

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСІЛКИ  
«Полтавський університет економіки і торгівлі»  
Навчально-науковий інститут заочно-дистанційного навчання  
Форма навчання заочна  
Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту  
Завідувач кафедри  
\_\_\_\_\_ О.ОЛЬХОВСЬКА  
(підпис)  
«\_\_\_» \_\_\_\_\_ 2023 р.

**КВАЛІФІКАЦІЙНА РОБОТА**

на тему  
**«СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТРЕНАЖЕРУ ЗА МОДУЛЕМ  
1 ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО КУРСУ “ОСНОВИ НАУКОВИХ  
ДОСЛІДЖЕНЬ В ІНФОРМАТИЦІ”»**

зі спеціальності 122 «Комп'ютерні науки»  
освітня програма «Комп'ютерні науки»  
ступеня магістра

**Виконавець роботи**

Литвиненко Алла Іванівна  
\_\_\_\_\_ «\_\_\_» \_\_\_\_\_ 20\_\_ р.  
(підпис)

**Науковий керівник**

к.ф.-м.н., доц., Черненко Оксана Олексіївна  
\_\_\_\_\_ «\_\_\_» \_\_\_\_\_ 20\_\_ р.  
(підпис)

**ПОЛТАВА 2023 р.**

**ЗАТВЕРДЖУЮ**Завідувач кафедри \_\_\_\_\_ Олена ОЛЬХОВСЬКА  
(підпис)

«\_\_» \_\_\_\_\_ 202\_\_ р.

**ЗАВДАННЯ ТА КАЛЕНДАРНИЙ ГРАФІК  
ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

**на тему «Створення програмного забезпечення тренажеру за модулем 1  
дистанційного навчального курсу “Основи наукових досліджень в інформатиці”»  
зі спеціальності 122 «Комп’ютерні науки»**

освітня програма «Комп’ютерні науки»

ступеня магістр

Прізвище, ім'я, по батькові Литвиненко Алла Іванівна

Затверджена наказом ректора № 138-Н від «01» вересня 2022 р.

Термін подання студентом роботи «\_\_» \_\_\_\_\_ 202\_\_ р.

Вихідні дані до кваліфікаційної роботи: публікації по темі роботи, методичні  
рекомендації, стандарти, матеріали дистанційного навчального курсу «Основи  
наукових досліджень в інформатиці», а також умови та алгоритми задач, з яких  
буде складатися тренажер.

Зміст пояснювальної записки (перелік питань, які потрібно розробити)

**ВСТУП****1. ПОСТАНОВКА ЗАДАЧІ**

- 1.1. Визначення завдань кваліфікаційної роботи.
- 1.2. Етапи виконання роботи.
- 1.3. Алгоритм роботи програми-тренажеру.

**2. ІНФОРМАЦІЙНИЙ ОГЛЯД**

- 2.1. Поняття електронного тренажеру як засобу підготовки майбутніх фахівців.
- 2.2. Актуальність теми.

**3. ТЕОРЕТИЧНА ЧАСТИНА**

- 3.1. Методологія створення програмного забезпечення.
- 3.2. Інформаційне наповнення програми-тренажеру.
- 3.3. Блок-схема програми-тренажера.
- 3.4. Мова програмування Java.
- 3.5. Інтегроване середовище програмування Eclipse.

**4. ПРАКТИЧНА ЧАСТИНА**

- 4.1. Створення проекту Java в інтегрованому середовищі Eclipse.
- 4.2. Створення програмного забезпечення тренажеру.

**ВИСНОВКИ****СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

## ДОДАТОК А

Перелік графічного блок-схеми алгоритму (2-4 блок-схем), рисунки (в необхідній кількості).

Консультанти розділів кваліфікаційної роботи

| Розділ              | Прізвище, ініціали, посада консультанта | Підпис, дата   |                  |
|---------------------|-----------------------------------------|----------------|------------------|
|                     |                                         | завдання видав | завдання прийняв |
| Постановка задачі   | Черненко О.О.                           |                |                  |
| Інформаційний огляд | Черненко О.О.                           |                |                  |
| Теоретична частина  | Черненко О.О.                           |                |                  |
| Практична частина   | Черненко О.О.                           |                |                  |

Календарний графік виконання кваліфікаційної роботи

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

Дата видачі завдання «\_\_» \_\_\_\_\_ 202\_\_ р.

Здобувач вищої освіти \_\_\_\_\_ Литвинено А.І.

(підпис)

Науковий керівник \_\_\_\_\_

(підпис)

к.ф.-м.н., доц. Черненко О.О.

(науковий ступінь, вчене звання, ініціали та прізвище)

### Результати захисту кваліфікаційної роботи

Кваліфікаційна робота оцінена на \_\_\_\_\_  
(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № \_\_\_\_ від «\_\_» \_\_\_\_\_ 2023 р.

Секретар ЕК \_\_\_\_\_

(підпис)

\_\_\_\_\_

(ініціали та прізвище)

**Затверджую**

Зав. кафедрою \_\_\_\_\_  
 к.ф.-м.н. Олена ОЛЬХОВСЬКА  
 « \_\_\_\_ » \_\_\_\_\_ 202\_\_ р.

**Погоджено**

Науковий керівник \_\_\_\_\_  
 к.ф.-м.н. Оксана ЧЕРНЕНКО  
 « \_\_\_\_ » \_\_\_\_\_ 202\_\_ р.

**План**

кваліфікаційної роботи ступеня магістр  
 зі спеціальності 122 Комп'ютерні науки  
 освітня програма 122 Комп'ютерні науки

Литвиненко Алла Іванівна

Прізвище, ім'я, по батькові

на тему «Створення програмного забезпечення тренажеру за модулем 1  
 дистанційного навчального курсу “Основи наукових досліджень в інформатиці”»

**ВСТУП****1. ПОСТАНОВКА ЗАДАЧІ**

- 1.4. Визначення завдань кваліфікаційної роботи.
- 1.5. Етапи виконання роботи.
- 1.6. Алгоритм роботи програми-тренажеру.

**2. ІНФОРМАЦІЙНИЙ ОГЛЯД**

- 2.3. Поняття електронного тренажеру як засобу підготовки майбутніх фахівців.
- 2.4. Актуальність теми.

**3. ТЕОРЕТИЧНА ЧАСТИНА**

- 3.6. Методологія створення програмного забезпечення.
- 3.7. Інформаційне наповнення програми-тренажеру.
- 3.8. Блок-схема програми-тренажера.
- 3.9. Мова програмування Java.
- 3.10. Інтегроване середовище програмування Eclipse.

**4. ПРАКТИЧНА ЧАСТИНА**

- 4.3. Створення проекту Java в інтегрованому середовищі Eclipse.
- 4.4. Створення програмного забезпечення тренажеру.

**ВИСНОВКИ****СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ****ДОДАТОК А**

Здобувач вищої освіти \_\_\_\_\_ А. ЛИТВИНЕНКО  
 « \_\_\_\_ » \_\_\_\_\_ 202\_\_ р.

## АНОТАЦІЯ

**Записка:** 73 сторінок, 20 рисунків, 2 таблиці, 1 додатка (на 21 сторінках), 15 літературних джерел.

**Мета роботи** – проектування та створення програмного забезпечення тренажеру за першим модулем дистанційного курсу «Основи наукових досліджень в інформатиці».

**Об'єкт розробки** – дистанційне навчання студентів спеціальності «Комп'ютерні науки» за дистанційним навчальним курсом «.

**Методи дослідження та інформаційне забезпечення** - використання матеріалів першого модуля дистанційного курсу «Основи наукових досліджень в інформатиці», використання редактора Eclipse на мові програмування Java та мовах розмітки HTML, CSS. Тренажер розроблено на основі віконної програми з використанням бібліотек Java. Програма-тренажер містить не лише питання тестового характеру, а й методичні рекомендації з вивчення 1 модулю дистанційного курсу та теоретичну довідку при виборі неправильної відповіді.

**Результати дослідження.** Виконано огляд поняття програми-тренажера, його видів та особливостей використання. Розроблено алгоритм програмного забезпечення тренажеру за модулем 1 дистанційного навчального курсу «Основи наукових досліджень в інформатиці», побудовано його блок-схему. Виконана програмна реалізація тренажеру в інтегрованому середовищі Eclipse мовою програмування Java.

**Рекомендації щодо використання результатів дослідження.** Програмне забезпечення тренажеру за модулем 1 дистанційного навчального курсу «Основи наукових досліджень в інформатиці» можна використовувати для самостійного опрацювання відповідних тем дисципліни здобувачами освіти за спеціальністю «Комп'ютерні науки».

**Наукова апробація.** За матеріалами роботи розроблено тези для наукового семінару «Комп'ютерні науки». За матеріалами роботи опубліковано статтю на Всеукраїнській онлайн-платформі для викладачів «ВСЕОСВІТА».

