09.2020

6. Календарний графік виконання бакалаврської роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ	04.05.21	
2. Вивчення методичних рекомендацій і стандартів та звіт керівнику	1.10.20	
3. Постановка задачі	10.10.20	
4. Інформаційний огляд джерел бібліотек та інтернету	1.11.20	
5. Теоретична частина	13.01.21	
6. Практична частина	15.04.21	
7. Закінчення оформлення	04.05.21	
8. Доповідь студента на кафедрі	25.05.21	
9. Доробка (за необхідності), рецензування	07.06.21	

Дата видачі завдання «5» вересня 2020 р.

Студент _____ Рижков А. В.
(підпис)

Науковий керівник _____ к. ф.-м. н., проф. Є. М.
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Результати захисту бакалаврської роботи

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

Протокол засідання ЕК № ____ від «____» _____ 20___ р.

Секретар ЕК_____ _____
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Записка: 45 с., 50 рис., 4 додатка (на 47 сторінках), 12 джерел.

Предмет розробки – електронний симулятор з теми «Побудова блок-схем алгоритмів циклічної структури на прикладі циклу repeat…until» для дистанційного курсу ПУЕТ «Інформатика. Частина 1».

Мета роботи – розробити та програмно реалізувати симулятор для теми «Побудова блок-схем алгоритмів циклічної структури на прикладі циклу repeat … until».

Методи розробки – мова програмування С++, середовище програмування Borland Builder, пакет MS Visio.

Створено алгоритм тренажеру з двох прикладів. У першому прикладів 11 кроків, у другому – 17 кроків.

Розроблено програмну реалізацію симулятору з теми «Побудова блок-схем алгоритмів циклічної структури на прикладі циклу repeat…until».

Ключові слова: СИМУЛЯТОР, АЛГОРИТМ, БЛОК-СХЕМА, ЦИКЛ, REPEAT ... UNTIL.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ …….. 6

ВСТУП …….. 7

1. ПОСТАНОВКА ЗАДАЧІ …… 8

2. ІНФОРМАЦІЙНИЙ ОГЛЯД …… 9

2.1. Огляд програм, аналогічних поставленій темі ……..…….. 9

2.2. Позитивні риси переглянутих програм ……..…….. 13

2.3. Негативні риси переглянутих програм ……..…… 13

2.4. Необхідність та актуальність теми …… 13

3. ТЕОРЕТИЧНА ЧАСТИНА …….. 14

3.1. Алгоритм тренажеру …… 14

3.1.1. Приклад № 1 …… 14

3.1.2. Приклад № 2 …… 25

3.2. Блок-схема алгоритму …… 26

4. ПРАКТИЧНА ЧАСТИНА ……..... 27

4.1. Опис програмної реалізації тренажеру …….. 27

4.2. Інструкція по роботі з програмою ……..…… 29

ВИСНОВКИ ……. 43

СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ ……... 44

ДОДАТОК А. АЛГОРИТМ ТРЕНАЖУР ПРИКЛАДУ № 2 …… 46

ДОДАТОК Б. КОПІЇ ЕКРАНІВ ПРИКЛАДУ № 2 …..……..…… 68

ДОДАТОК В. ПРОДОВЖЕННЯ БЛОК-СХЕМИ …..……..……. 85

ДОДАТОК Г. ПРОГРАМНИЙ КОД ……..…… 88

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Умовні позначення, символи, одиниці, скорочення, терміни	**Пояснення умовних позначень, символів, одиниць, скорочень, термінів**
Алгоритм	Чіткий опис послідовних дій, які необхідно виконати для розв’язання задачі.
Блок-схема	Наочне графічне зображення алгоритму, в якому окремі етапи алгоритму зображуються за допомогою різних геометричних фігур, а зв’язки між етапами зображуються за допомогою стрілок, що з’єднують ці фігури. Побудова блок-схем регламентуються стандартом ГОСТ 19.701-90.
Цикл repeat … until	Цикл, який виконується до тих пір, доки умова, зазначена біля слова until хибна.

ВСТУП

Актуальність теми. Науково-технічний прогрес, поширення інтернету призвели до можливості самостійного навчання через персональні комп'ютери, ноутбуки, планшети і навіть смартфони. Для ефективного застосування дистанційного навчання є потреба у розробці програм, які навчають студента розв'язуванню типових задач з цієї дисципліни. Такі програми називаються ***симуляторами*** або ***тренажерами***.

Крім того, є і економічна складова. Створити симулятори і навчати велику кількість учнів, навчати декілька поколінь студентів є дешевшим, ніж платити заробітну плату вчителям та викладачам. Крім того, деякі країни мають дефіцит вчителів (викладачів). Наприклад, за відомостями BBC у такий високо розвинутій країні як Великобританія є дефіцит у 40 тисяч вчителів. Підсумовуючи, створення електронних засобів навчання, в тому числі симуляторів, є актуальною задачею.

Об'єкт розробки – електронний симулятор.

Предмет розробки – електронний симулятор з теми «Побудова блок-схем алгоритмів циклічної структури на прикладі циклу repeat…until» для дистанційного курсу ПУЕТ «Інформатика. Частина 1».

Мета випускової роботи – розробити та програмно реалізувати симулятор для теми «Побудова блок-схем алгоритмів циклічної структури на прикладі циклу repeat … until».

Методами розробки є мова програмування C++, середовище програмування Borland Builder, пакет MS Visio.

Структура записки. Записка складається з трьох основних складових. У першій – висвітлено постановку задачі. У другій – описано існуючі аналогічні розробки, проведено їх аналіз. У третій – сформульовано алгоритм тренажеру та подано блок-схему алгоритму. У четвертій – описано створену програму. ***Обсяг записки*** – 45 сторінок.

1. ПОСТАНОВКА ЗАДАЧІ

В рамках бакалаврської роботи слід виконати наступне.

Вивчити правила побудови блок-схем. Для цього у нагоді стане дистанційний курс «Інформатика. Частина 1».

Вивчити правила написання програм мовою Object Pascal з використанням циклічного оператору repeat … until. Для цього у нагоді стане той самий дистанційний курс та інші джерела.

Вивчити доступні симулятори з подібної теми. Якщо симуляторів з аналогічної теми немає, то ознайомитись з будь-якими тренажерами, які є в наявності. Проаналізувати розробки, які їх переваги, недоліки, які ідеї можна використати у власному тренажері.

Підібрати два приклади програми на repeat … until. Впевнитись в коректній роботі цих програм. На базі прикладів розробити алгоритм симулятору. Для алгоритму побудувати блок-схему.

Здійснити програму реалізацію алгоритму симулятору.

У програмі умови прикладів повинна бути видимі на усіх кроках тренінгу. Програма повинна на кожному кроці або підтверджувати вірність відповіді користувача, або пояснювати його помилку. Програма повинна містити ілюстрації, що показують етапи побудови блок-схеми.

Перевірити роботу програми. У випадку помилок чи неточностей, усунути їх.

Створити супровідну документацію по програмі.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Огляд програм, аналогічних поставленій темі

Розглянемо тренажери подібної тематики.

1) У 2021 р. студенткою спеціальності «Комп'ютерні науки та інформаційні технології» ПУЕТ Сузанською А. О. був створений тренажер ***«Побудова блок-схем алгоритмів розгалуженої структури»*** для курсу «Програмування ІІ» (рис. 2.1-2.3). Симулятор містить два приклади на структури if, if-else. За викладений кодом на мові С++ слід створити блок-схему. На кожному кроці користувач обирає потрібний символ, заповнений текстом, або фрагмент блок-схеми. Після цього з'являється побудована на цей момент блок-схема алгоритму.

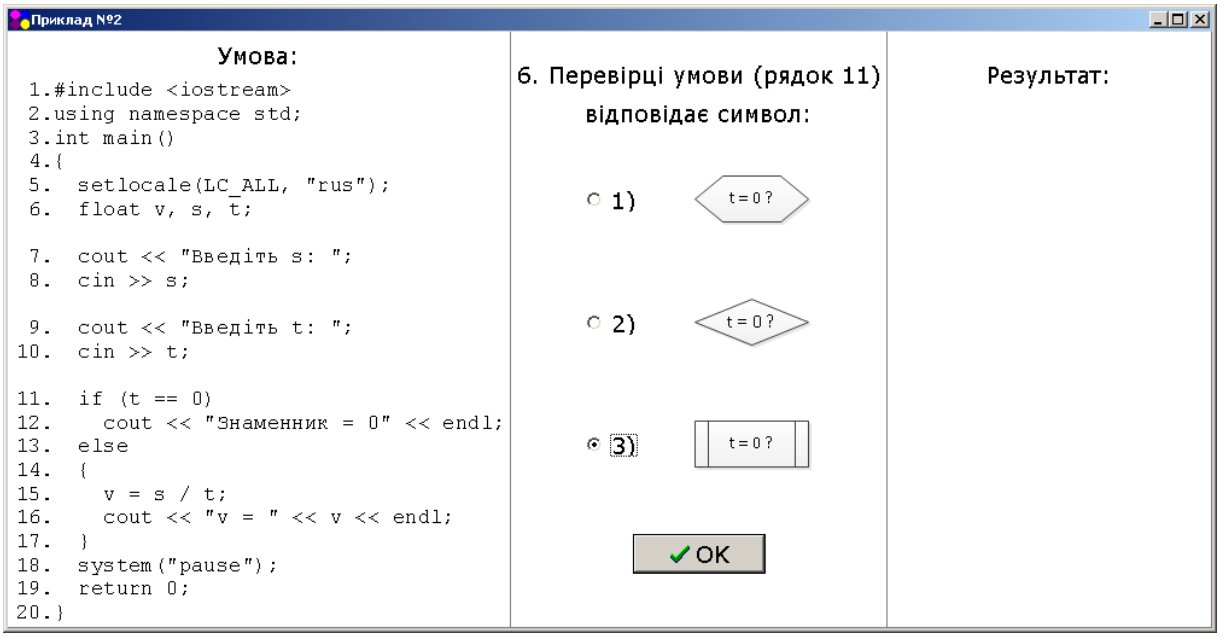

Рисунок 2.1 – Умова шостого кроку другого прикладу (тренажер 1)

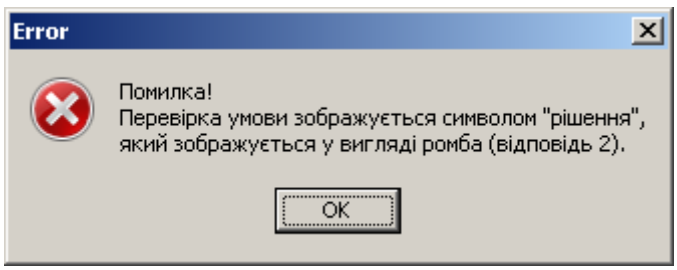

Рисунок 2.2 – Пояснення помилки

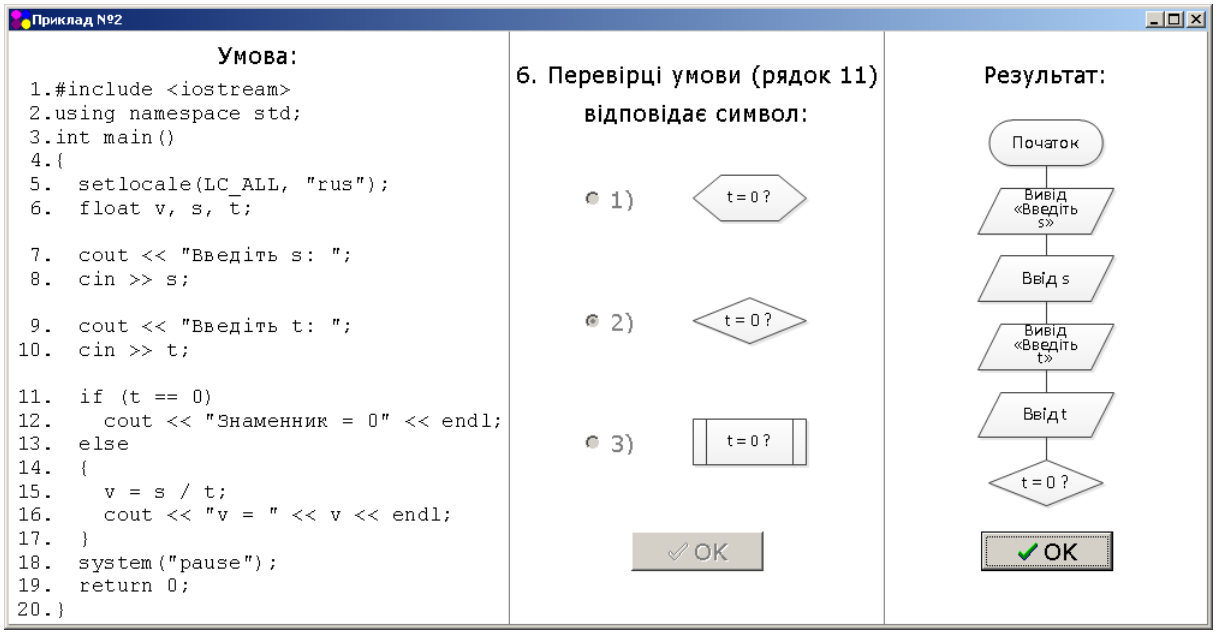

Рисунок 2.3 – Після шостого кроку

2) Студентом спеціальності «Комп'ютерні науки» ПУЕТ Олефіренком В. В. у 2021 р. був створений тренажер ***«Сортування включеннями»*** для курсу «Алгоритми і структури даних» (рис. 2.4-2.6). Симулятор містить ряд питань по сортуванню – теоретичні, з використанням блок-схем алгоритмів.