*Мова програмування, програма-тренажер, java, eclipse, середовище програмування, освітня діяльність.*

## ЗМІСТ

### ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І

|                                                                             |           |
|-----------------------------------------------------------------------------|-----------|
| <b>ТЕРМІНІВ.....</b>                                                        | <b>8</b>  |
| <b>ВСТУП.....</b>                                                           | <b>9</b>  |
| <b>1. ПОСТАНОВКА ЗАДАЧІ .....</b>                                           | <b>11</b> |
| 1.1. Визначення завдань курсового проєкту.....                              | 11        |
| 1.2. Етапи виконання роботи.....                                            | 11        |
| 1.3. Алгоритм роботи програми-тренажеру .....                               | 12        |
| <b>2. ІНФОРМАЦІЙНИЙ ОГЛЯД .....</b>                                         | <b>13</b> |
| 2.1. Поняття електронного тренажеру як засобу підготовки майбутніх фахівців | 13        |
| 2.2. Актуальність теми .....                                                | 15        |
| <b>3. ТЕОРЕТИЧНА ЧАСТИНА .....</b>                                          | <b>17</b> |
| 3.1. Методологія створення програмного забезпечення .....                   | 17        |
| 3.2. Інформаційне наповнення програми-тренажеру .....                       | 21        |
| 3.3. Блок-схема програми-тренажера.....                                     | 26        |
| 3.4. Мова програмування Java.....                                           | 26        |
| 3.5. Інтегроване середовище програмування Eclipse .....                     | 33        |
| <b>4. ПРАКТИЧНА ЧАСТИНА .....</b>                                           | <b>36</b> |
| 4.1. Створення проєкту Java в інтегрованому середовищі Eclipse.....         | 36        |
| 4.2. Створення елементів програмного забезпечення тренажеру .....           | 41        |
| <b>ВИСНОВКИ.....</b>                                                        | <b>51</b> |
| <b>ВИКОРИСТАНІ ДЖЕРЕЛА .....</b>                                            | <b>52</b> |
| <b>ДОДАТОК А. ПРОГРАМА.....</b>                                             | <b>54</b> |

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

| Умовні позначення, символи,<br>скорочення, терміни | Пояснення умовних<br>позначень, скорочень,<br>символів |
|----------------------------------------------------|--------------------------------------------------------|
| IDE                                                | інтегроване середовище<br>програмування                |
| JSP                                                | Java Server Pages                                      |
| WWW                                                | World Wide Web                                         |
| ББК                                                | бібліотечно-<br>бібліографічна<br>класифікація         |
| ВНЗ                                                | вищий навчальний заклад                                |
| НДР                                                | науково-дослідна робота                                |
| УДК                                                | універсальна десяткова<br>класифікація                 |

## ВСТУП

Важко уявити сьогодення без сервісів мережі Інтернет та ІТ. Сучасні люди тісно пов'язані з мережевим соціальним спілкуванням та інформаційними ресурсами. Можна спостерігати як інформаційні технології розповсюджуються на всі сфери людського життя великими кроками. В зв'язку з цим виникає необхідність використання нових світових інформаційних розробок і в освітній діяльності.

Сучасні здобувачі освіти 21 століття значно відрізняються від попередніх поколінь тим, яким способом отримує та сприймає інформацію, саме тому потрібно використовувати нові методи та підходи, які зможуть зацікавити їх працювати наполегливо на досягнення результату. Викладач повинен бути наставником, що націлює студента на пошук нового, цікавого та оригінального. Щоб йти в один крок зі здобувачами освіти та гармонійно з ними співпрацювати, потрібно, перш за все, цікавитися сучасними тенденціями, пізнавати те, що їм цікаво. А це і є саме використання ресурсів мережі Інтернет, ІТ в цілому, хмарні технології, додатки та візуальні веб-сервіси, що дозволяють отримати миттєвий результат.

Враховуючи останні зміни в нашому суспільстві та в світі, перед майбутніми фахівцями постають завдання, що потребують комплексного підходу й використання знань різних галузей. Вміння шукати інформацію, швидко її аналізувати та перетворювати і в результаті використовувати в своїй роботі, це одна з ознак креативного сучасного фахівця. Крім того, сучасна молодь, при виконанні завдань, вимагає майже миттєвого отримання результату, не чекаючи інших реакцій чи сторонніх думок.

Актуальність роботи полягає в створенні програмного забезпечення тренажеру, що дає можливість студенту застосувати його для самостійного та дистанційного вивчення першого модуля дистанційного навчального курсу

“Основи наукових досліджень в інформатиці”. Це в подальшому дозволить легко використовувати дану циклічну структуру, що спрощує час в написанні коду та автоматизує процес.

*Мета кваліфікаційної роботи* – створити програмного забезпечення тренажеру за модулем 1 дистанційного навчального курсу “Основи наукових досліджень в інформатиці”.

*Об’єктом роботи* є матеріали першого модуля дистанційного навчального курсу “Основи наукових досліджень в інформатиці”.

*Предметом розробки* є віконний програмний продукт на мові програмування Java, що в подальшому дозволить покращувати рівень знань за першим модулем дистанційного курсу «Основи наукових досліджень в інформатиці».

*Методи розробки* – об’єктно-орієнтована мова програмування Java та технології й інструментальні засоби створення комп’ютерних додатків, середовище програмування Eclipse та використання матеріалів першого модуля дистанційного курсу «Основи наукових досліджень в інформатиці»

Кваліфікаційна робота складається з чотирьох розділів, а саме: постановка задачі, інформаційний огляд, теоретична частина, практична частина.

Результатом виконання кваліфікаційної роботи є створення програмного забезпечення тренажеру за 1 модулем дистанційного курсу «Основи наукових досліджень в інформатиці» на мові програмування Java у середовищі Eclipse.

Обсяг пояснювальної записки: 73 сторінки, 20 рисунків, 2 таблиці, 1 додаток (на 21 сторінках), 15 літературних джерел.

## **1. ПОСТАНОВКА ЗАДАЧІ**

### **1.1. Визначення завдань курсового проєкту**

Основним завданням кваліфікаційної роботи є створення програмного забезпечення тренажеру за модулем 1 дистанційного курсу «Основи наукових досліджень в інформатиці», користуючись яким під час освітнього процесу здобувачі освіти можуть перевірити рівень засвоєння навчального матеріалу з даного модулю та покращити свої результати переглядаючи теоретичний матеріал.

Програма-тренажер повинна містити методичні рекомендації з вивчення першого модулю курсу та запитання у тестовій формі двох рівнів складності, що супроводжуються теоретичною довідкою при виборі чи введенні відповіді.

*I рівень складності:*

Тестові запитання з вибором правильного варіанту відповіді. Якщо здобувач освіти відповідає на питання неправильно, то програма-тренажер повертає його до теоретичного матеріалу з даного питання.

*II рівень складності:*

Здобувач освіти повинен ввести відповідь на питання самостійно у відповідне текстове поле і якщо відповідь невірна, то з'являється вікно з теоретичним матеріалом по даному питанню.

### **1.2. Етапи виконання роботи**

Щоб програма-тренажер виконувала повноцінно та ефективно свою основну освітню функцію, необхідно покроково розробити всі її елементи та дії, враховуючи деталі виконання кожного з них. Тому було визначено наступні етапи виконання даної роботи:

I етап. Постановка завдань. На даному етапі визначається мета та завдання майбутньої програми-тренажера та її ціль при подальшому використанні під час освітнього процесу.

II етап. Створення описової інформаційної моделі майбутньої програми-тренажера.

III етап. Побудова логічної структури та алгоритму роботи програми-тренажера.

IV етап. Вибір середовища та мови програмування, які в подальшому впливатимуть на можливості тренажера.

V етап. Створення елементів програмного забезпечення тренажеру в обраному середовищі відповідною мовою програмування.

VI етап. Перевірка складових частин програми-тренажера, тестування її під час роботи та, за потреби, її корекція чи вдосконалення.

### **1.3. Алгоритм роботи програми-тренажеру**

Для створення ефективного програмного забезпечення тренажеру за модулем 1 дистанційного навчального курсу «Основи наукових досліджень в інформатиці» потрібно розробити чіткий алгоритм його дії, що реалізує всі поставлені задачі.

**Крок 1.** Вікно реєстрації здобувача освіти.

**Крок 2.** Ознайомлення зі структурою та методичними рекомендаціями по вивченню теоретичного матеріалу 1 модулю дистанційного навчального курсу «основи наукових досліджень в інформатиці».

**Крок 3.** Постановка питань першого рівня складності.

При неправильній відповіді програма-тренажер повертає до теоретичного матеріалу по даному питанню.

При виборі правильної відповіді програма-тренажер переходить до наступного питання.

**Крок 4.** Постановка питань другого рівня складності

При неправильній відповіді програма-тренажер повертає до теоретичного матеріалу по даному питанню.

При виборі правильної відповіді програма-тренажер переходить до наступного питання.

**Крок 5.** Виведення результатів роботи з програмою.

## 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

### 2.1. Поняття електронного тренажеру як засобу підготовки майбутніх фахівців

Тренування є складовою освіти. Комп'ютерні тренажери використовують в тих випадках, коли фактично практична підготовка фахівців неможлива, вимагає значних витрат або є необхідність залучити велику кількість здобувачів освіти. Для здобуття навичок розв'язання типових завдань використовують програми-тренажери, які пропонують користувачу виконати низку завдань і контролюють процес його роботи. Така програма одразу видає результат та дає можливість виправити помилки, якщо такі є, допомагає правильно виконати завдання, аналізує результати роботи в цілому. Програми-тренажери є надзвичайно зручними для освіти та дозволяють відпрацювати різні уміння й навички. [2]

Використання тренажеру в освіті дозволяє:

- ✓ послідовно виводити на екран завдання різного рівня з обраної теми;
- ✓ виконувати контроль за розв'язком завдання на різних етапах;
- ✓ реагувати на невірний обраний крок;
- ✓ виправляти помилки здобувача освіти;
- ✓ демонструвати правильний алгоритм розв'язку з детальним поясненням;
- ✓ отримувати підсумкову оцінку роботи користувача з поясненнями та коментарями.