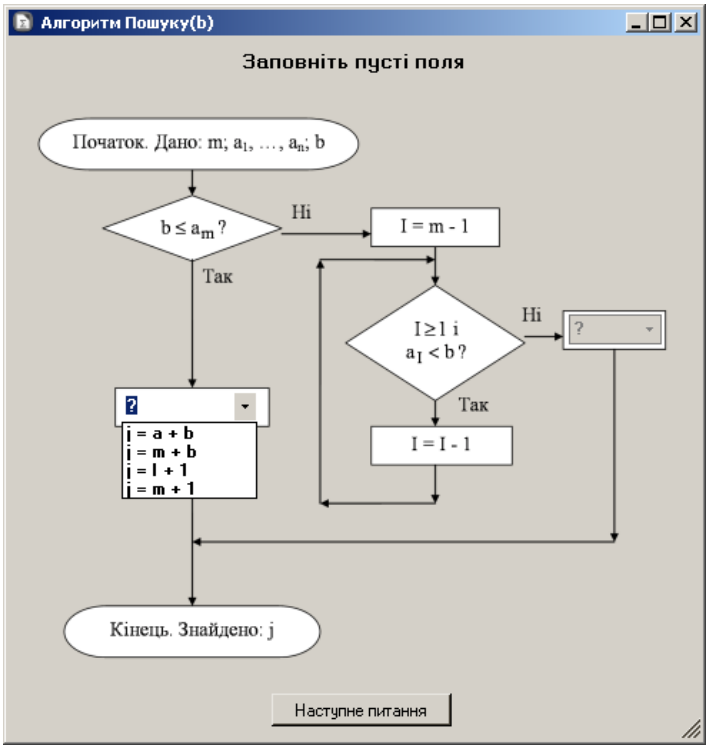

Рисунок 2.4 – Умова другого кроку (тренажер 2)

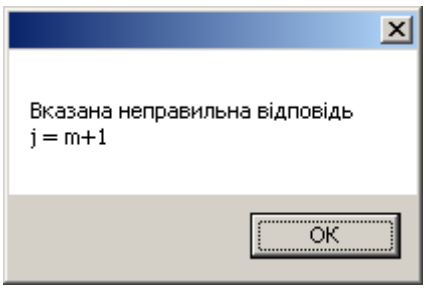

Рисунок 2.5 – Пояснення помилки

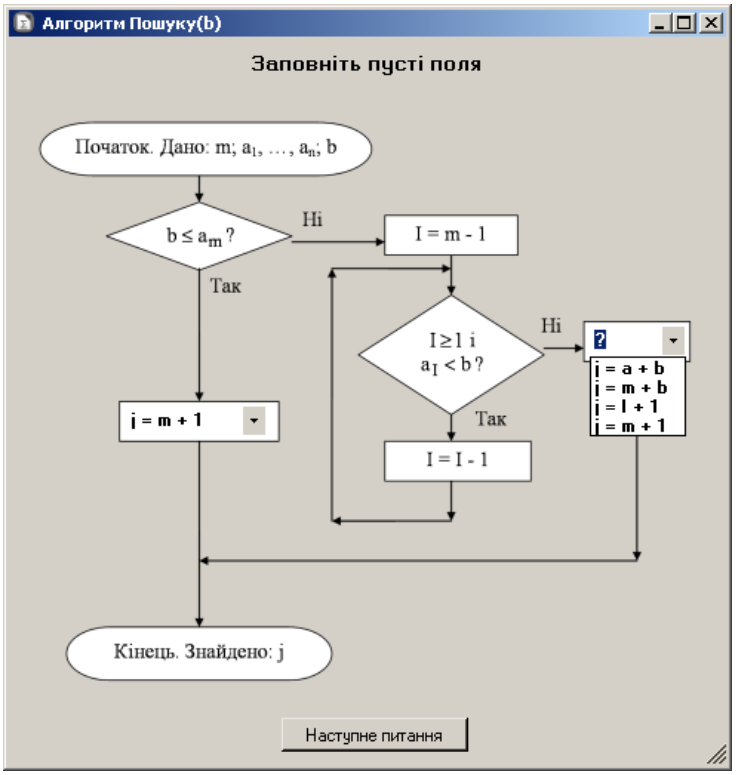

Рисунок 2.6 – Продовження другого кроку

3) Студентом спеціальності «Комп’ютерні науки» ПУЕТ Бибкою Б. М. у 2020 р. був створений тренажер ***«Побудова блок-схем алгоритмів циклічної структури»*** для курсу «Інформатика» (рис. 2.7-2.9). У симуляторі є умова – код мовою Pascal. Для кожного рядку коду слід перетягнути підходящий символ блок-схеми на жовтий прямокутник. Символ є пустим, не заповненим текстом. Далі з’являється побудована з врахуванням на поточного кроку, блок-схема.

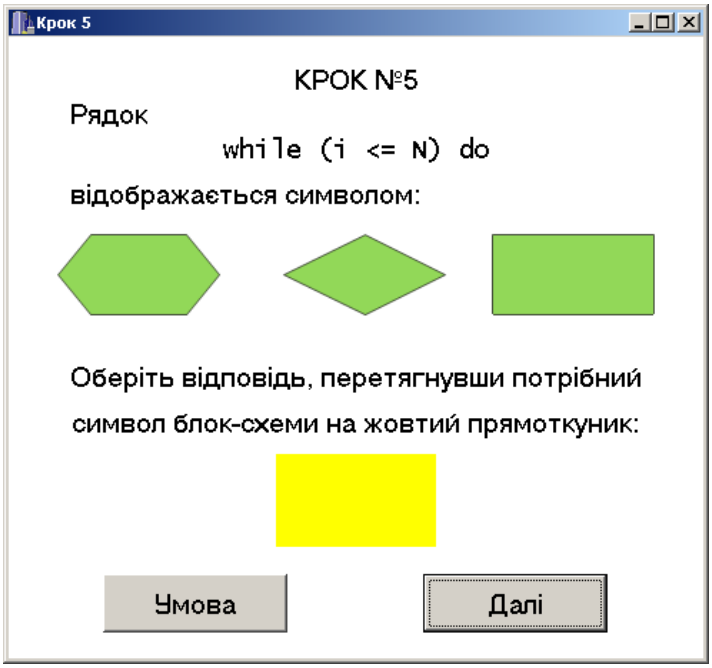

Рисунок 2.7 – Умова п’ятого кроку (тренажер 2)

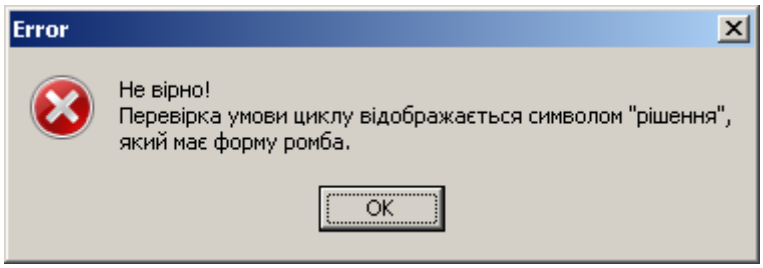

Рисунок 2.8 – Пояснення помилки

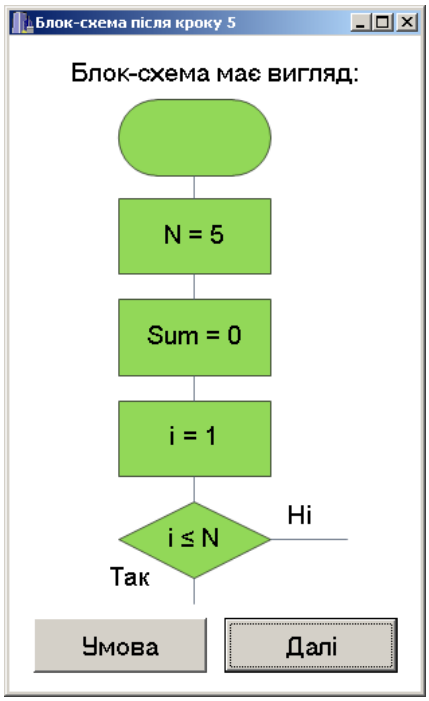

Рисунок 2.9 – Після п’ятого кроку

2.2. Позитивні риси переглянутих програм

Тренажер 1 – вдалим є поділ форми на три складові: умову, питання, блок-схему після виконання кроку. Пояснюються всі помилки.

Тренажер 2 – цікавою є ідея використання блок-схем алгоритмів.

Тренажер 3 – цікавою є ідея перетягування графічних блоків. Вдалою є задумка показувати сконструйовану блок-схему після кожного кроку. Умова прикладу весь час доступна. Пояснюються всі помилки.

2.3. Негативні риси переглянутих програм

Тренажер 1 та 3 істотних недоліків не мають.

Тренажер 2 містить посилання на дистанційний курс, в якому і буде впроваджений тренажер (тобто є певна зацикленість процесу). Питання безсистемні. Для блок-схем немає пояснень алгоритму, позначень. Оскільки, алгоритм однієї і тієї задачі може бути реалізований по-різному, то питання по блок-схемам в тому виді, в якому вони реалізовані в тренажері, не коректні. Не пояснюють толком помилки.

2.4. Необхідність та актуальність теми

Отже, огляд показав, що певні тренажери по побудові або використанню блок-схем алгоритмів вже існують, але такі тренажери поки що розроблені не для кожної теми і не для кожного дистанційного курсу. Крім того, очевидно, що різні курси використовують різні мови програмування. Тому, актуальність теми цією випускової роботи незаперечна.

.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Алгоритм тренажеру для прикладу № 1

Умова. Є алгоритм [3], записаний мовою `Object Pascal`, в якому на екран виводиться значення функції $y = e^{\sin x} \cos x$ на відрізку $[\,0, \pi\,]$ з кроком 0,1. Створити блок-схему.

```
var
  x, y:real;
const
  pi=3.14;
begin
  x:=0;
  repeat
    y:=exp(sin(x))*cos(x);
    writeln('x=', x:4:2,' y=', y:4:2);
    x:=x+0.1;
  until (x>pi);
  readln;
end.
```

Під час проходження тренінгу умова завдання весь час видима користувачу.

Крок № 1. Користувач бачить рисунок 3.1. У першому символі є список опцій для вибору:

1) Кінець;

2) Зупинка;

3) Початок;

4) var.

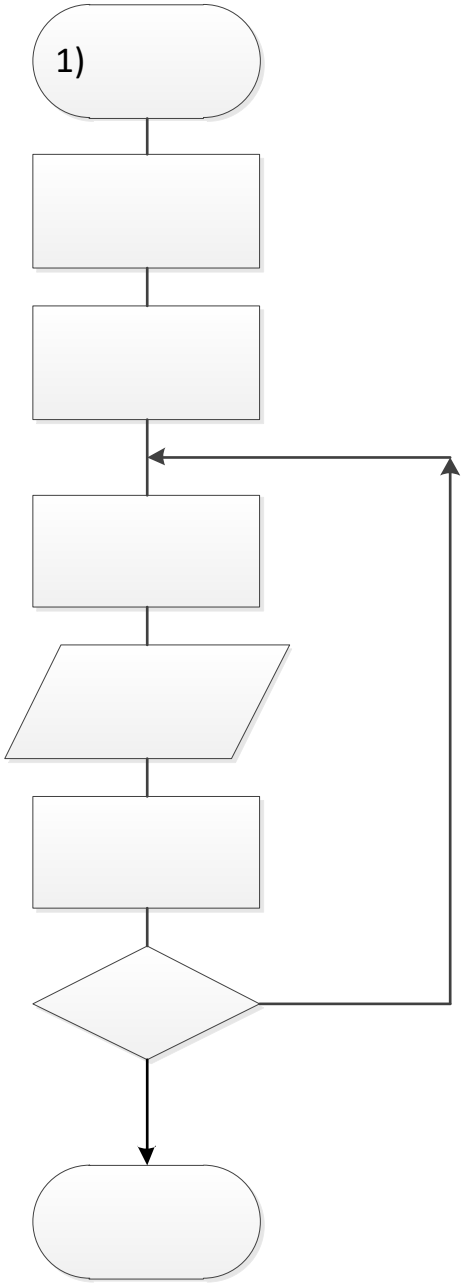

Рисунок 3.1 – Крок 1

Користувач обирає один з варіантів.

Якщо обрано варіант 3, то з’являється підтверджуюче повідомлення. Відбувається перехід на наступний крок.

В іншому випадку з’являється повідомлення: «Даний блок символізує старт алгоритму. Тому вірний варіант – це «Початок». Користувач повинен виправити помилку.

Крок № 2. Користувач бачить рисунок 3.2. У другому символі є список:

1) var;

2) var x, y:real;

3) const;

4) pi=3.14.

Користувач обирає один з варіантів.

Якщо обрано варіант 4, то з’являється підтверджуюче повідомлення. Відбувається перехід на наступний крок.

В іншому випадку з’являється повідомлення: «Даний блок символізує маніпуляцію з даними. Тому вірний варіант – це «pi=3.14». Користувач повинен виправити помилку.

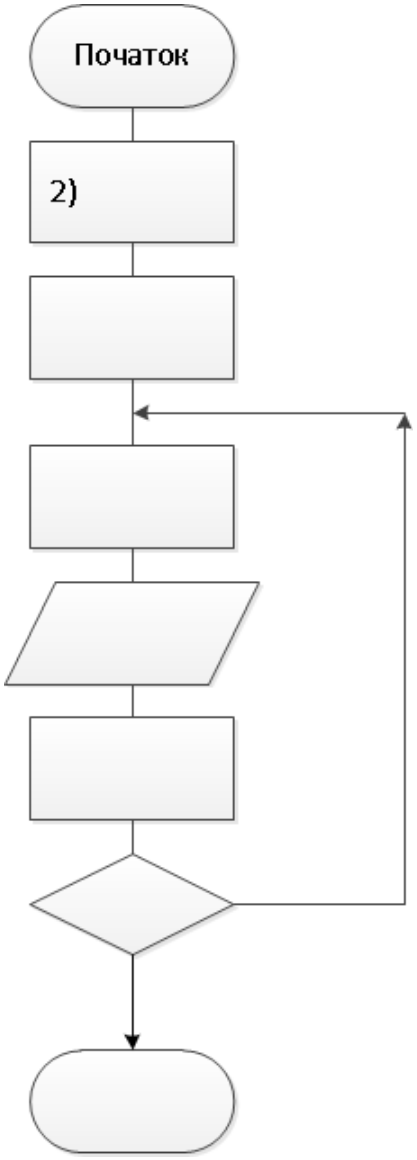

Рисунок 2.2 – Крок 2

Крок № 3. Користувач бачить рисунок 3.3. У третьому символі є список:

1) x=0;

2) var x, y:real;

3) repeat;

4) var.

Користувач обирає один з варіантів.

Якщо обрано варіант 1, то з’являється підтверджуюче повідомлення. Відбувається перехід на наступний крок.

В іншому випадку з’являється повідомлення: «Даний блок символізує маніпуляцію з даними. Тому вірний варіант – це «x=0». Користувач повинен виправити помилку.

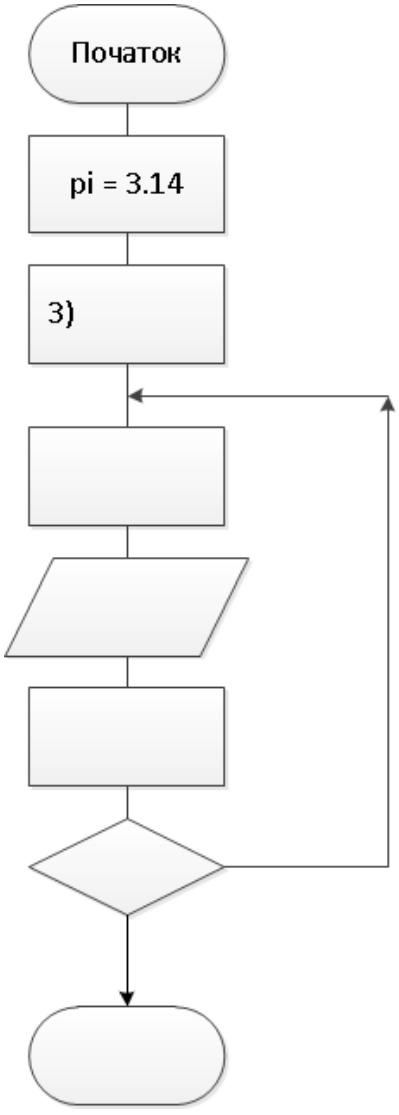

Рисунок 3.3 – Крок 3

Крок № 4. Користувач бачить рисунок 3.4. У четвертому символі є список:

1) repeat;

2) $y = e^{\sin x} \cos x$***;***

3) writeln('x=', x:4:2,' y=', y:4:2);

4) x>pi.

Користувач обирає один з варіантів.

Якщо обрано варіант 2, то з'являється підтверджуюче повідомлення. Відбувається перехід на наступний крок.

В іншому випадку з'являється повідомлення: «Даний блок символізує маніпуляцію з даними. Тому вірний варіант – це «$y = e^{\sin x} \cos x$». Користувач повинен виправити помилку.

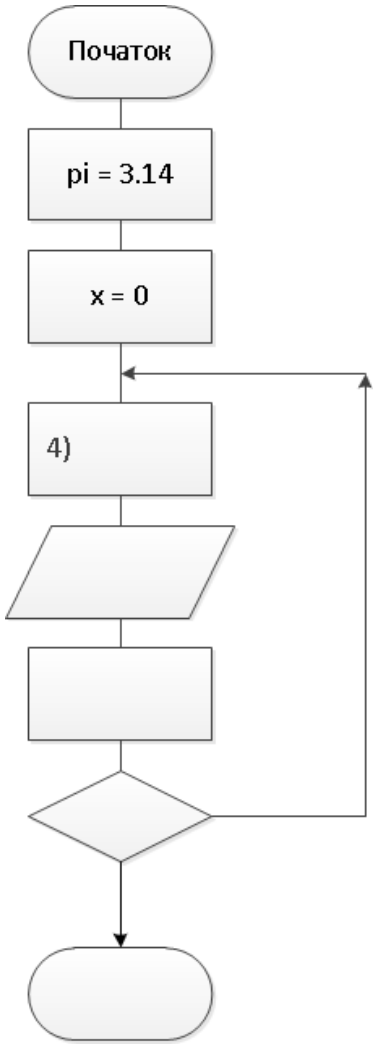

Рисунок 3.4 – Крок 4

Крок № 5. Користувач бачить рисунок 3.5. У п’ятому символі є список:

1) Вивід x та y;

2) repeat;

3) x:=x+0.1;

4) x>pi.

Користувач обирає один з варіантів.

Якщо обрано варіант 1, то з’являється підтверджуюче повідомлення. Відбувається перехід на наступний крок.

В іншому випадку з’являється повідомлення: «Даний блок символізує вивід даних. Тому вірний варіант – це «Вивід x та y». Користувач повинен виправити помилку.

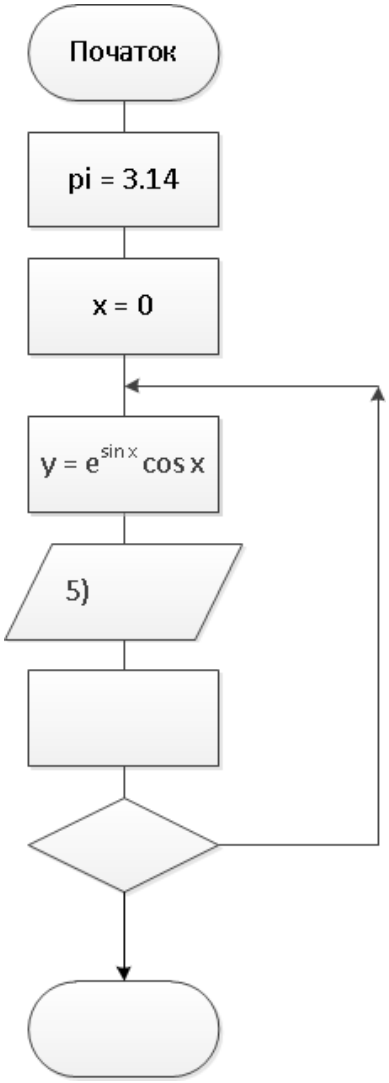

Рисунок 3.5 – Крок 5

Крок № 6. Користувач бачить рисунок 3.6. У шостому символі є список:

1) Вивід x та y;

2) x>pi;

3) x=x+0.1;

4) until (x>pi);

Користувач обирає один з варіантів.

Якщо обрано варіант 3, то з’являється підтверджуюче повідомлення. Відбувається перехід на наступний крок.

В іншому випадку з’являється повідомлення: «Даний блок символізує маніпуляцію з даними. Тому вірний варіант – це «x=x+0.1». Користувач повинен виправити помилку.

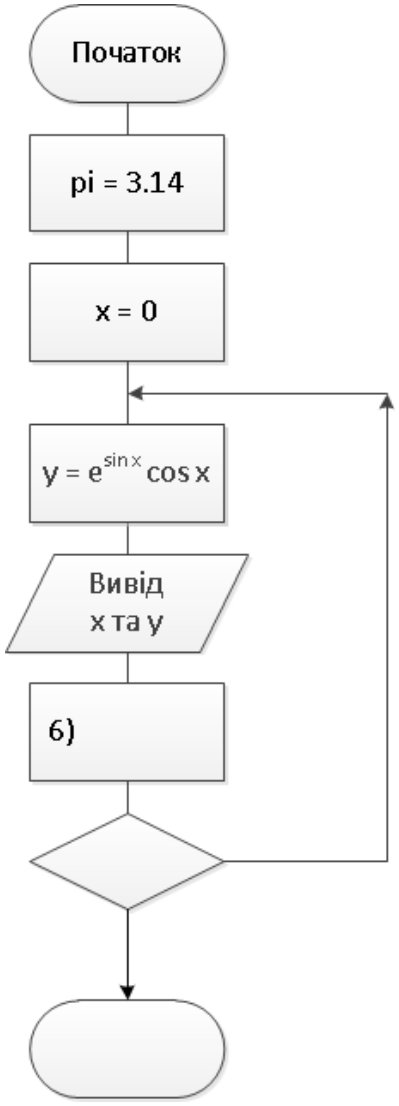

Рисунок 3.6 – Крок 6

Крок № 7. Користувач бачить рисунок 3.7. У сьомому символі є список:

1) repeat;

2) readln;

3) until;

4) x>pi?

Користувач обирає один з варіантів.

Якщо обрано варіант 4, то з’являється підтверджуюче повідомлення. Відбувається перехід на наступний крок.

В іншому випадку з’являється повідомлення: «Даний блок символізує перевірку умови. Тому вірний варіант – це «x>pi?». Користувач повинен виправити помилку.

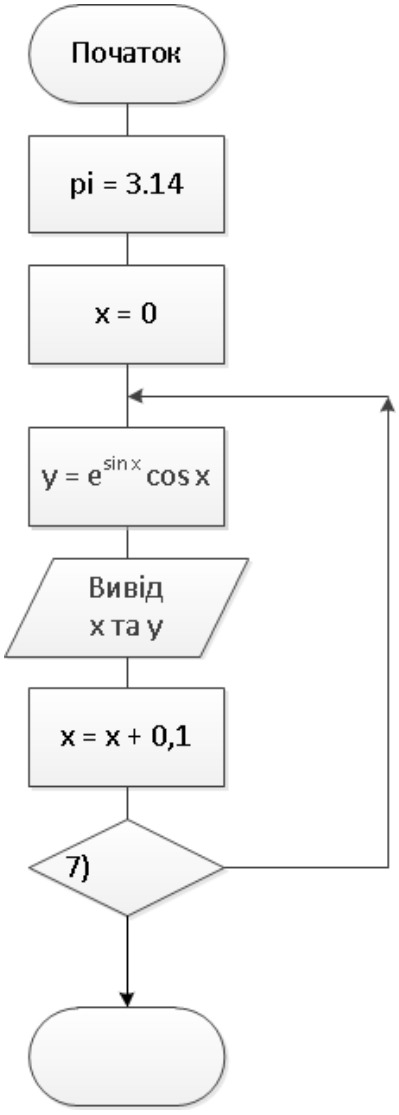

Рисунок 3.7 – Крок 7

Крок № 8. Користувач бачить рисунок 3.8. На лінії (біля цифри 8) є список:

1) так;

2) ні.

Користувач обирає один із запропонованих варіантів.

Якщо обрано варіант 2, то з’являється підтверджуюче повідомлення. Відбувається перехід на наступний крок.

В іншому випадку з’являється повідомлення: «Цикл repeat … until виконується тоді, коли умова циклу не виконується. Отже, вірний варіант – «ні». Користувач повинен виправити помилку.

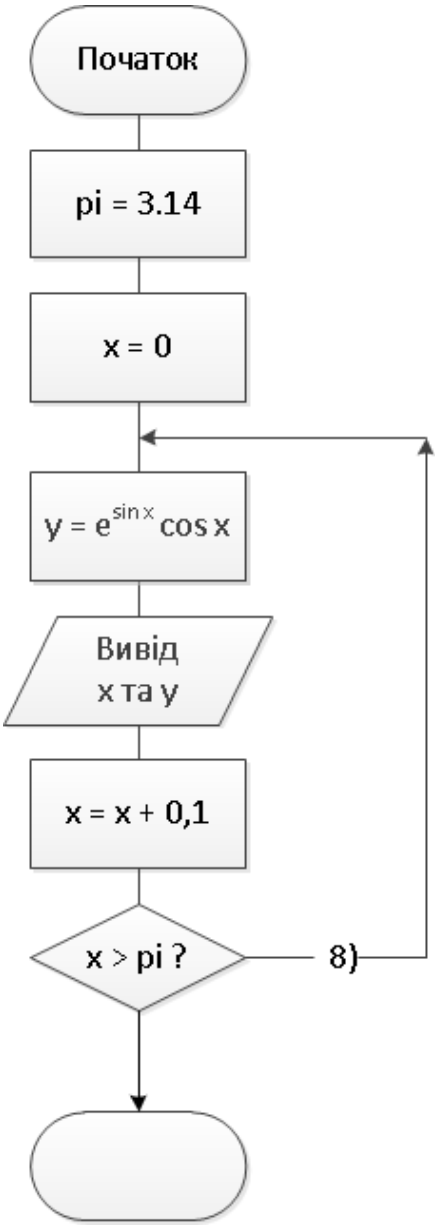

Рисунок 3.8 – Крок 8

Крок № 9. Користувач бачить рисунок 3.9. На лінії (біля цифри 9) є список:

1) так;

2) ні.

Користувач обирає один із запропонованих варіантів.

Якщо обрано варіант 1, то з’являється підтверджуюче повідомлення. Відбувається перехід на наступний крок.

В іншому випадку з’являється повідомлення: «Цикл repeat … until завершує роботу тоді, коли умова циклу виконується. Отже, вірний варіант – «так». Користувач повинен виправити помилку.

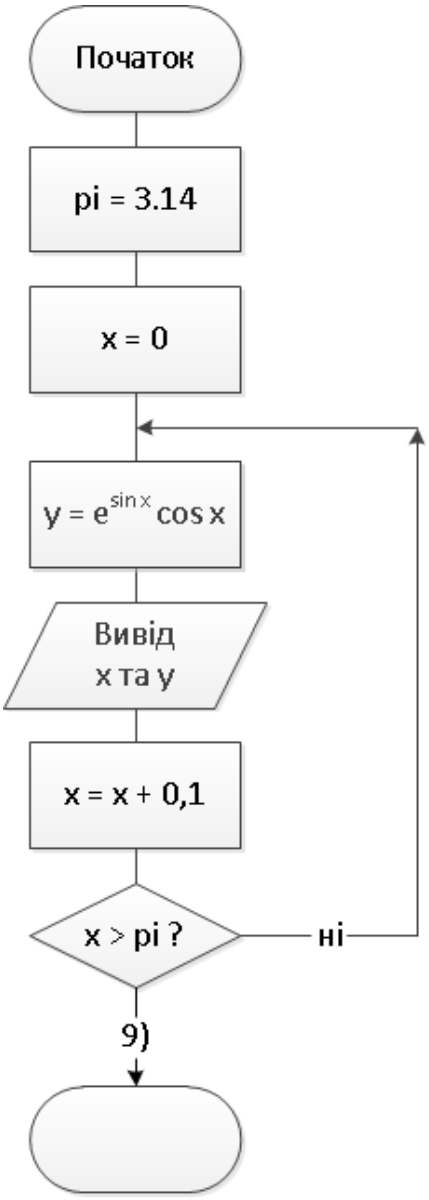

Рисунок 3.9 – Крок 9

Крок № 10. Користувач бачить рисунок 3.10. У символі (біля цифри 10) список:

1) Початок;

2) Кінець;

3) readln;

4) repeat.

Користувач обирає один із запропонованих варіантів.

Якщо обрано варіант 2, то з’являється підтверджуюче повідомлення. Відбувається перехід на наступний крок.

В іншому випадку з’являється повідомлення: «Даний блок символізує завершення алгоритму. Тому вірний варіант – це «Кінець». Користувач повинен виправити помилку.