Завдання, що вирішуються під час використання програм-тренажерів:

- ✓ **інформування** (підручники, довідники, навчальні плакати, відеозаписи, алгоритми, покажчики, інформаційно-технологічні карти, навчальні програми та тренажери);
- ✓ **організація** (алгоритми, покажчики, інформаційно-технологічні карти, навчальні програми та тренажери);
- ✓ **управління** (тести, навчальні програми та тренажери). [5]

Застосування в освітньому процесі комп'ютерного тренування показало, що в здобувачів освіти з'являється впевненість у собі, а можливість самоконтролю є додатковою мотивацією до вивчення дисциплін. Використання програми дає змогу перейти на вищий рівень засвоєння знань, що у свою чергу підвищує якість підготовки учня (рис. 2.1).

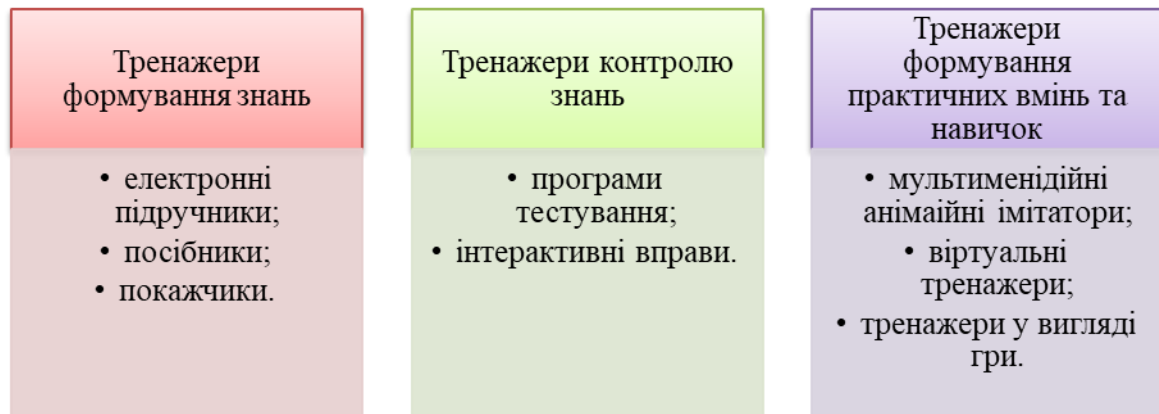


Рисунок 2.1. - Види навчальних тренажерів

Використання таких засобів в освітньому процесі має ряд переваг та недоліків, що відображаються як і на діяльності здобувачів освіти так і на його результаті.

Переваги використання:

- ✓ відображає зв'язки з практичною стороною;
- ✓ можливість використання в дистанційній формі освіти;
- ✓ з психологічної точки зору студент швидко пристосовується до віртуального способу отримання знань;
- ✓ в процесі виконання вправ є можливість контролювати хід виробничого процесу;
- ✓ в процесі виконання вправ студент здобуває кваліфікацію і формує вміння розподіляти свої фізичні та психічні сили;
- ✓ в процесі засвоєння нових знань, навичок та кваліфікації досягає повної свободи;
- ✓ визнаючи свої помилки, усуває їх через повторення імітаційного віртуального становища.

Недоліки використання:

- ✓ завдання в практичному комп'ютерному тренажері не завжди можуть враховувати всі відхилення виробничого процесу, що можуть вивникнути в житті;
- ✓ зацікавленість до віртуального процесу настільки посилюється, що фахівець більше цікавиться підсумком процесу ніж прагненням вивчати сам процес;
- ✓ в результаті постійного використання на наступних рівнях навчання не фахівець буде контролювати процес, а навпаки, процес буде управляти фахівцем. [5]

З цієї причини при використанні тренажерів необхідно звернути особливу увагу на запобігання виникненню негативних проявів. Зокрема, доцільно точно встановити час використання тренажерів і його дотримуватися.

## **2.2. Актуальність теми**

Світ веб-розробки постійно змінюється. Досить важливим є уважно стежити за всім, що відбувається: актуальність існуючих фреймворків, вибір оптимального серед них, тенденції розробки та оформлення сайтів та інші.

Розробнику важливо бути в курсі поточних технологічних розробок, наскільки це можливо. Він повинен дивитися на можливості, але також враховувати можливі ризики, пов'язані з новими технологіями. Регулярні експерименти з різними структурами допомагають відчутти, як потенційні вигоди, так і обмеження.

Навчання у ВНЗ дає найголовніше вміння – самостійно знаходити та дослідити новий матеріал. Більшість студентів оцінюють курси як місце для отримання не знань, а поштовху для пошуку престижної роботи. Проте насправді більшість із них приведуть студента-слухача до конкретного проміжного фінішу, що завершиться сертифікатом. Найголовніша цінність будь-якого курсу – це не працевлаштування, а розвиток у студента критичного мислення та вміння думати не шаблонами, а опираючись на власні навички та досвід. [1]

У зв'язку з високим попитом на онлайн-курси на просторах мережі Інтернет створюється велика кількість онлайн-тренажерів та посібників. Перевага даних ресурсів полягає в тому, що сам користувач вирішує, коли йому зручно витратити час на виконання того чи іншого завдання. Онлайн-курси слідує такому принципу як дотримання часових рамок здачі завдання. У свою чергу, онлайн-тренажер/посібник не містять обмежень у часі.

На сьогоднішній день існує велика кількість тренажерів різного формату, серед яких наступні:

- тренажери для отримання навиків сліпого введення;
- тренажери для вивчення правил дорожнього руху;
- тренажери для вивчення іноземних мов;
- тренажери для вивчення мов програмування. [5]

Тому майбутній онлайн-тренажер необхідно розробити в форматі, де користувач зможе обрати як тренажер для освоєння теоретичного матеріалу, так і тренажер із динамічними підказками або перейти до виконання практичного завдання для закріплення знань теоретичного матеріалу.

### 3. ТЕОРЕТИЧНА ЧАСТИНА

#### 3.1. Методологія створення програмного забезпечення

Щоб створити дійсно якісний програмний продукт, потрібно чітко визначити весь його життєвий цикл. Це чіткі кроки, що дозволяють зрозуміти, що повинно вийти в розробника, як досягти позитивного результату та які методи для цього використати.

*Методологія розробки програмного забезпечення* – це перевірені способи та практики, що дозволяють створити програмний продукт правильно та якісно.

В ІТ існує декілька основних методик розробки програмного забезпечення.

Кожен проект має власний життєвий цикл. Це етапи, якими проходить вся розробка. Починається він із створення ідеї та прийняття рішень.

Далі виконується підготовка та аналіз, пошук концепції та шляху створення продукту.

Під час етапу аналітики ідея перетворюється на покроковий чіткий план дій, підбираються технології, обираються й методології розробки програмного забезпечення.

Вибір методики дозволяє досягти стабільності під час розробки, а це одне з основних завдань.

Сам процес розробки програмного забезпечення складається з приблизно однакових етапів. Але як ці етапи будуть реалізовуватися, як вони пов'язані між собою та за якими принципами працюють – визначає методологія розробки ПЗ. Її завдання наступні:

- допомагає зрозуміти, як реалізувати ідею та створити продукт, корисний для визначеного осередку;
- дає чітке уявлення про методи роботи та план дій, що знижує ймовірність помилок під час розробки;
- допомагає заощадити на етапі розробки, мінімізуючи кількість правок та прискорюючи процес завдяки чіткому плану дій;

- дозволяє розподіляти завдання залежно від обраного методу;
- створює послідовність процесів, формує вимоги до продукту.

Необхідність обирати методологію розробки програмного забезпечення обґрунтована націленістю на результат. В іншому випадку можна почати створювати продукт й ніколи його не закінчити, оскільки не буде чіткого плану дій. За основу завжди беруться 7 базових та працюючих методик.

### **Основні методології розробки**

Кожна модель розробки програмного забезпечення має свої унікальні особливості, переваги та недоліки. Визначити, яка з них краща – не можна, оскільки під різні завдання, продукти та ідеї обирається свій принцип розробки.

#### **"Waterfall Model" (каскадна модель або "водоспад")**

Це класична модель, яка використовувалася на початку епохи розробки, але продовжує активно застосовуватись й сьогодні. Принцип роботи даної методології побудований лінійно: кожний наступний етап виконується лише тоді, коли повністю завершено попередній. Існує чітке розподілення етапів, й технологія реалізується ніби каскадом, поступово рухаючись від першого до останнього етапу.

Ця модель розробки ПЗ досить жорстка, має суворі правила. Вона чітко визначає терміни проходження кожного етапу. Але є недолік: зробити крок назад дуже складно.

#### **"Incremental Model" (інкрементна модель)**

Інкрементну модель розробки програмного забезпечення використовують в тих випадках, коли є чітко визначені кроки, але продукт потрібно запустити швидко, а зміни можна буде вносити потім. Принцип роботи полягає в тому, що спочатку розробляється план дій, що, в свою чергу, розбивається на невеликі завдання. Потім кожне завдання розробляється за традиційною каскадною моделлю. Спочатку робиться «базовий» продукт з мінімальними, але важливими функціями. Поступово він доповнюється завдяки розробці інших компонентів, які називаються інкрементами. Процес зациклюється, доки не буде повністю зібраної єдиної системи.

Кожна частина роботи в такому проєкті є готовим елементом. Іноді його можна використовувати окремо. Як правило, він розробляється таким чином, щоб потім його не переробляти. Саме тому й використовується каскадна модель всередині інкрементної моделі.

Що пропонує така методологія розробки програмного забезпечення? Насамперед, мінімізуються ризики, забезпечується швидкий реліз та запуск продукту. Крім того, базовий функціонал вже працюватиме й приносить користь для бізнесу, а за необхідності завжди можна впровадити нові сформовані інструменти.