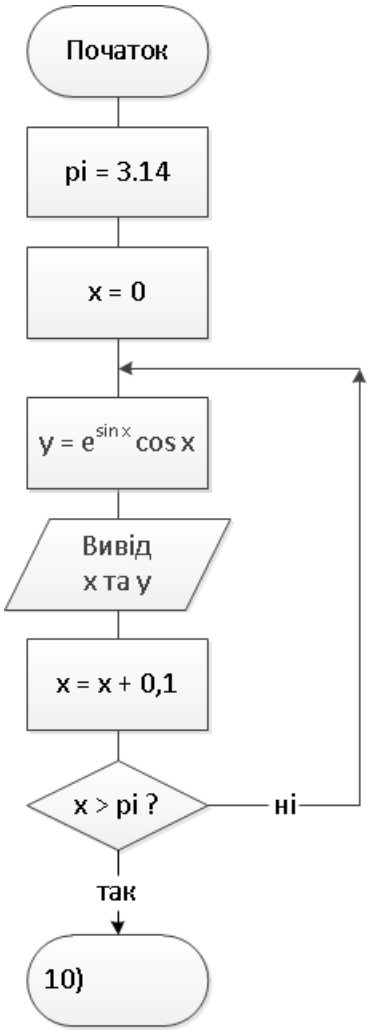

Рисунок 3.10 – Крок 10

Крок № 11. Таким чином, блок-схема алгоритму приймає вигляд (рис. 3.11):

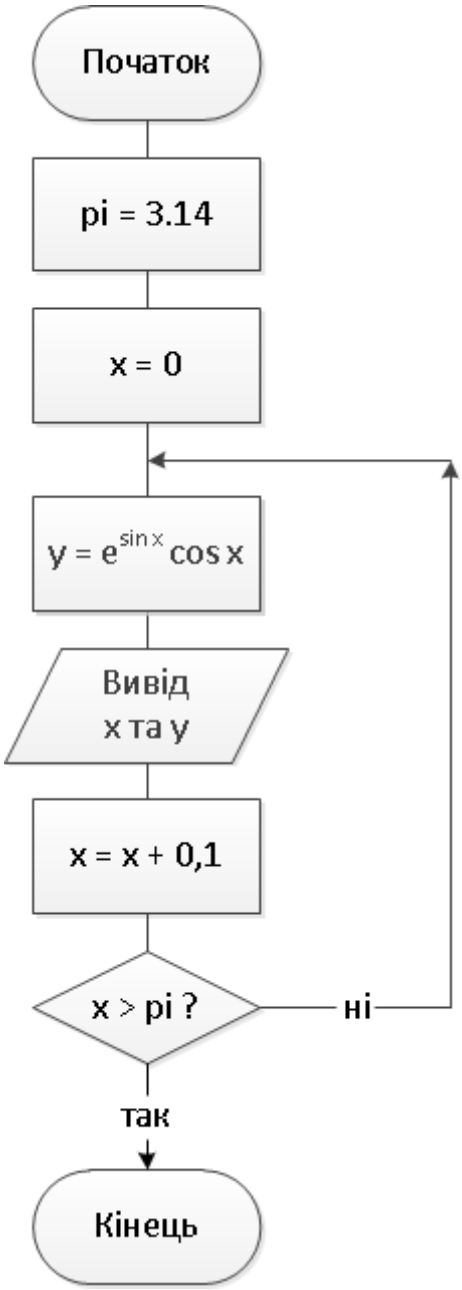

Рисунок 3.11 – Крок 11

3.2. Алгоритм тренажеру для прикладу № 2

Алгоритм другого прикладу винесено у додаток А.

3.3. Блок-схема алгоритму

На рис. 3.12 показана загальна схема роботи алгоритмі симулятору. Конкретизована блок-схема розташована у додатку В (рис. В.1-В.3).

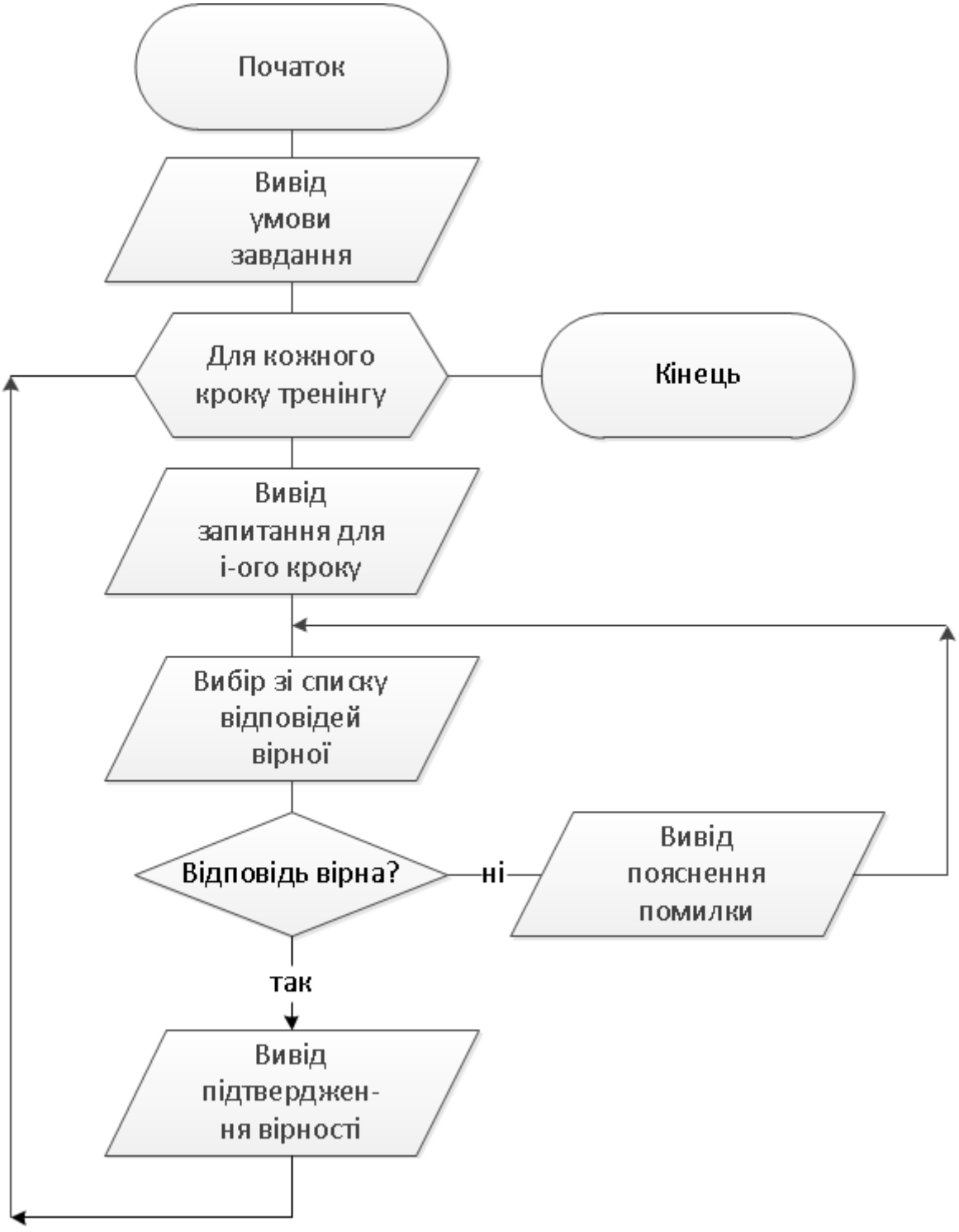

Рисунок 3.12 – Блок-схема

4. ПРАКТИЧНА ЧАСТИНА

4.1. Опис програмної реалізації тренажеру

Алгоритми обох прикладів були програмно реалізовані мовою C++ у середовищі Borland Builder (рис. 4.1-4.29).

Розглянемо будову типового кроку алгоритму. Візьмемо для визначеності крок № 1 (рис. 4.4-4.7) прикладу № 1.

На цьому і на інших кроках користувачу слід обрати вірну відповідь із запропонованих у списку, що розгортається (див., наприклад, рис. 4.5).

Було взято компонент TComboBox. За допомогою властивості Items було створено список варіантів відповіді (рис. 4.1).

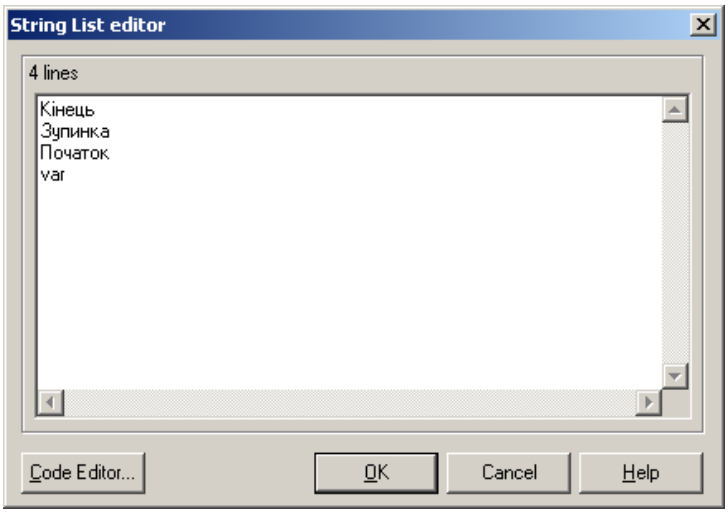

Рисунок 4.1 – Створення списку відповідей

Із використанням властивості Text було введено декілька знаків питання (рис. 4.1).

Код, написаний для кнопки наступний (рис. 4.2):

```
void __fastcall TForm2::BitBtn1Click(TObject *Sender)
{
  if (ComboBox1->ItemIndex==2)
  {
    MessageDlg(Positive_Answer(),       mtInformation,
TMsgDlgButtons() << mbOK, 0);
    Form3->Show();
    Form2->Hide();
  }
  else
  {
     MessageDlg("Даний    блок    символізує    старт
алгоритму.\nТому вірний варіант - це \"Початок\".",
mtError, TMsgDlgButtons() << mbOK, 0);
  }
}
```

Рисунок 4.2 – Код програми

Проаналізуємо його. Якщо обрана вірна відповідь

```
if (ComboBox1->ItemIndex==2)
```

(тут це третя фраза у списку, але, оскільки, нумерація починається з нуля, то номер цієї відповіді є другим), то з’являється підтверджуюче повідомлення:

```
MessageDlg     (Positive_Answer(),     mtInformation,
TMsgDlgButtons() << mbOK, 0);
```

Підтверджуюче повідомлення – це одна з чотирьох фраз: «Гарна робота!», «Блискуче!», «Прекрасно!», «Правильно!». Фраза обирається випадковим чином і повертається функцією `Positive_Answer()`.

Після цього відбувається закриття поточного кроку

```
Form2->Hide();
```

та відкриття наступного кроку:

```
Form3->Show();
```

Якщо обрана відповідь не є вірною, то з’являється повідомлення з поясненням помилки:

```
MessageDlg("Даний блок символізує старт алгоритму.\nТому вірний варіант - це \"Початок\".", mtError, TMsgDlgButtons() << mbOK, 0);
```

Код декількох кроків та функцією `Positive_Answer` є у додатку В.

4.2. Інструкція по роботі з програмою

Робота симулятору показана на рисунках 4.3-4.29, Б.1-Б.33.

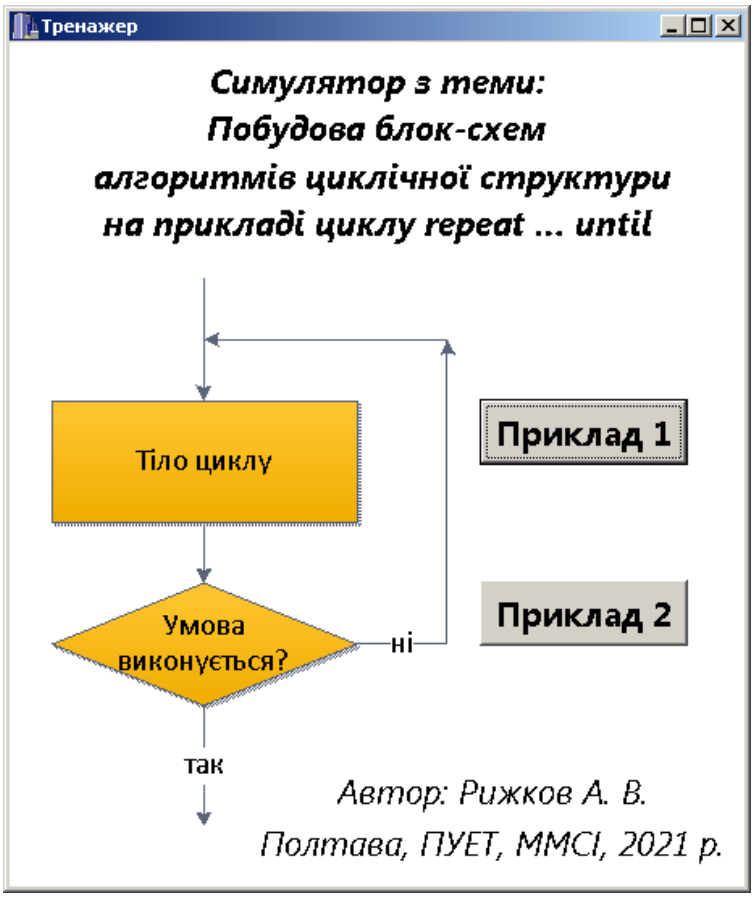

Рисунок 4.3 – Симулятор

На кожному кроці пропонується обрати вірну відповідь зі списку, що розгортається (рис. 4.4-4.5).