### **"Iterative Model" (ітеративна або ітераційна модель)**

Такий підхід за своєю конструкцією дещо нагадує попередній. Принцип використання ітеративної методології розробки програмного забезпечення полягає в тому, щоб створити базові елементи та поступово їх покращувати. Схоже з попереднім варіантом, але є різниця.

Інкрементну модель можна порівняти з пазлом, де всі елементи викладаються один за одним й поступово збираються до єдиної картинки.

Такий метод підходить для великих проєктів, у якому визначаються основні завдання та є загальне уявлення, що має вийти. Недолік цієї методології полягає в тому, що поняття кінцевого продукту є розмитим. Такий підхід дозволяє поступово додавати нові компоненти та покращувати наявні.

### **"Agile Model" (гнучка методологія розробки)**

Гнучка методологія розробки програмного забезпечення використовується тоді, коли кінцевий результат не відомий, а особливість даного методу полягає в тому, що розробник та потенційний користувач можуть одразу спостерігати за змінами у розробці та коригувати дії.

### **"Spiral Model" (спіральна модель)**

Спіральні моделі розробки ПЗ підходять для довготривалих проєктів, де помилки призводять до великих втрат та негативних наслідків.

Принцип роботи даної методології полягає в здійснюється 4-х основних дій: створення плану, аналіз ризиків, розробка та конструювання, оцінка результату та

збирання відгуків. Якщо всі 4 етапи успішно пройдені, то технологія переходить до наступного кроку.

Перехід на новий крок нагадує інкрементну модель, де кожен блок розробляється незалежно та приєднується до спочатку створеного базового елементу. Особливість підходу полягає в тому, що більше часу йде на аналіз, розрахунки та оцінку наслідків, ніж безпосередньо на саму розробку. Розробники іноді набагато швидше реалізують функцію, ніж вона вивчається аналітиками.

### **"V-Model" (V-подібна модель)**

Дана методологія робить акцент на тестування та перевірку працездатності систем під час розробки. Тести виконуються одночасно із самим процесом створення продукту. Сам принцип успадковує базовий підхід при каскадній розробці. Процес іде покроково, існує чіткий план дій, складається суворе технічне завдання. Але паралельно виконуються тести, у разі виявлення помилок вони одразу виправляються, незалежно від етапу розробки.

До нового кроку можна перейти лише тоді, коли відсутні всі помилки. При цьому на новому кроці тести аналізують й усі попередні. Це дозволяє контролювати взаємозв'язок всіх елементів та їх працездатність.

### **"RAD Model" (rapid application development model - швидка розробка додатків)**

Дана методологія підходить тим розробникам, які хочуть максимально швидко реалізувати програмний продукт. Суть у тому, що етапи створення додатка розподіляються на кілька окремих блоків. [10]

Після цього робочі невеликі модулі збираються до єдиної системи та утворюють робочий прототип, який можна реалізовувати. Така методологія має низку особливостей:

- клієнт має брати активну участь в розробці та постійно спостерігати за результатами;
- вартість використання такого методу висока, оскільки доведеться наймати великий штат розробників;

- Необхідно чітко розуміння, що має вийти у результаті, щоб кожна команда знала, яку задачу вирішує її модуль.

### **3.2. Інформаційне наповнення програми-тренажеру**

Основним завдання програми-тренажеру є підвищення рівня знань здобувачів освіти за 1 модулем дистанційного навчального курсу «Основи наукових досліджень в інформатиці», тому необхідно чітко визначити теоретичну наповнюваність тренажеру та сформулювати питання двох рівнів складності для самоперевірки знань.

Перший модуль дистанційного навчального курсу «Основи наукових досліджень в інформатиці» передбачає вивчення чотирьох тем:

Тема 1. Організація науково-дослідної роботи у вищому навчальному закладі;

Тема 2. Поняття науки та її нормативне регулювання;

Тема 3. Методологічні засади наукових досліджень;

Тема 4. Інформаційне забезпечення у наукових дослідженнях.

У процесі самостійного вивчення модуля особливу увагу слід звернути на означення таких понять:

- ✓ науково-дослідна робота (НДР),
- ✓ об'єкт,
- ✓ суб'єкт наукової діяльності,
- ✓ наука,
- ✓ наукова діяльність,
- ✓ глобалізація науки,
- ✓ інтеграція науки,
- ✓ наукове дослідження,
- ✓ об'єкт,
- ✓ предмет дослідження,
- ✓ метод,
- ✓ методика,

- ✓ методологія,
- ✓ інформація, первинна, вторинна інформація,
- ✓ бібліографічний опис,
- ✓ каталог.

У процесі самостійного вивчення модуля слід розглянути:

- ✓ особливості сучасної науки,
- ✓ види і форми НДР,
- ✓ загальну класифікацію наук,
- ✓ форми наукового міжнародного співробітництва,
- ✓ загальнофілософські методи,
- ✓ загально-наукові методи,
- ✓ часткові методи дослідження,
- ✓ класифікацію інформаційного забезпечення,
- ✓ принципи каталогів.

Слід приділити особливу увагу таким питанням:

- ✓ організація НДР у ВНЗ,
- ✓ функції науки, пріоритетні напрямки розвитку науки в Україні,
- ✓ класифікація загальнонаукових методів та приклади їх використання
- ✓ інформаційно пошукова мова бібліотечно-бібліографічного типу: УДК, ББК;
- ✓ копірайт,
- ✓ код ISBN.

Враховуючи методичні рекомендації по вивченню даного модуля та вимоги до знань та вмінь здобувачів освіти по завершенню вивчення дисципліни, було обрано 12 питань. Дані питання мають два рівня складності. Перший рівень – це 8 питань тестового характеру з вибором варіанту відповіді. По 2 питання з кожної теми першого модуля. Другий рівень – це 4 питання з введенням відповіді в текстове поле. По 1 питанню з кожної теми.

*Тестові питання 1 рівня:*

1. Про яке поняття йде мова в наступному визначенні: «...- це інтелектуальна творча діяльність, спрямована на одержання і використання нових знань»?
  - а) Дослідницька діяльність;
  - б) *Наукова діяльність;*
  - в) Творча діяльність.
2. Що НЕ належить до видів науково-дослідницької роботи?
  - а) Самостійна науково-дослідна пошукова робота;
  - б) Науково-дослідна робота під науковим керівництвом;
  - в) *Творчо-дослідницька робота в навчальному закладі.*
3. Про яку функцію науки йде мова: «...удосконалення виробництва і системи суспільних відносин, тобто безпосередньої виробничої сили матеріального виробництва»?
  - а) Пізнавальну;
  - б) *Практичну;*
  - в) Культурно-виховну.
4. Що таке громадські спеціалізовані академії?
  - а) *Наукові установи, що об'єднують учених на громадських засадах за профілем їх наукової діяльності;*
  - б) Вища наукова організація України, яка організує і здійснює фундаментальні та прикладні дослідження з найважливіших проблем природничих, технічних і гуманітарних наук, а також координує здійснення фундаментальних досліджень в наукових установах та організаціях незалежно від форм власності;
  - в) Державні спеціалізовані наукові установи, що координують, організують і проводять дослідження у відповідних галузях науки і техніки.
5. Скільки рівнів пізнання виділяють в методології наукових досліджень?
  - а) 4;
  - б) 3;

- в) 2.
6. Метод дослідження, при якому загальний висновок про ознаки множини елементів виводиться на основі вивчення цих ознак у частини елементів однієї множини – це...
- а) *індукція*;
  - б) *дедукція*;
  - в) *інструкція*.
7. Вкажіть п'ять головних етапів в процесі підготовки та проведення будь-якого дослідження.
- а) *пошук та копіювання інформації; формулювання теми, мети і завдання дослідження; теоретичне дослідження; проведення експерименту; підведення підсумків;*
  - б) *накопичення наукової інформації; формулювання теми, мети і завдання дослідження; теоретичне дослідження; проведення експерименту; оформлення результатів наукового дослідження;*
  - в) *накопичення наукової інформації; визначення основних завдань; вивчення літератури; проведення експерименту, оформлення висновків.*
8. Основою інформаційно-пошукового апарату бібліотек є каталоги. В якому каталозі картки згруповані в логічному порядку за галузями знань?
- а) *В систематичних каталогах;*
  - б) *В алфавітних каталогах;*
  - в) *В предметних каталогах.*

*Питання другого рівня:*

9. Хто з суб'єктів науково-дослідницької роботи реалізовує свої можливості через участь у гуртках, проблемних групах, конкурсах студентських наукових робіт, практику тощо?

*Студенти.*

10. В даному законодавчому акті визначено, що кожен має право на освіту; громадянам гарантується свобода наукової і технічної творчості, захист

інтелектуальної власності, їхніх авторських прав, моральних і матеріальних інтересів, що виникають у зв'язку з різними видами інтелектуальної діяльності; держава сприяє розвитку науки, встановленню наукових зв'язків України зі світовим співтовариством.

*Конституція України.*

11.Метод наукового дослідження, завдяки якому досягається пізнання одних предметів і явищ на основі їх подібності з іншими.

*Аналогія*

12.Наука, яка вивчає будову і загальні властивості інформації, закономірності її створення, перетворення, передавання і використання у різних сферах діяльності людини.

*Інформатика. [11]*

Якщо здобувач освіти дає неправильну відповідь на питання, то програма-тренажер повертає його до теоретичного матеріалу з відповідної теми модуля, а після повторного ознайомлення пропонує ще раз дати відповідь на питання.