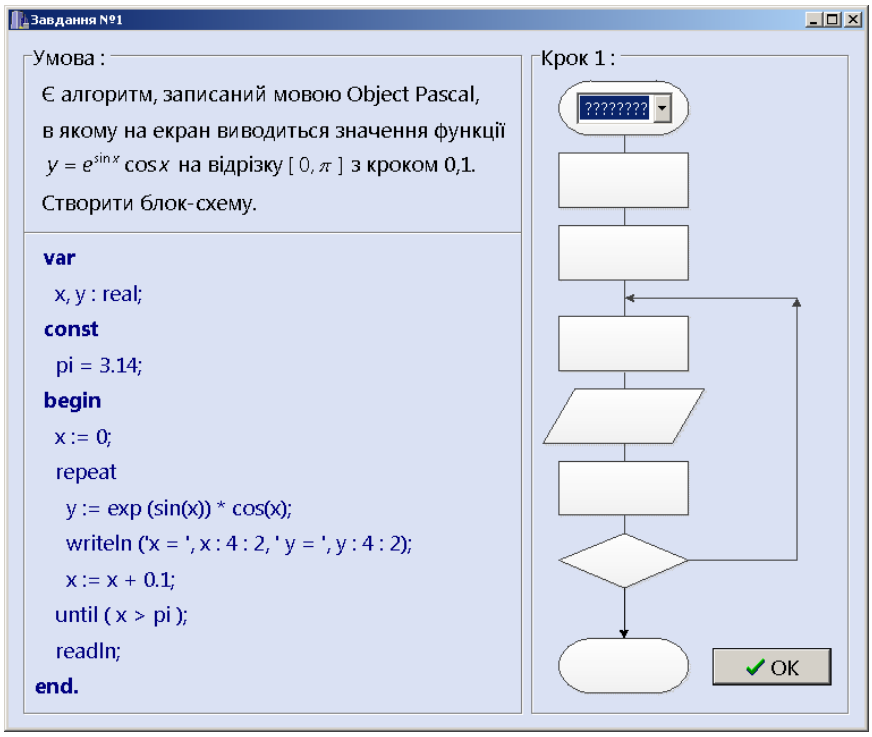

Рисунок 4.4 – Крок 1

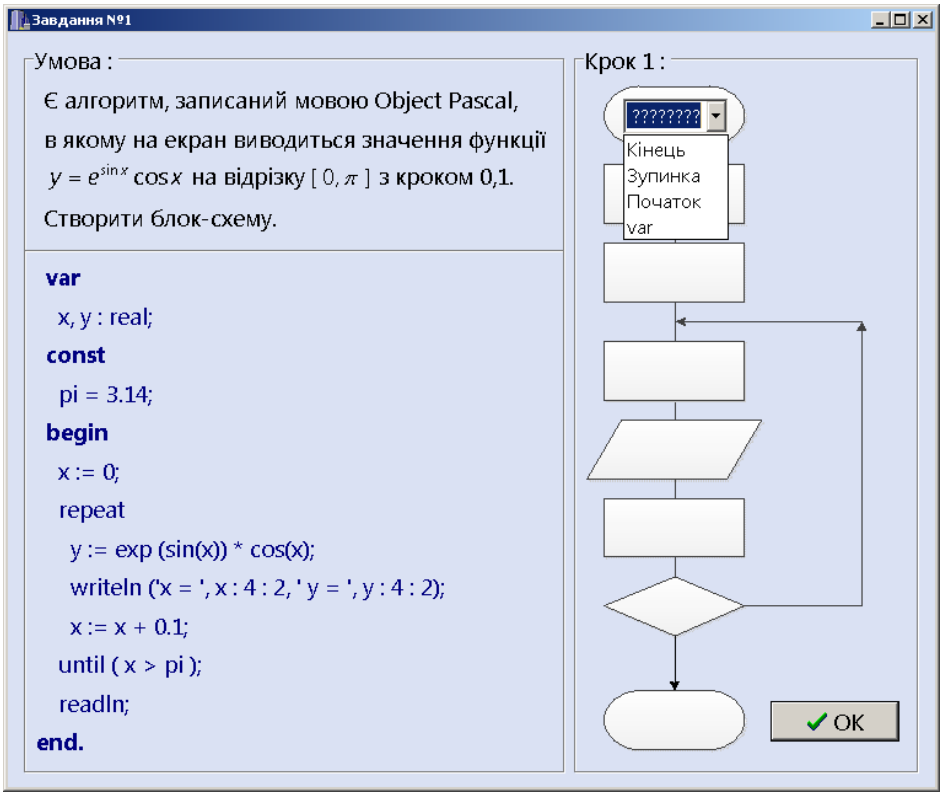

Рисунок 4.5 – Крок 1

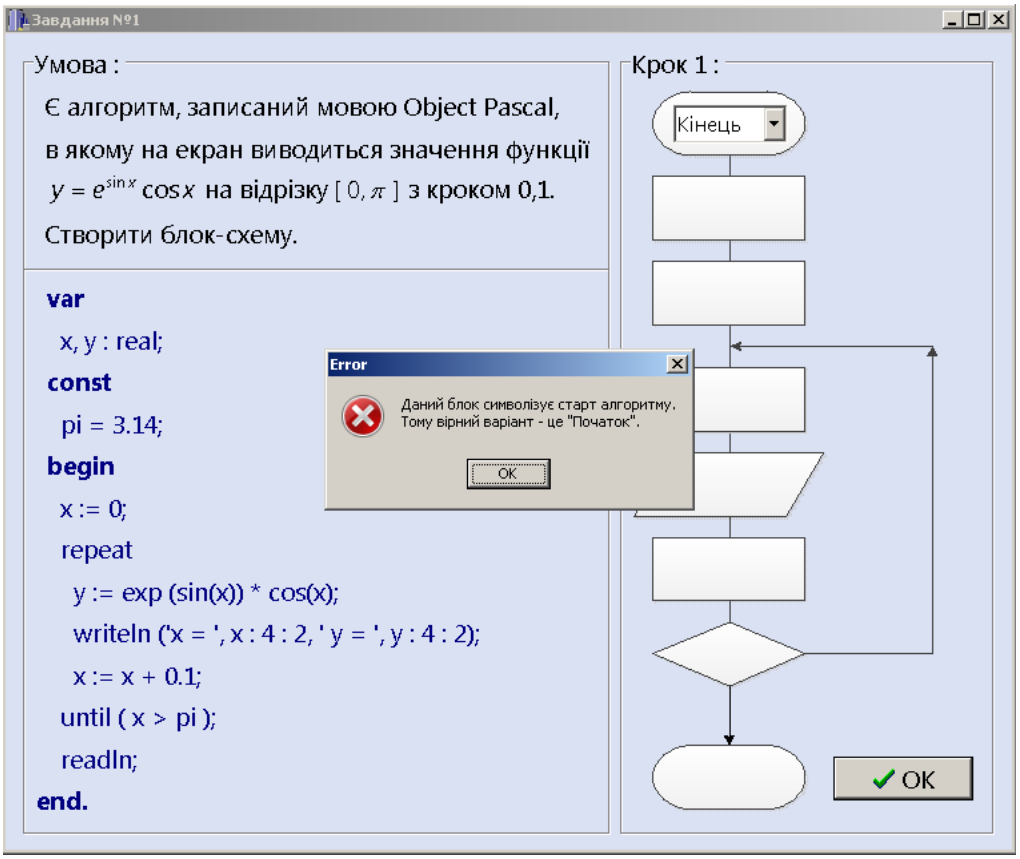

Рисунок 4.6 – Пояснення помилки на кроці 1

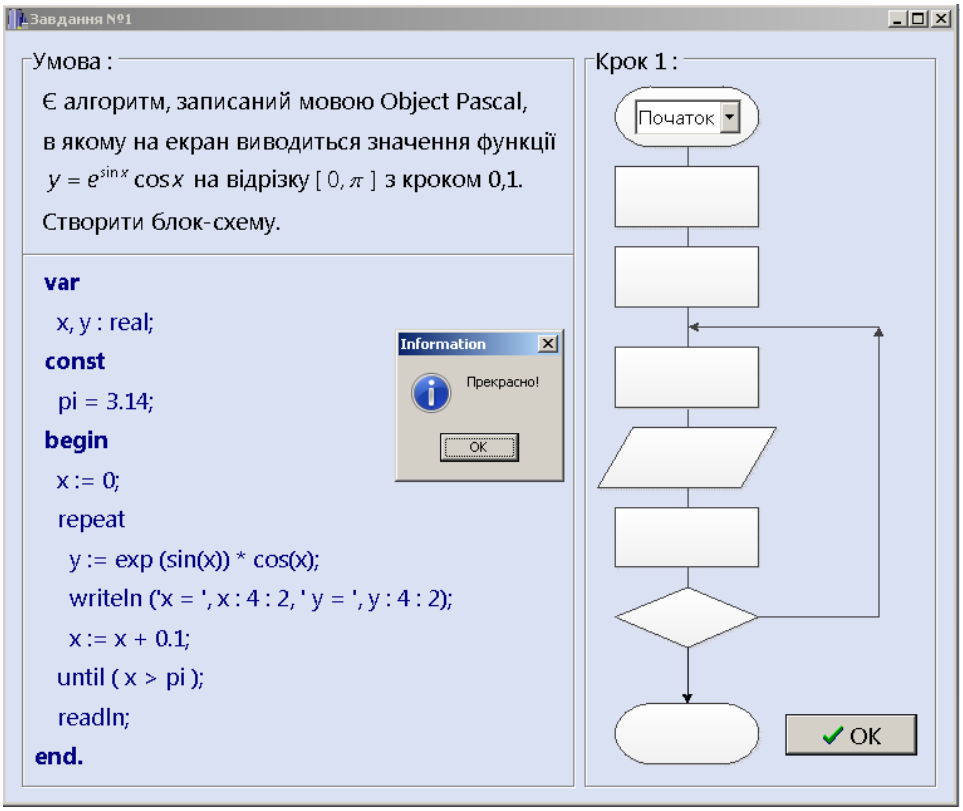

Рисунок 4.7 – Повідомлення, що підтверджує вірність відповіді (одне з чотирьох можливих)

При вірності (рис. 4.7) у випадковому порядку з’являється одне з чотирьох можливих підтверджуючих повідомлень (рис. 4.7-4.10). Відбувається перехід на наступний крок.

При помилці з’являється її пояснення (рис. 4.6). Від користувача очікується виправлення помилки.

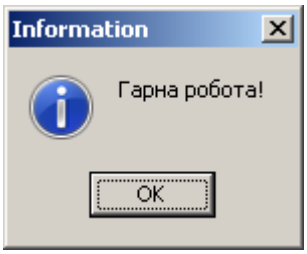

Рисунок 4.8 – Повідомлення, що підтверджує вірність відповіді (одне з чотирьох можливих)

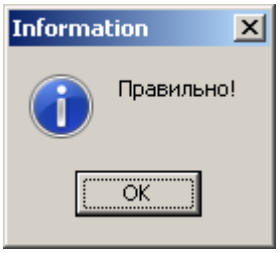

Рисунок 4.9 – Повідомлення, що підтверджує вірність відповіді (одне з чотирьох можливих)

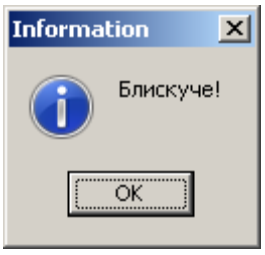

Рисунок 4.10 – Повідомлення, що підтверджує вірність відповіді (одне з чотирьох можливих)

Останній крок відображає побудовану блок-схему (рис. 4.29).