### 3.3. Блок-схема програми-тренажера



Рисунок 3.1 – Блок-схема роботи тренажера

### 3.4. Мова програмування Java

Для створення елементів програмного забезпечення тренажера за модулем 1 дистанційного навчання курсу «Основи наукових досліджень в інформатиці» я використала об'єктно-орієнтовану мову програмування Java.

*Мова програмування* – це структура позначень та символів для опису алгоритмів обробки даних, яка призначена для написання програмного забезпечення.

*Java* – це об'єктивно-орієнтована мова програмування для загального призначення. Метою її розробки є можливість створювати коди та використовувати їх у будь-якому місці.

Java дозволяє створювати самодостатні програми для різних операційних систем як то Windows, Linux тощо. Крім того, в даний час Java широко застосовується для програмування різних пристроїв, наприклад, мобільних телефонів, на ній також пишуться комп'ютерні ігри для них, створюють також програми для інтернету - Аплети і програми для серверів - Сервлети та JSP (Java Server Pages). [6]

Започатковано мову Java у проекті фірми Sun Microsystems під назвою Green (1991). Головними розробниками першої робочої версії були Джеймс Гослінг (James Gosling), Патрік Ноутон (Patric Naughton), Кріс Ворс (Chris Warth), Ед Френк (Ed Frank) і Майкл Шерідан (Mike Sheridan). До 1995 року мову називали Оак, однак не пройшовши перевірку на допустимість торгової марки, було перейменовано на Java. Ідеєю створення нової мови програмування були не потреби Internet, а необхідність створення програмного забезпечення, яке не залежить від платформи (тобто архітектури) і використовується в побутових електронних приладах. Під час відпрацювання деталей Java виник вагомий чинник, який відіграв важливу роль щодо цієї мови. Таким чинником була всесвітня 6 інформаційна служба World Wide Web (WWW), розквіт якої припадає на 1994–1995 р. Ідеї створення ефективних і незалежних (працюючих на різноманітних процесорах під керівництвом різних операційних систем) програм такі ж давні, як і саме програмування, і завжди витіснялися нагальнішими проблемами. Якщо зважити на те, що більшість програмістів належить до одного з трьох кланів (Intel, Macintosh, UNIX), які безупинно змагаються між собою, то зрозумілим стає те, що нагальної потреби в переміщенні коду довгий час не виникало. Проте з виникненням Internet і WWW проблема переміщення програм

стала втричі гострішою. Різко змінилися акценти: від створення коду для вбудованих контролерів побутової техніки до програмування для Internet. Це і спричинило великий успіх мови Java. [8]



Рисунок 3.2 – Базові принципи мови програмування Java

Головною вимогою щодо мови для роботи в розподіленому просторі комп'ютерів (наприклад, в Internet) – це можливість працювати на різнорідних і розподілених платформах (рис. 3.2).

**Багатопоточність.** У багатопоточних операційних системах для кожного застосування (процесу) надається окрема захищена область пам'яті, в якій зберігаються коди програми і дані. А час одного процесора квантується між цими процесами. З метою запуску процесу або переключення з одного на інший на рівні операційної системи необхідно виконати значний об'єм роботи. Тому для розробників прикладних програм спеціально створили "полегшену" версію системного процесу – потік.

Найбільшою проблемою, пов'язаною з процесами і потоками, є їхнє функціонування під керівництвом конкретної операційної системи. [8] Спеціалісти

компанії Sun зробили потоки частиною мови програмування. Тому багатопоточне застосування, написане мовою Java, працюватиме і в операційних середовищах Windows, Unix, MacOS.

**Висока продуктивність.** Інтерпретатор Java може виконувати байткоди зі швидкістю, яка наближається до швидкості виконання коду, відкомпільованого до машинного формату, що досягається завдяки використуванню інтерпретатором багатьох потоків виконання. Наприклад, доки комп'ютер чекає на введення даних, фонові потоки можуть зайнятися очищенням пам'яті. [8]

**Стійкість до помилок.** Java – це мова строгого використання типів, що зумовлює зменшення числа помилок під час написання програми. У мові Java відбувається автоматична перевірка виконання граничних умов під час роботи з масивами і стрічками, які в Java є класами.

У Java немає арифметики покажчиків, а керування пам'яттю здійснюється автоматично. Програмний код, написаний мовою Java не може зіслатися на пам'ять поза простором програми або зробити помилку внаслідок вивільнення пам'яті і тим самим вичерпати всю пам'ять. У Java організовано процес автоматичного збирання сміття, тобто об'єктів, на які більше ніхто не вказує.

**Безпека.** Функції забезпечення безпеки дуже важливі для розподілених мереж з безліччю вірусів, "троянських коней" і т. п. Для реалізації цієї мети розробники мови Java створили механізм, який отримав назву пісочниці (sandbox):

- перевірку на рівні JVM;
- захист на рівні мови;
- інтерфейс Java Security (цифрового підпису).

[6]

Java – мова, яка чітко дотримується типів, що значною мірою гарантує безпеку програм і стійкість до помилок. Що це означає?

По-перше, кожна змінна має свій тип, і кожний тип чітко визначено.

По-друге, усі присвоєння, які виконують в явному вигляді або через пересилання параметрів у методи, перевіряються на співпадання типів.

У Java не передбачено автоматичне приведення типів або перетворення конфліктуючих типів, як в інших мовах.

Компілятор Java перевіряє усі вирази і параметри щодо сумісності типів.

## Прості типи

До простих типів у Java належать чотири групи:

1. Цілочисельні: `byte`, `short`, `int` і `long`, які призначені для цілих зі знаком;
2. Числа з плаваючою комою: `float` і `double` – числа з дробовою частиною;
3. Символи: `char` – всі символи, включаючи букви і цифри;
4. Логічні: `boolean` – "істина" і "фальш".

Ці типи використовують у програмах, об'єднують в масиви тощо.

Чому введено термін прості типи? Тому, що вони не є об'єктами.

Незважаючи на те, що в усьому іншому Java – цілковито об'єктно-орієнтована мова, прості типи є винятком. Все пояснюється ефективністю. Перетворення простих типів в об'єкти спричинило б до суттєвого зниження продуктивності. Для простих типів мови Java явно задається діапазон значень і чітко визначені правила виконання операцій над ними. У мові Java, виходячи з потреб мобільності, усі типи даних мають строго визначений діапазон. Це дає змогу писати програми з впевненістю, що вони будуть працювати на комп'ютерах з будь-якою архітектурою. Строге задання розмірів цілого приводить до деякого зниження швидкості виконання програм у деяких середовищах, однак цього не уникнути, коли головною метою є мобільність.

### Цілочисельні типи.

У Java визначено чотири цілочисельні типи: `byte`, `short`, `int` і `long`. Усі вони визначають знакові, тобто додатні і від'ємні значення. Java не підтримує без знакові цілі. У табл. 3.1 наведено ширину і діапазони цілочисельних типів.

Таблиця 3.1

### Цілочисельні типи

| Тип                | Ширина | Діапазон                                         |
|--------------------|--------|--------------------------------------------------|
| <code>long</code>  | 64     | від - 9223372036854775808 до 9223372036854775807 |
| <code>int</code>   | 32     | від - 2147483648 до 2147483647                   |
| <code>short</code> | 16     | від - 32768 до 32767                             |
| <code>byte</code>  | 8      | від - 128 до 127                                 |

Ширину цілочисельного типу, не дивлячись на те, що її вимірюють в бітах, треба розглядати не як об'єм пам'яті, яку він займає, а як поведінку, визначену для змінних і виразів заданого типу. Виконуване середовище Java використовує свою

кількість байтів, яка краще відповідає апаратній платформі, забезпечуючи у цьому випадку поведінку, визначену через заданий програмістом тип. Скоріше всього виконуване середовище Java відводить на деяких платформах 32 біти для типів `byte` і `short`. Змінні типу `byte` особливо корисні при роботі з потоком даних або файлом, а також з неформатованими даними. `Short` – цей тип рідко використовують, тому що розміщення старшого і молодшого байтів не завжди збігається з реальним підходом, який використовують переважно на 16-тирозрядних комп'ютерах. `Int` – найуживаніший цілочисельний тип. Його необхідно використовувати при створенні лічильника, індексації масивів, виконанні обчислень над цілими числами. Іноді здається, що використання `byte` чи `short` економить пам'ять, однак найвірогідніше, що конкретне середовище виконання все одно перетворить ці типи в `int`. Необхідно пам'ятати, що тип в мові Java визначає поведінку, а не розмір пам'яті. Єдиним винятком є масив, коли `byte` гарантує використання 8-мибітів на елемент масиву, `short` – 16-тибітів, а `int` – 32-хбітів.

Змінні типу `long` необхідні в тих випадках, коли тип `int` є замалим для збереження бажаного результату. [8]

#### *Дійсні типи.*

У Java існує два дійсних типи `float` і `double` для представлення чисел з одинарною і подвійною точністю відповідно. Основні характеристики цих типів наведено в табл. 3.2.

Таблиця 3.2

#### *Дійсні типи*

| Тип                 | Ширина | Діапазон                                    |
|---------------------|--------|---------------------------------------------|
| <code>double</code> | 64     | від $-1.7\text{e-}308$ до $1.7\text{e}+308$ |
| <code>float</code>  | 32     | від $-3.4\text{e-}038$ до $3.4\text{e}+038$ |

Змінні типу `float` використовують для представлення чисел з дробовою частиною, проте з незначною точністю. На деяких процесорах операції зі змінними типу `float` виконуються швидше, ніж з типами `double` і вимагають менше пам'яті. Проте коли значення стають дуже великими або дуже малими втрачається точність обчислень. Тип `double` використовують для представлення дійсних чисел з

подвійною точністю і він вимагає біти пам'яті. На деяких сучасних процесорах, оптимізованих для високошвидкісних математичних обчислень, операції з подвійною точністю виконуються швидше ніж, з одинарною. Наприклад, усі математичні функції так і як `sin()`, `cos()`, `sqrt()` повертають значення типу `double`. Якщо необхідно виконати багато ітерацій або оперувати з великими числами, краще вибирати `double`.

### *Символи.*

Виходячи з того, що мову Java створено для всесвітнього використання, для кодування символів використовують Unicode (повну інформацію можна отримати за адресою <http://www.unicode.org>). Unicode визначає повний міжнародний символний набір, який дає змогу представляти усі символи мов усіх народів світу. Тому необхідно для символу 16 бітів. Отже, `char` умові Java є 16-тибітовим типом. Значення типу `char` є цілі числа з діапазону від 0 до 65536. Стандартний набір символів, відомий як ASCII, займає інтервал від 0 до 127, а розширений 8-мибітовий символний набір (ISO-Latin-1) - від 0 до 255. Значення змінних `char` можна додавати та інкрементувати.

### *Логічний тип.*

У мові Java є тип `boolean`, який використовується для задання двох значень: `true` ("істина") і `false` ("фальш"). На відміну від мови C++, де будь-яке число відмінне від нуля є істина, в мові Java цей тип не є числом.

### *Змінні*

Змінна – це базова одиниця для збереження інформації в програмі мовою Java. Вона визначається комбінацією типу, ідентифікатора і не обов'язкової ініціалізації. Змінна має область видимості і час існування.

Оголошення змінної. У мові Java всі змінні повинні бути оголошені ще до їхнього використання. Основна форма оголошення змінної має вигляд:

Тип ідентифікатор [=значення][,ідентифікатор[=значення]...]; Під типом розуміють один із простих типів Java, ім'я класу або інтерфейсу. Деякі приклади оголошення змінних у Java:

```
int a,b,c;
```

```
int d=2, e, f=5;  
byte z=22;  
double pi=3.1415926;  
char x;
```

### 3.5. Інтегроване середовище програмування Eclipse

Для створення елементів програми-тренажеру мовою Java у курсовому проєкті було використано поширене інтегроване середовище програмування (IDE) Eclipse. Використання IDE надає можливості для швидшої та якіснішої роботи по розробці програмного забезпечення за допомогою мови програмування та полегшує більшість технічних моментів.

*Eclipse* – це вільне модульне інтегроване середовище розробки програмного забезпечення, що надає користувачу сукупність неухильних свобод, таких як:

- ❖ виконання програм будь-яким способом та для довільних цілей;
- ❖ можливість вивчати та удосконалювати програму, щоб вона задовольняла вимоги користувача (доступ до початкового тексту);
- ❖ свобода створення копій, як початкових програм так і змінених версій та надання їх іншим користувачам.

Спочатку інтегроване середовище Eclipse було комерційним продуктом, але починаючи з листопада 2001 року були опубліковані у вільному доступі його вихідні коди. Розробником середовища є компанія Object Technology International. [7]

Eclipse є інфраструктурою програмних рішень, для полегшення розробки модульних застосунків, тобто це своєрідна збірка об'єктів чи підпрограм для вирішення задач за певною тематикою, але не зважаючи на всі можливості існують і обмеження, що формують правила розробки структури проєктів та написання коду.

Основою інтегрованого середовища Eclipse є платформа розширеного клієнта, що складається з :

- ❖ ядра платформи (завантаження середовища, запуск модулів);
- ❖ середовища постачання комплектів;

- ❖ віджетів;
- ❖ файлових буферів, текстових редакторів, тощо;
- ❖ робочого середовища.



Рисунок 3.3 – Базові принципи мови програмування Java

По суті, середовище Eclipse – це своєрідна платформа для створення доповнень (плагінів), що потім включаються в програму. [7]

За допомогою вищезазначеного інтегрованого середовища можна розробити доповнення для різних мов програмування, а саме:

- ❖ C та C++\$
- ❖ Perl;
- ❖ Html;
- ❖ Php;
- ❖ Ruby, тощо.

Користуватися середовищем Eclipse можна у таких операційних системах:

- ❖ Mac OS;
- ❖ Linux;
- ❖ Microsoft Windows.

Виняткові функції модульного інтегрованого середовища розробки програмного забезпечення Eclipse сприяє мотивації користувача та збільшенню продуктивності його роботи. Особливо, дане середовище може стати актуальним для використання у освітній діяльності під час розробки програмного забезпечення

програм для урізноманітнення навчальної діяльності, результатом чого стане підвищення рівня знань здобувачів освіти.

## 4. ПРАКТИЧНА ЧАСТИНА

### 4.1. Створення проекту Java в інтегрованому середовищі Eclipse

Для розробки елементів програмного забезпечення тренажеру за 1 модулем дистанційного навчального курсу «Основи наукових досліджень в інформатиці» я використала мову програмування Java та інтегроване середовище Eclipse.

Перед початком роботи необхідно створити проєкт Java для розробки тренажеру. Для цього запускаємо програму Eclipse (рис. 4.1) та обираємо робочу папку для збереження файлів при роботі з програмою (рис. 4.2).



Рисунок 4.1 – Вікно завантаження Eclipse 2022-12

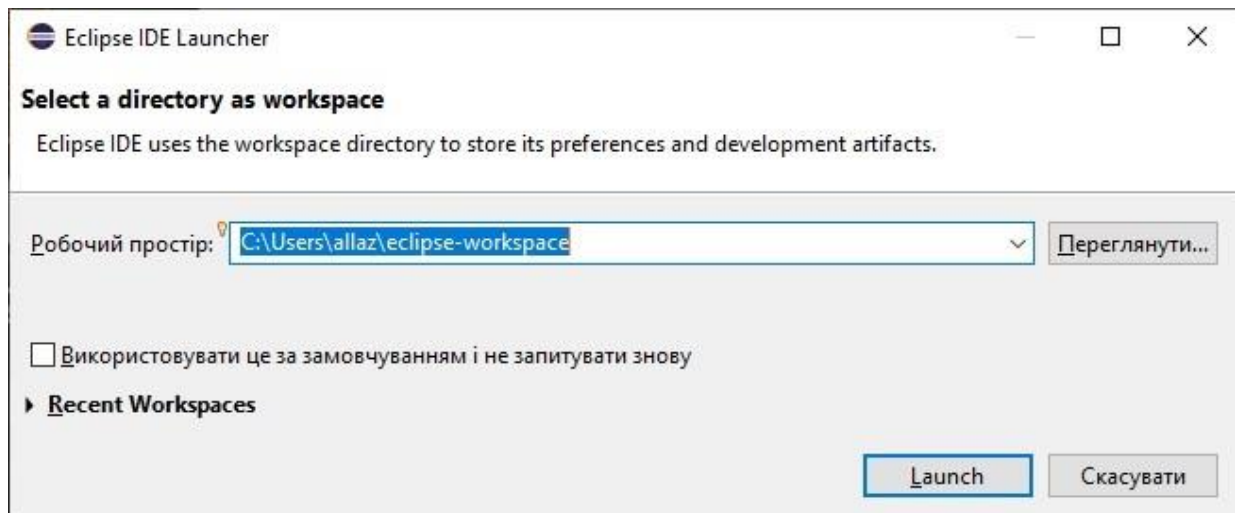


Рисунок 4.2 – Вибір папки для робочого простору з файлами в Eclipse

Вікно програми складається зі стандартних елементів, що присутні для всіх програм ОС Windows, це рядок заголовку, панель меню програми, кнопки управління вікном, та вікон інтегрованого середовища, що дозволяють працювати з робочим простором, кодом програми, консоль та інше (рис. 4.3).