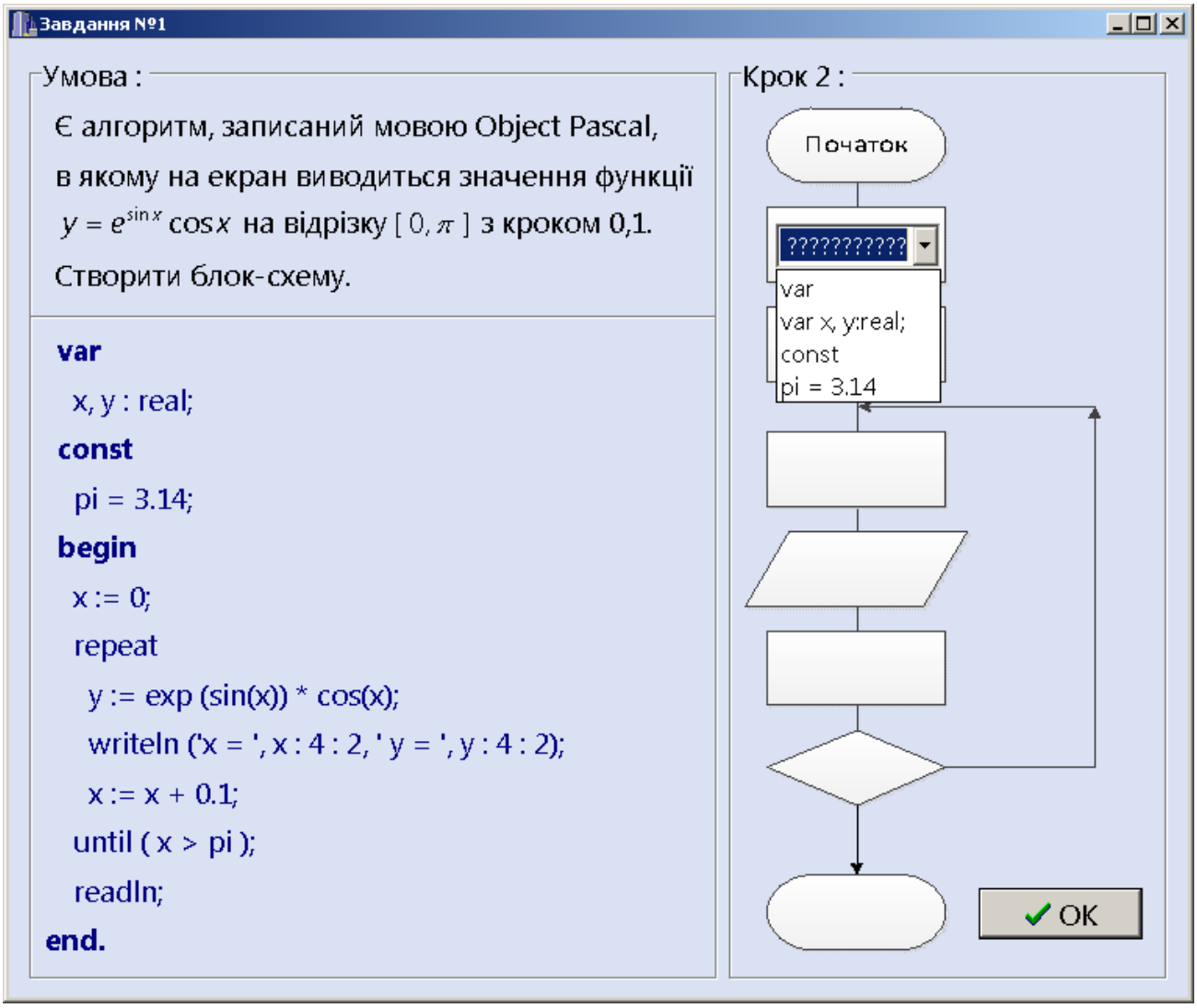

Рисунок 4.11 – Крок 2

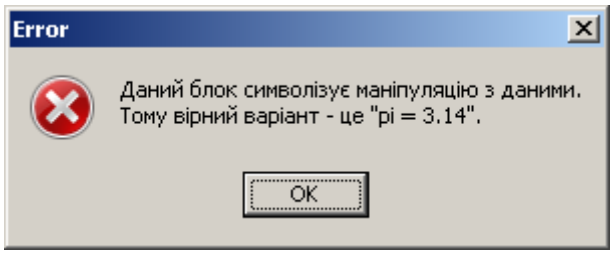

Рисунок 4.12 – Пояснення помилки на кроці 2

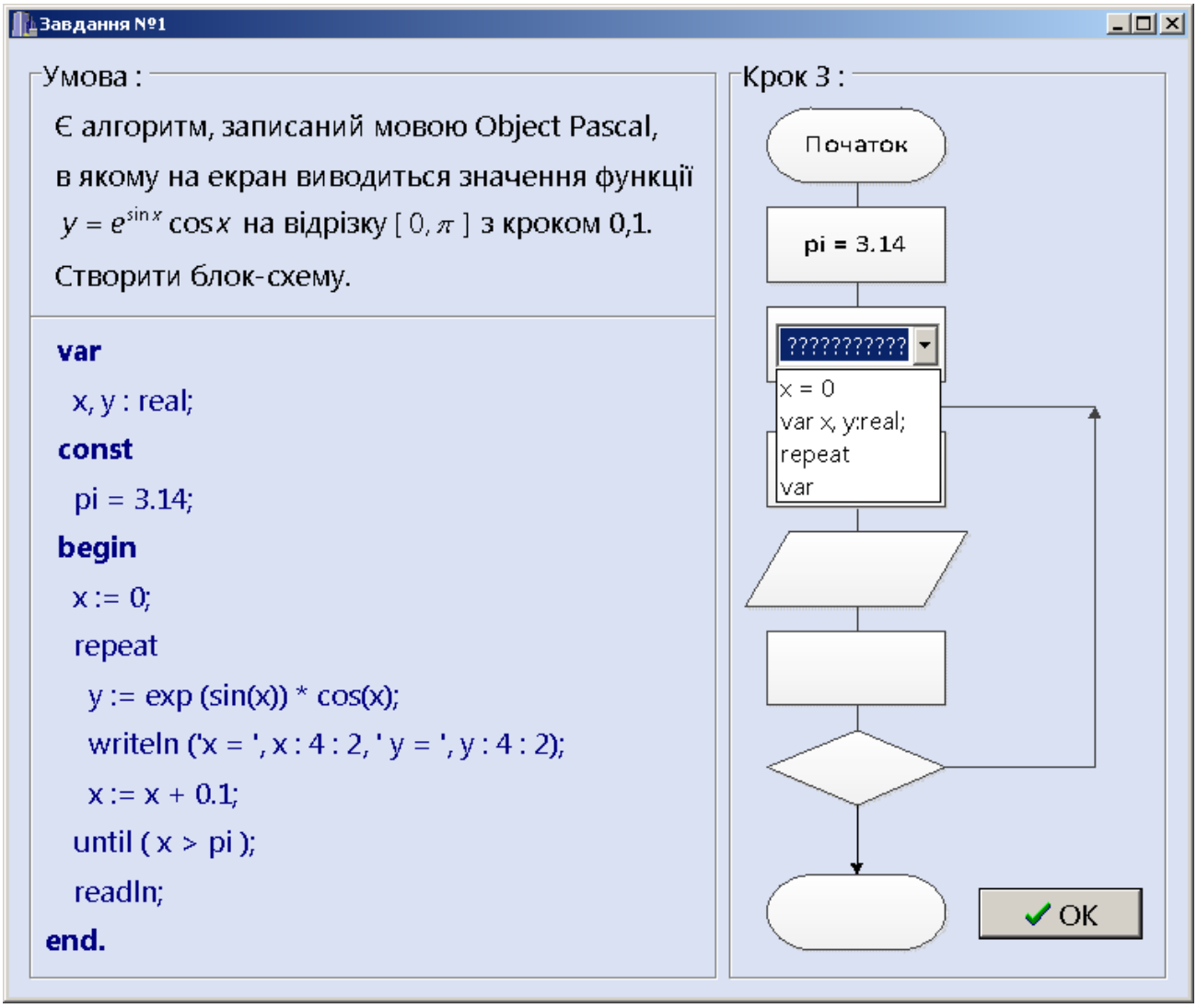

Рисунок 4.13 – Крок 3

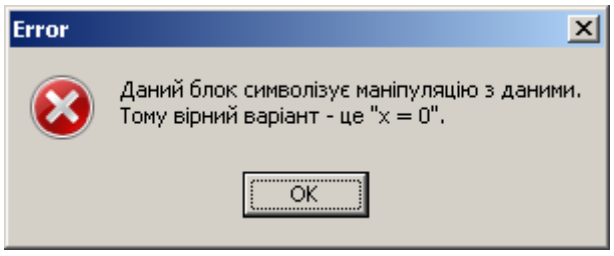

Рисунок 4.14 – Пояснення помилки на кроці 3

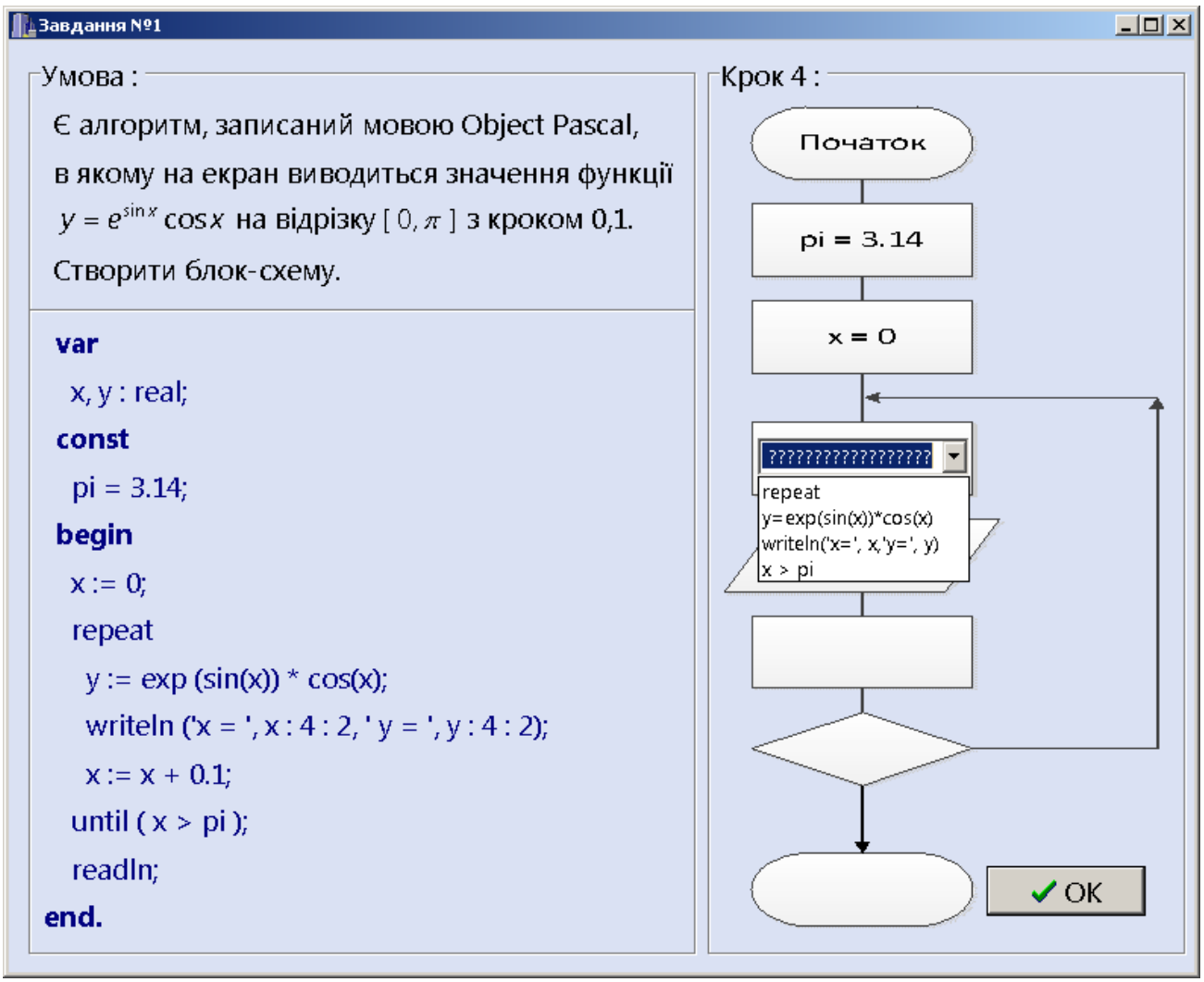

Рисунок 4.15 – Крок 4

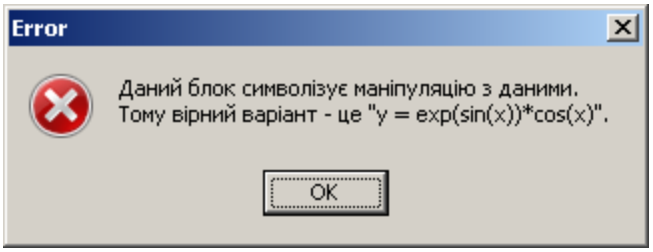

Рисунок 4.16 – Пояснення помилки на кроці 4

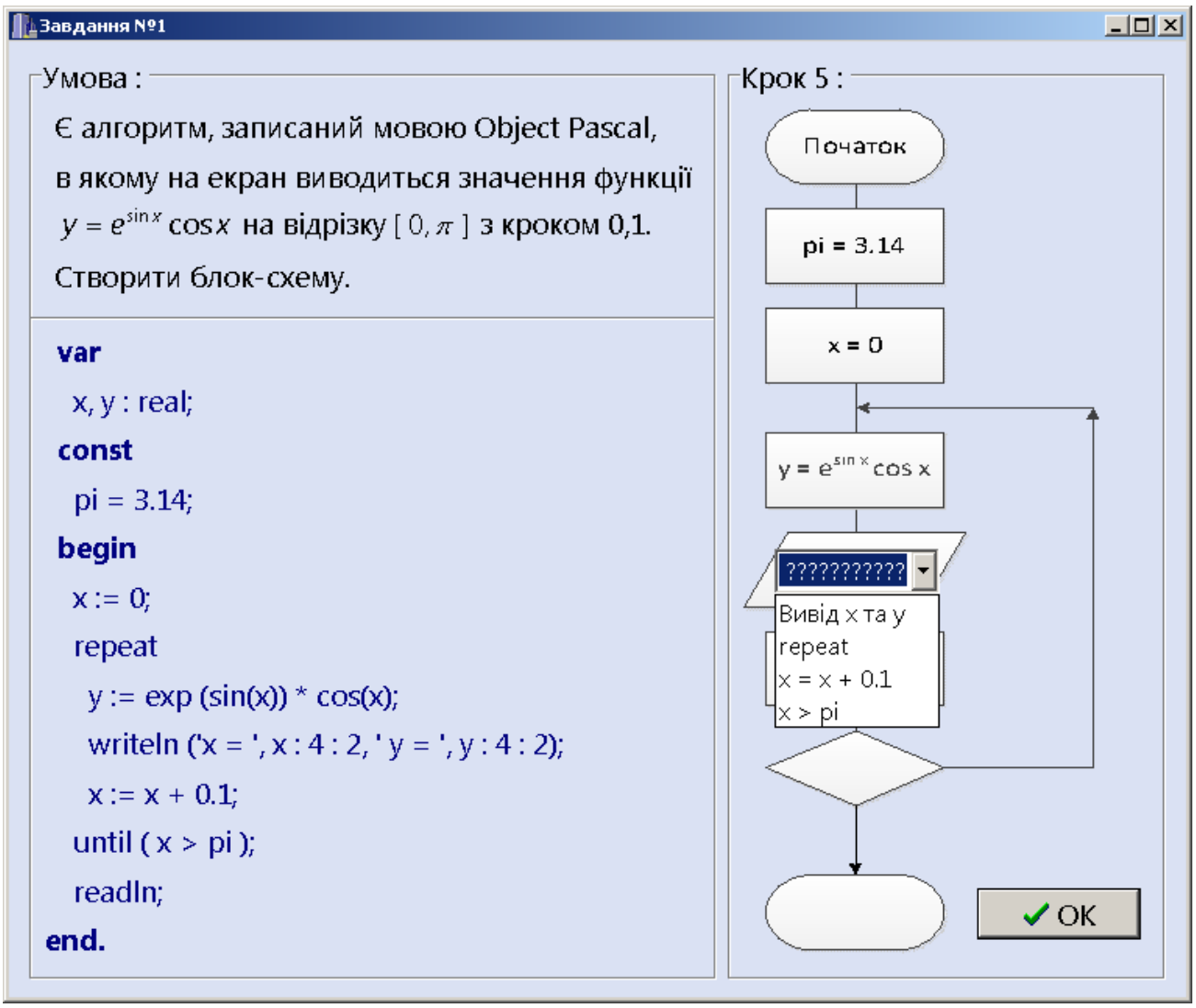

Рисунок 4.17 – Крок 5

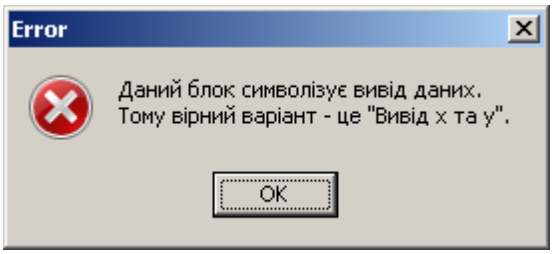

Рисунок 4.18 – Пояснення помилки на кроці 5

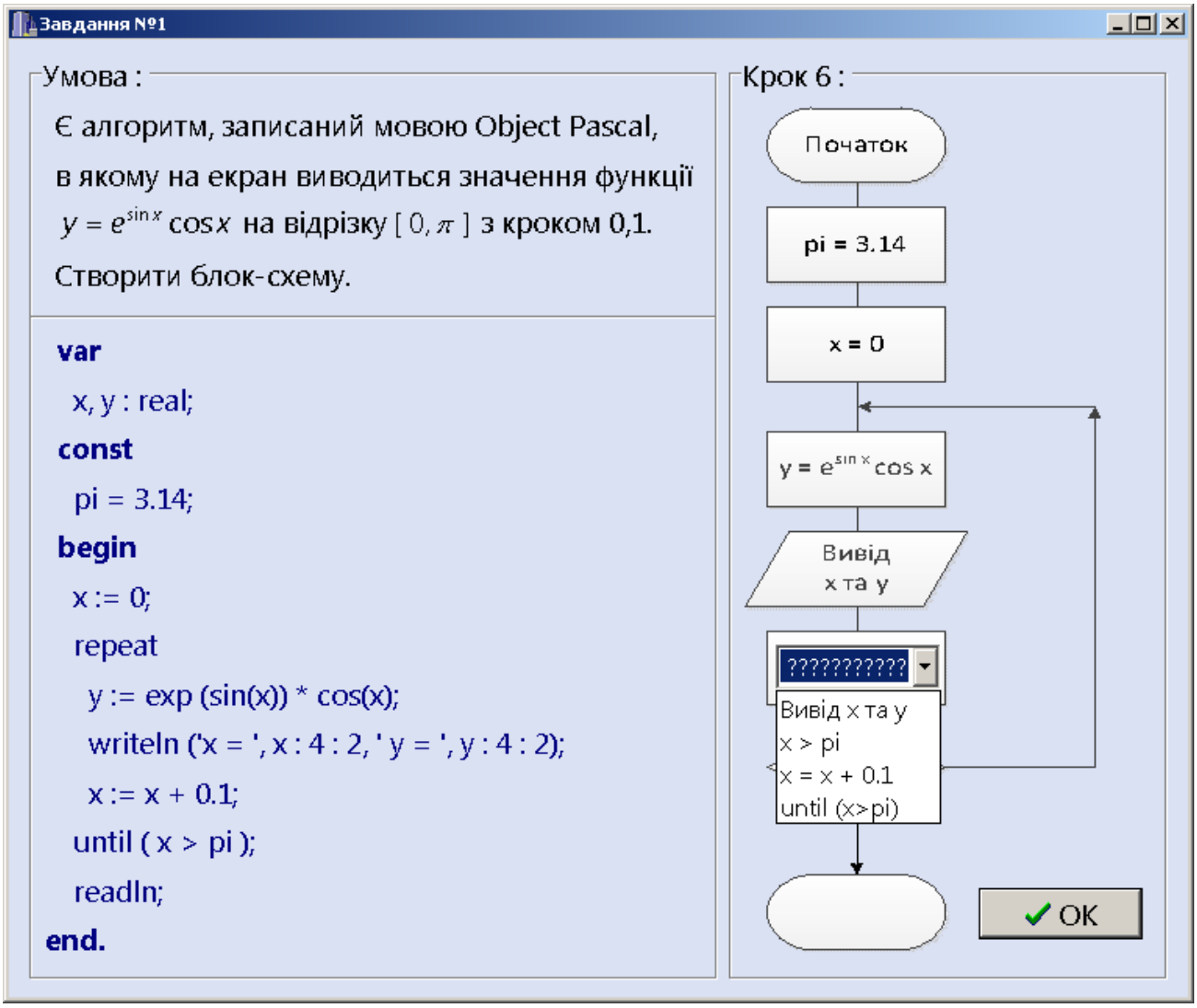

Рисунок 4.19 – Крок 6

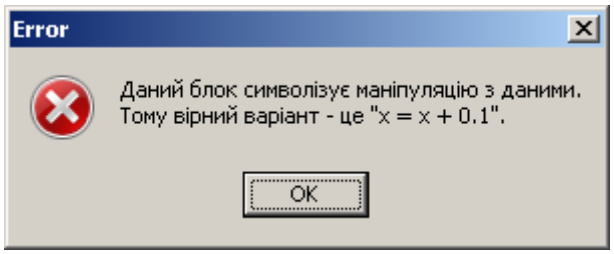

Рисунок 4.20 – Пояснення помилки на кроці 6

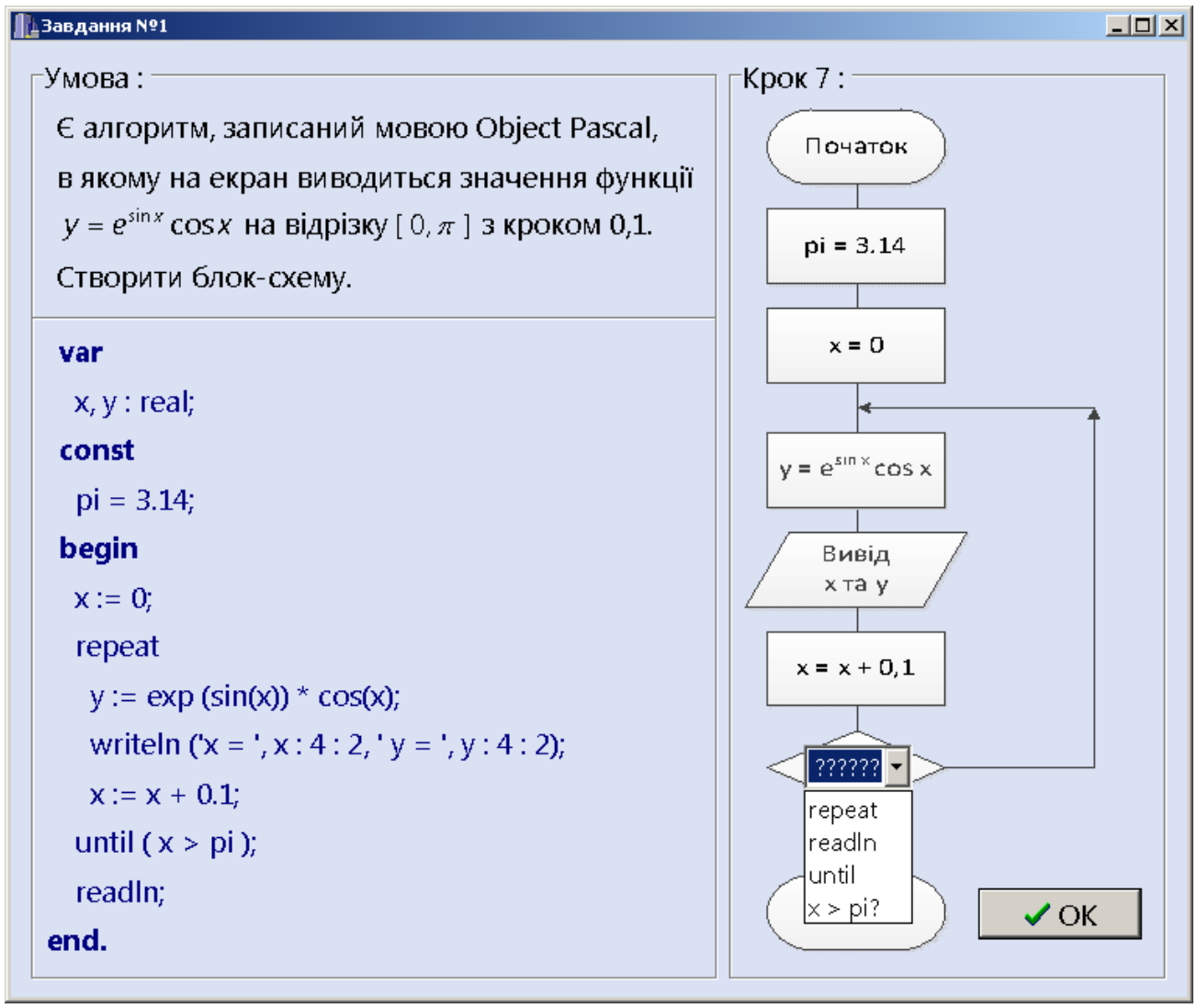

Рисунок 4.21 – Крок 7

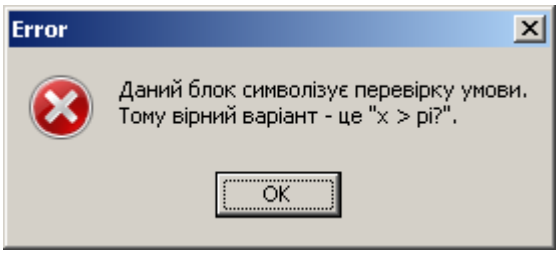

Рисунок 4.22 – Пояснення помилки на кроці 7

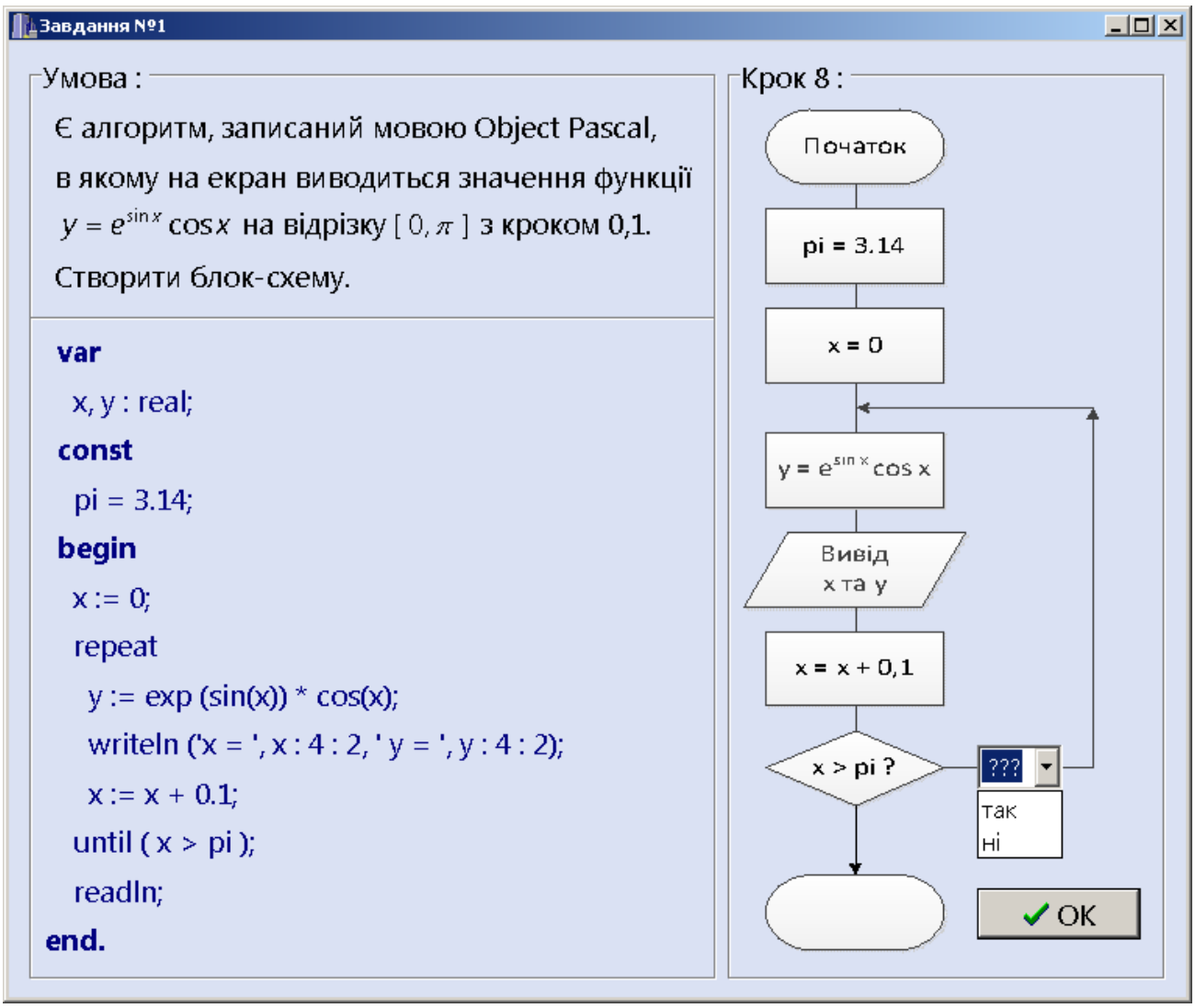

Рисунок 4.23 – Крок 8

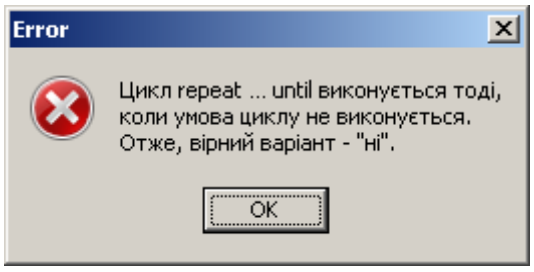

Рисунок 4.24 – Пояснення помилки на кроці 8

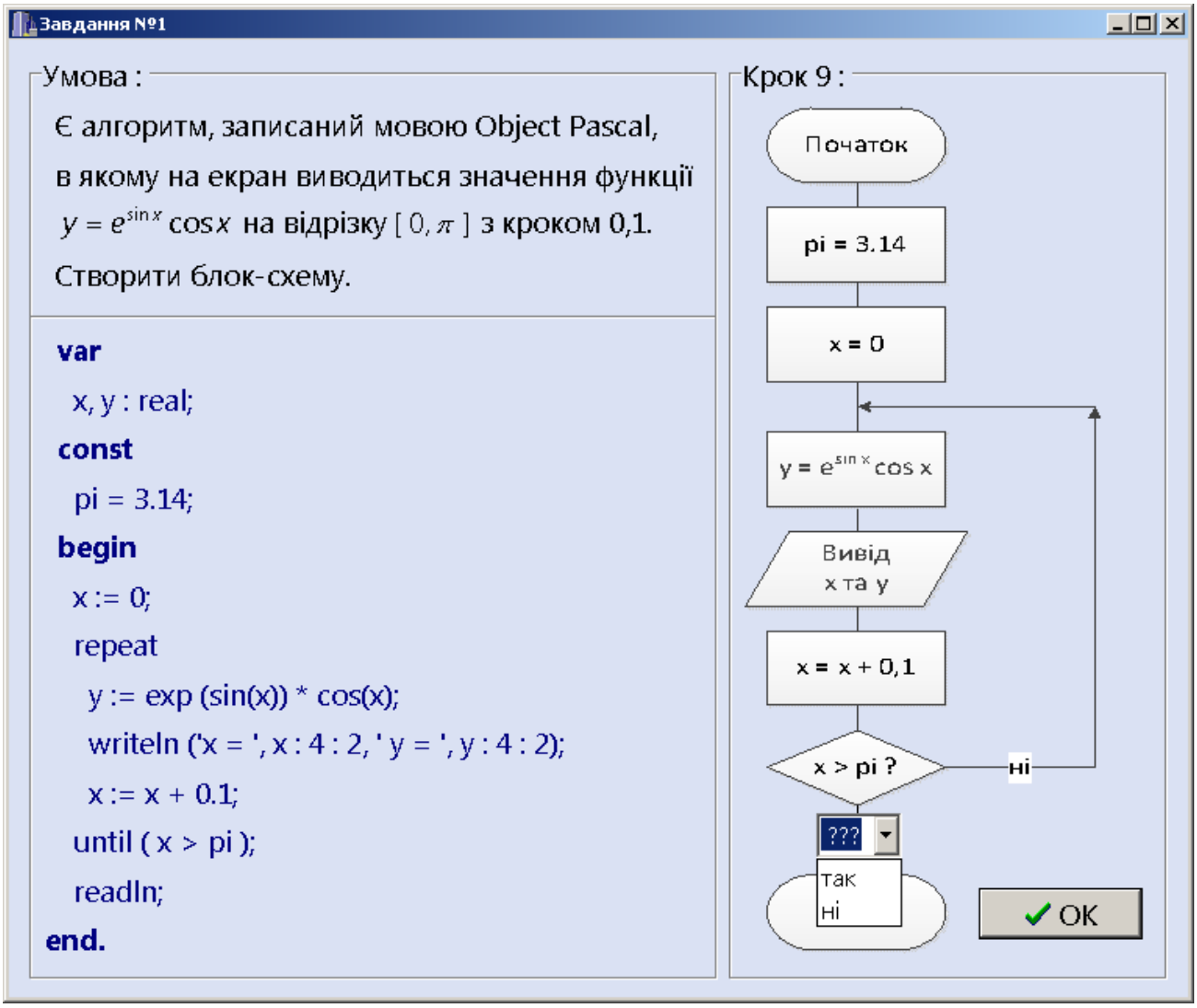

Рисунок 4.25 – Крок 9

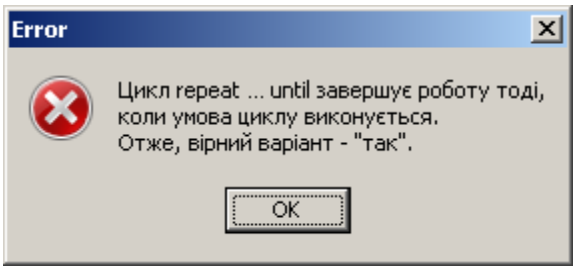

Рисунок 4.26 – Пояснення помилки на кроці 9

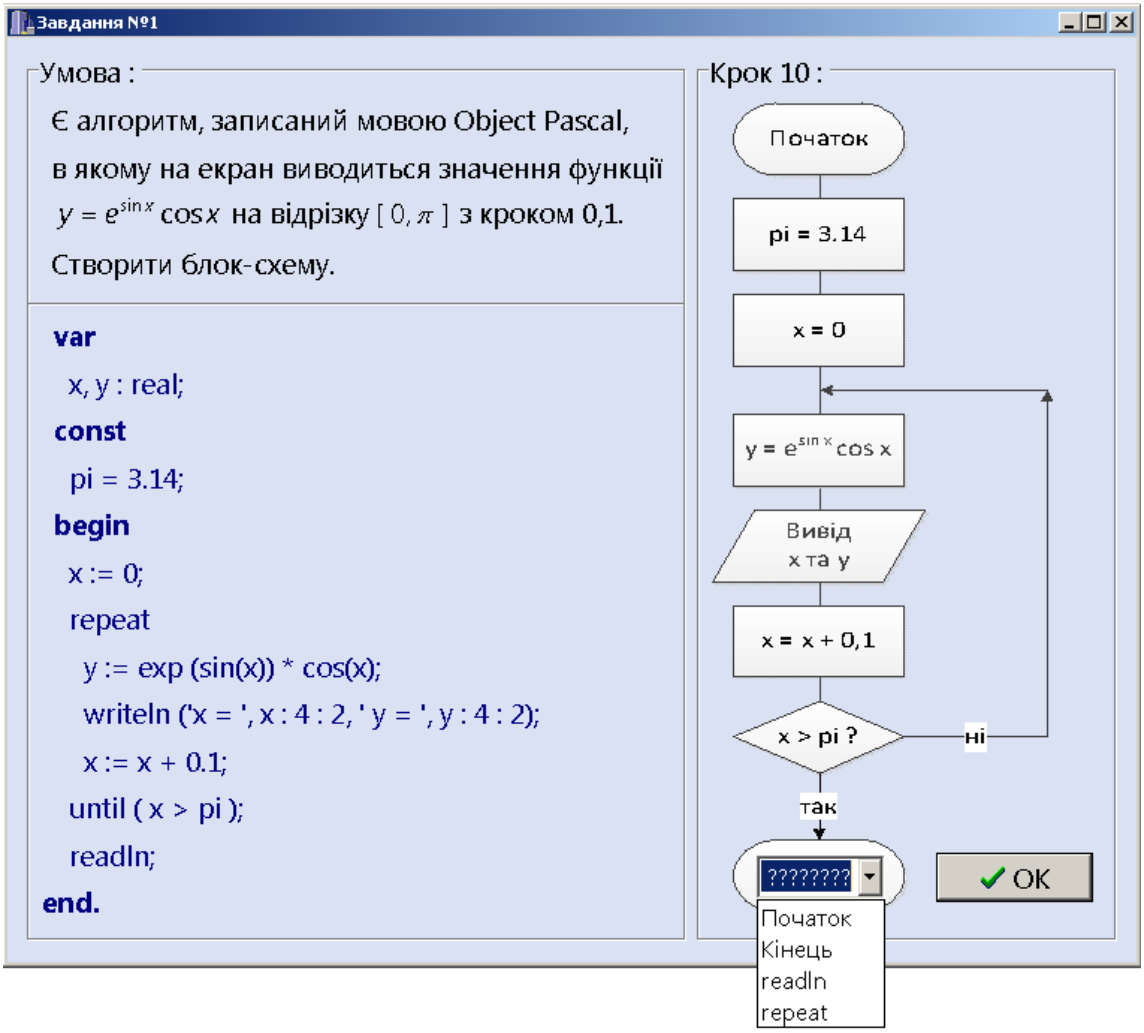

Рисунок 4.27 – Крок 10

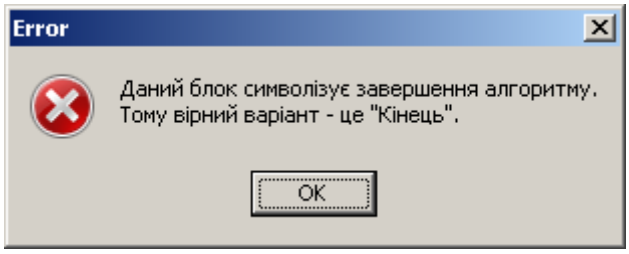

Рисунок 4.28 – Пояснення помилки на кроці 10

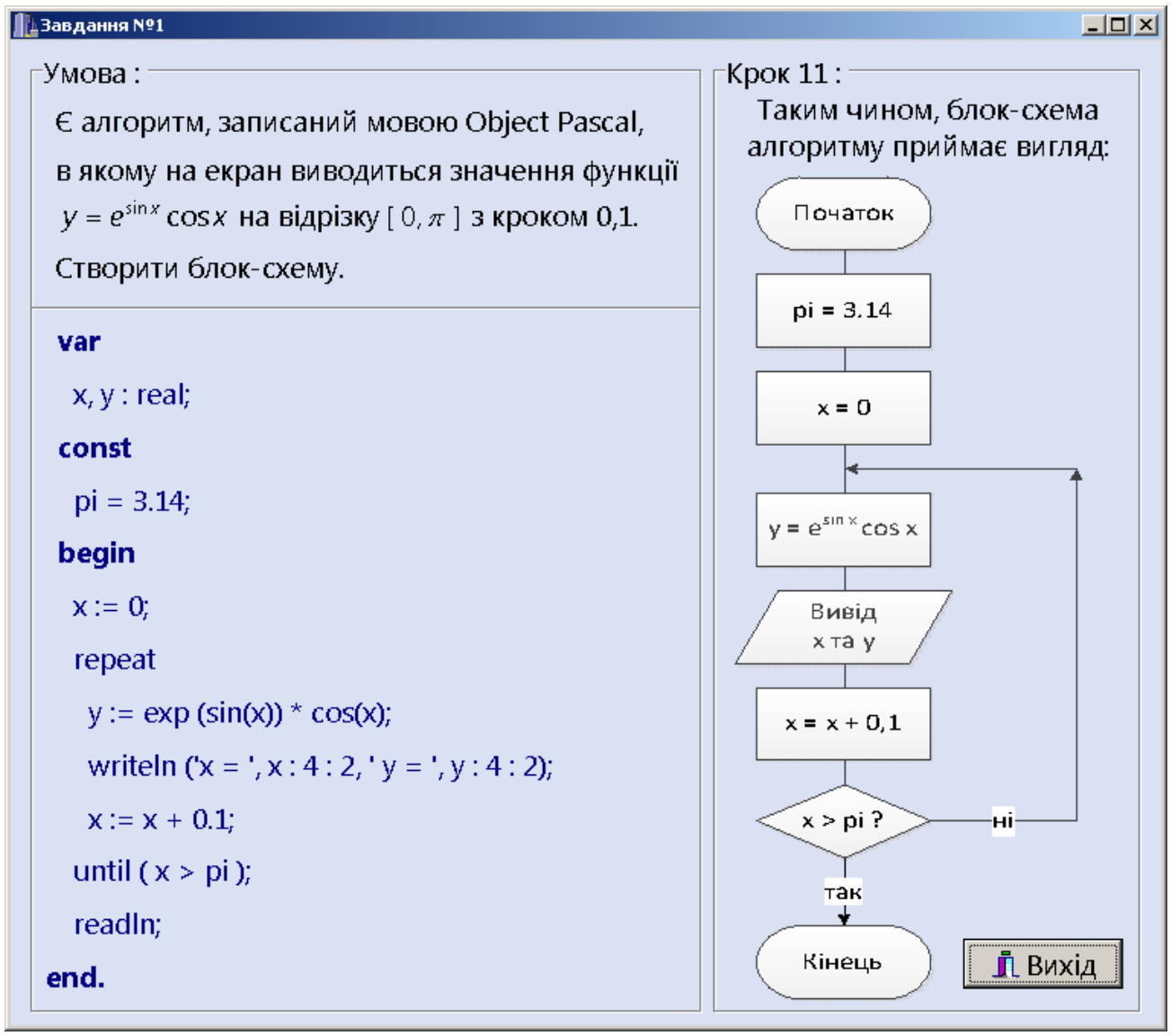

Рисунок 4.29 – Крок 11

ВИСНОВКИ

При роботі над бакалаврською роботою було досліджено тему «Цикли мови Object Pascal». Детально було вивчено тему «Цикл repeat … until мови Object Pascal».

Також була оглянута тема «Побудова блок-схем алгоритмів».

Було взято два приклади програм з використанням циклу repeat … until. Перевірено, що ці програми коректно працюють в середовищі Delphi. Для цих програм побудовані блок-схеми.

На основі даних програм і повних блок-схем розроблено алгоритм симулятору, що складається з двох прикладів. У першому прикладі 11 кроків, у другому – 17 кроків. Для цього алгоритму створено блок-схему та програмну реалізацію мовою C++ в середовищі Borland Builder.

Створений симулятор був протестований на предмет помилок та друкарських описок. Всі похибки усунуті.

Для програми створено документацію.