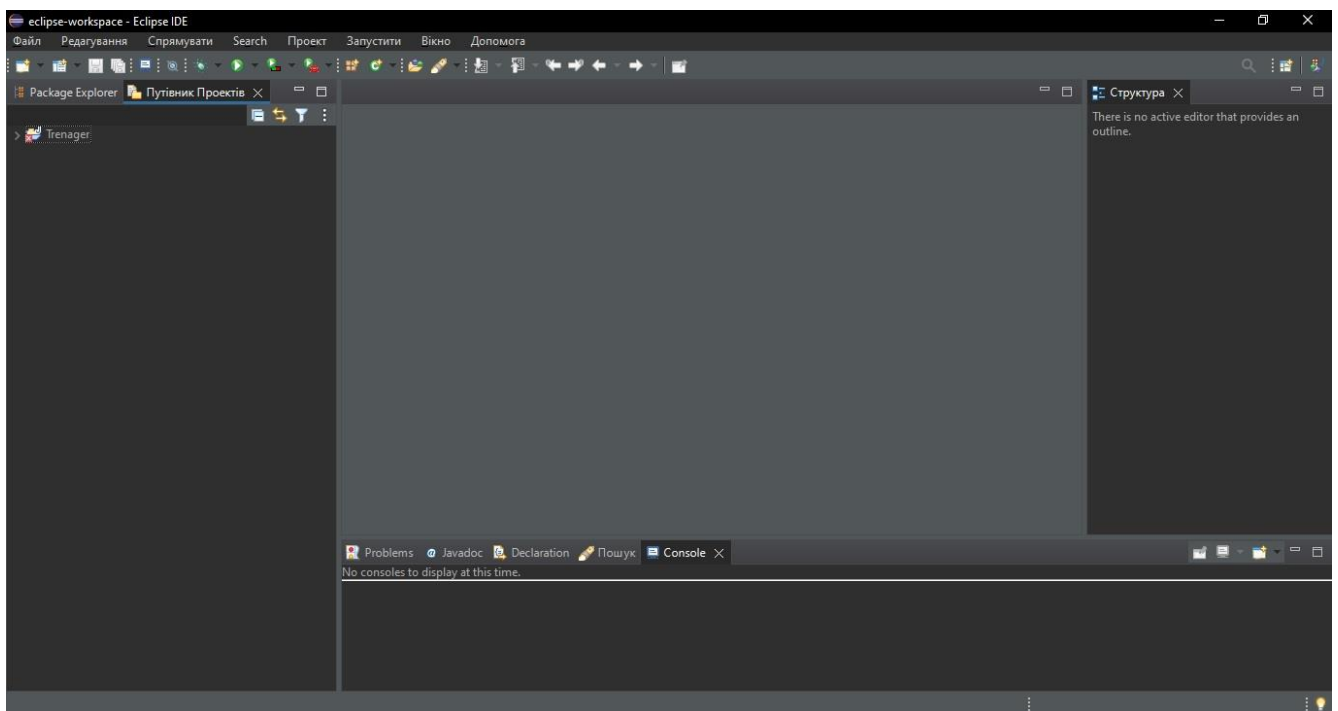


Рисунок 4.3 – Вікно програми Eclipse

Далі необхідно створити новий проєкт Java командою **Файл - Новий - Проект Java**. Вводимо ім'я майбутнього проєкту та обираємо папку для його

збереження на ПК. Можна залишити обрану папку, яку було вказано при відкритті інтегрованого середовища (рис. 4.4).

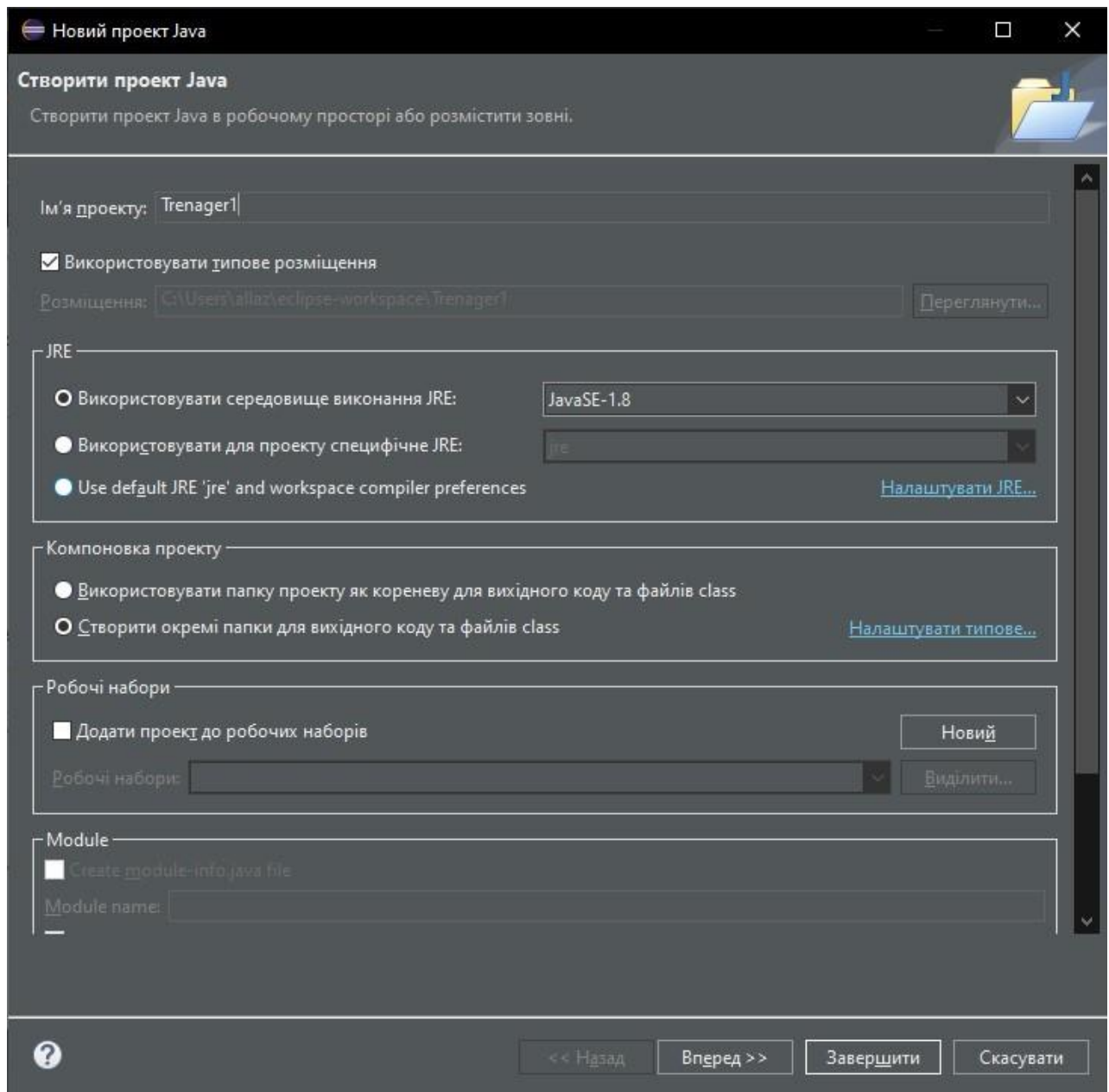


Рисунок 4.4 – Вікно створення нового проєсту Java

Для створення елементів проєкту потрібно в контекстному меню папки проєкту обрати пункт **Новий** та вказати тип створюваного елементу. Я обирала елемент для створення віконної програми **Application Windows** (рис. 4.5). Таким чином створювала відповідно класи та інші елементи, що використовуватимуться в проєкті. Програма Eclipse полегшує створення даних елементів використовуючи

відповідні бібліотеки Java, що попередні були інсталювані в програмі. Це спрощує роботу та прискорює процес розробки.

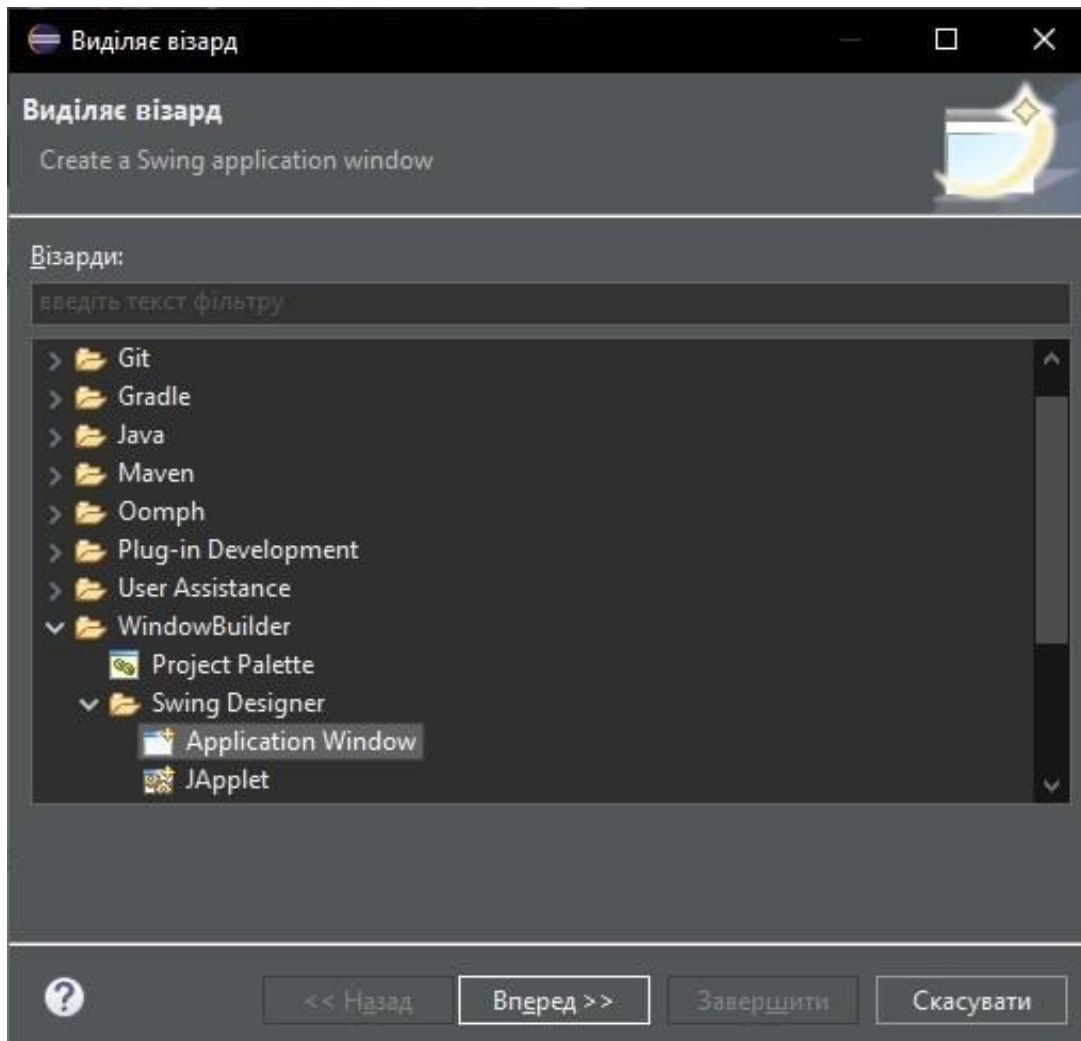


Рисунок 4.5 – Створення елементу *Application Window*

Після вибору елементу проєкту відкривається редактор коду, де відбувається безпосереднє створення програми (рис. 4.6). Інтегроване середовище має ряд засобів, що полегшують введення коду, виділяючи відповідні елементи різним кольором та повідомляючи користувача про наявні помилки. Крім того програма видає підказки для усунення помилок коду та автоматично може додати частини програмного коду в вибране відповідне місце.