Програмний комплекс передано впровадження у дистанційний курс «Інформатика. Частина 1» Полтавського університету економіки і торгівлі, що засвідчує акт впровадження.

Результати випускової роботи пройшли апробацію на науково-практичному семінарі «Комп’ютерні науки і прикладна математика (КНіПМ-2021)», що відбувся на базі кафедри математичного моделювання та соціальної інформатики Полтавського університету економіки і торгівлі. Були опубліковані тези.

СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Ємець О. О. Дистанційний курс Полтавського університету економіки та торгівлі «Інформатика. Частина 1» для студентів спеціальностей «Комп'ютерні науки та інформаційні технології», «Комп'ютерні науки» / О. О. Ємець.

2. Схемы алгоритмов, данных и систем. Условные обозначения и правила выполнения: ГОСТ 19.701-90. – М.: Из-во стандартов, 1990. – 25 с.

3. Алексеев Е. Р. Самоучитель по программированию на Free Pascal и Lazarus / Е. Р. Алексеев, О. В. Чеснокова, Т. В. Кучер. – Донецк.: ДонНТУ, Технопарк ДонНТУ УНИТЕХ, 2009. – 503 с.

4. Культин Н. Б. Delphi 6. Программирование на Object Pascal / Н. Б. Культин. – СПб: БХВ-Санкт-Петербург, 2004. – 526 с.

5. Барболіна Т. М. Використання комп'ютерних тренажерів при вивченні теорії ігор / Т. М. Барболіна // матеріали V Всеукр. наук.-практ. конф. – Полтава, 2019. – Режим доступу: http://dspace.pnpu.edu.ua/bitstream/123456789/13239/1/Barbolina.pdf.

6. Бибка Б. М. Тренажер «Побудова блок-схем алгоритмів циклічної структури на прикладі циклу while» / Б. М. Бибка, О. О. Ємець // Комп'ютерні науки і прикладна математика (КНіПМ-2020): матеріали наук.-практ. семінару. Випуск 6. / За ред. Ємця О. О. – Полтава: Кафедра ММСІ ПУЕТ, 2020. – Режим доступу: http://dspace.puet.edu.ua/handle/123456789/10039.

7. Гмиза Б. Ю. Тренажер з теми «Побудова блок-схем алгоритмів циклічної структури на прикладі циклу for» дистанційного навчального курсу «Інформатика» та розробка його програмного забезпечення / Б. Ю. Гмиза, О. О. Ємець // Комп'ютерні науки і прикладна математика (КНіПМ-2019): матеріали наук.-практ. семінару. В. 3. / За ред. Ємця О. О. – Полтава: Кафедра ММСІ ПУЕТ, 2019. – С. 38-39. – Режим доступу: http://dspace.puet.edu.ua/handle/123456789/7036.

8. Мордасова І. В. Тренажер з теми «Побудова блок-схем алгоритмів розгалуженої структури» дистанційного навчального курсу «Інформатика» та

розробка його програмного забезпечення / І. В. Мордасова, О. О. Ємець // Комп’ютерні науки і прикладна математика (КНіПМ-2019): матеріали наук.-практ. семінару. Випуск 3. / За ред. Ємця О. О. – Полтава: Кафедра ММСІ ПУЕТ, 2019. – С. 35-37. – Режим доступу: http://dspace.puet.edu.ua/handle/123456789/7037.

9. Ємець О. О. Про розробку тренажерів для дистанційних курсів навчальних дисциплін спеціальності Комп`ютерні науки в ПУЕТ / О. О. Ємець, Є. М. Ємець, О. О. Ємець // Економіка сьогодні: проблеми моделювання та управління: матеріали X Міжн. наук.-практ. інтернет-конф. (м. Полтава, 19-20 листопада 2020 р.). – Полтава: ПУЕТ, 2021. – С. 365-370.

10. Олефіренко В. В. Розробка елементів тренажеру з теми «Сортування включеннями з лінійним та бінарним пошуком» дистанційного навчального курсу «Алгоритми та структури даних» та розробка його програмного забезпечення / В. В. Олефіренко // Комп’ютерні науки і прикладна математика (КНіПМ-2021): матеріали наук.-практ. семінару. Випуск 6. / За ред. Ємця О. О. – Полтава: ММСІ ПУЕТ, 2021. – Режим доступу: http://dspace.puet.edu.ua/handle/123456789/10041.

10. Сузанська А. О. Тренажер «Побудова блок-схем алгоритмів розгалуженої структури» / А. О. Сузанська, Є. М. Ємець, О. О. Ємець // Комп’ютерні науки і прикладна математика (КНіПМ-2020): матеріали наук.-практ. семінару. Випуск 5. / За ред. Ємця О. О. – Полтава: Кафедра ММСІ ПУЕТ, 2020. – С. 56-62. – Режим доступу: http://dspace.puet.edu.ua/handle/123456789/8906.

11. Шакуро В. Є. Розробка програмного забезпечення з теми «Побудова блок-схема алгоритмів лінійної структури» дистанційного курсу «Інформатика»/ В. Є. Шакуро, О. О. Ємець // Комп’ютерні науки і прикладна математика (КНіПМ-2019): матеріали наук.-практ. семінару. Випуск 3. / За ред. Ємця О. О. – Полтава: Кафедра ММСІ ПУЕТ, 2019. – С. 40-42. – Режим доступу: http://dspace.puet.edu.ua/handle/123456789/7038.

12. Ємець О. О. Методичні рекомендації до виконання бакалаврської роботи для студентів спеціальності 122 «Комп`ютерні науки та інформаційні технології» освітня програма «Комп`ютерні науки» галузь знань – 12 «Інформаційні технології» / О. О. Ємець. – Полтава: ПУЕТ, 2017. – 71 с.