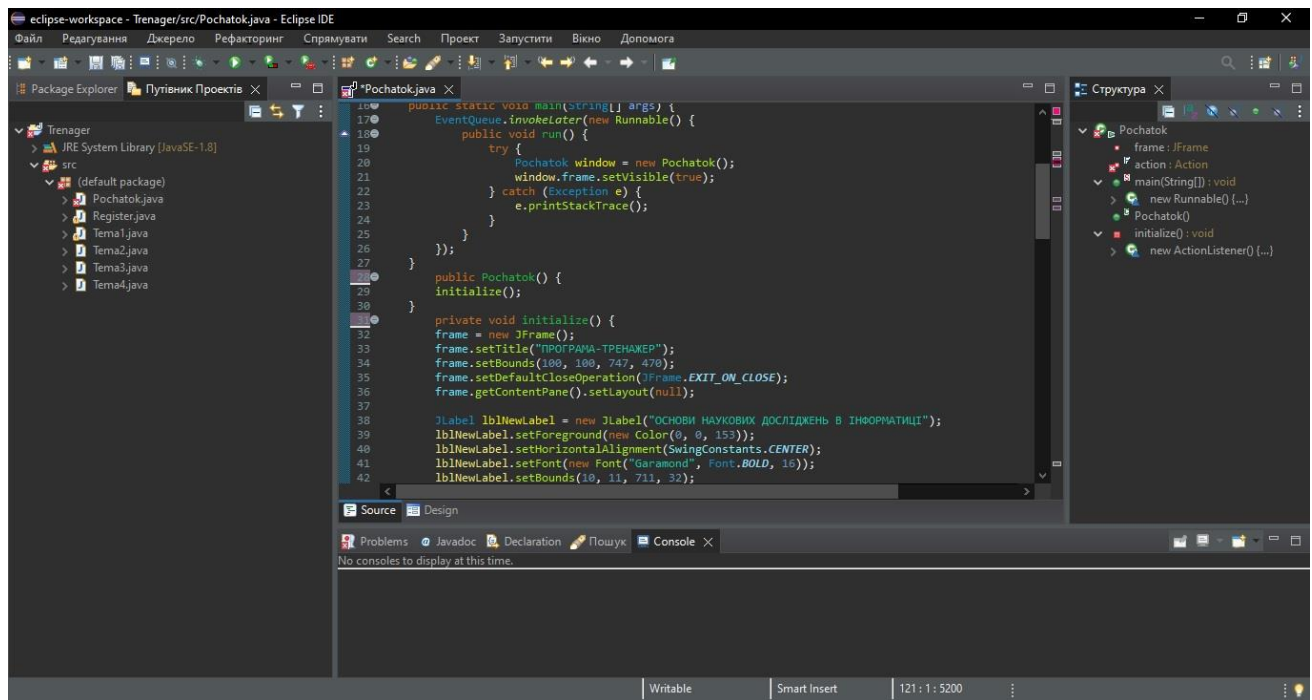


Рисунок 4.6 – Вікно програми з робочим простором для введення коду

Інтегроване середовище має в своєму арсеналі досить зручний інструмент у вигляді візуального редактора коду **Design** або Дизайнер (рис. 4.7). Цей інструмент доступний лише при роботі саме з віконною програмою Application Window. Він дозволяє відразу бачити результат та додавати у створене вікно різні елементи методом Drag and Draw. Кожен елемент має певні властивості, що розробник може застосувати в Дизайнері, пропустивши написання відповідного коду вручну. Але не всі властивості можна вказати в Дизайнері, код неодмінно потрібно редагувати та дописувати вручну. Наприклад, обробник подій вимагає написання коду вручну.

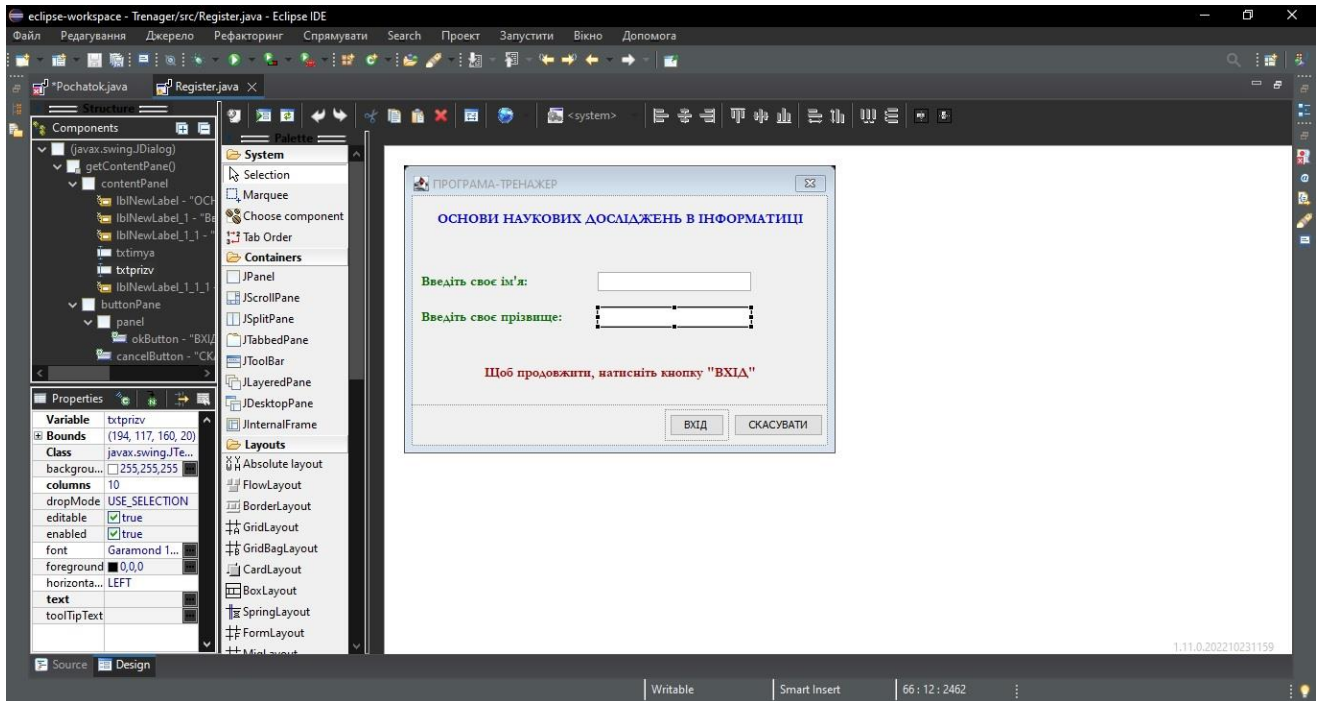


Рисунок 4.7 – Вікно програми з Дизайнером віконної програми

Таким чином можна створювати різні елементи програми, перевіряти їх наявність помилок та запускати в тестовому режимі, перевіряючи функціональність всіх елементів програми.

Тип віконної програми було обрано для майбутнього тренажера для більш комфортної з ним роботи, бо користувачі ПК звикли до вікон ОС Windows і при роботі з ними відчувають себе впевнено, знаючи велику кількість елементів таких вікон.

## 4.2. Створення елементів програмного забезпечення тренажера

Робота програми-тренажера за 1 модулем дистанційного навчального курсу “Основи наукових досліджень в інформатиці” починається з вікна реєстрації користувача. Тобто з введення його імені та прізвища для подальшого збереження результатів роботи з програмою-тренажером (рис. 4.8).

Рисунок 4.8 – Вікно реєстрації програми-тренажера

Дане вікно містить текстові поля для введення даних, а відповідно кнопка ВХІД виконує запис цих даних в змінні для їх подальшого використання програмою. Кнопка СКАСУВАТИ очищує текстові поля в разі неправильного введення даних. Перейти до наступного кроку дозволяє кнопка ВХІД. Повний код даного елемента міститься в Додатку А.

***Частина коду вікна реєстрації:***

...

```
public Register() {
    setBackground(new Color(255, 204, 204));
    setTitle("ПРОГРАМА-ТРЕНАЖЕР");
    setBounds(100, 100, 450, 300);
    getContentPane().setLayout(new BorderLayout());
    contentPanel.setBorder(new EmptyBorder(5, 5, 5, 5));
```

...

Наступний елемент – це вікно з довідкою по модулю 1 дистанційного навчального курсу «Основи наукових досліджень в інформатиці» (рис. 4.9). В даному вікні програми користувач ознайомлюється з темами модуля, може перейти та детальніше прочитати методичні рекомендації (рис. 4.10 – 4.13) до кожної теми модуля, натиснувши відповідні кнопки, та почати тестування.



Рисунок 4.9 – Вікно ознайомлення з вмістом першого модуля

***Частина коду вікна ознайомлення з темами модуля 1:***

...

```
private void initialize() {
    frame = new JFrame();
    frame.setTitle("ПРОГРАМА-ТРЕНАЖЕР");
    frame.setBounds(100, 100, 747, 470);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

...

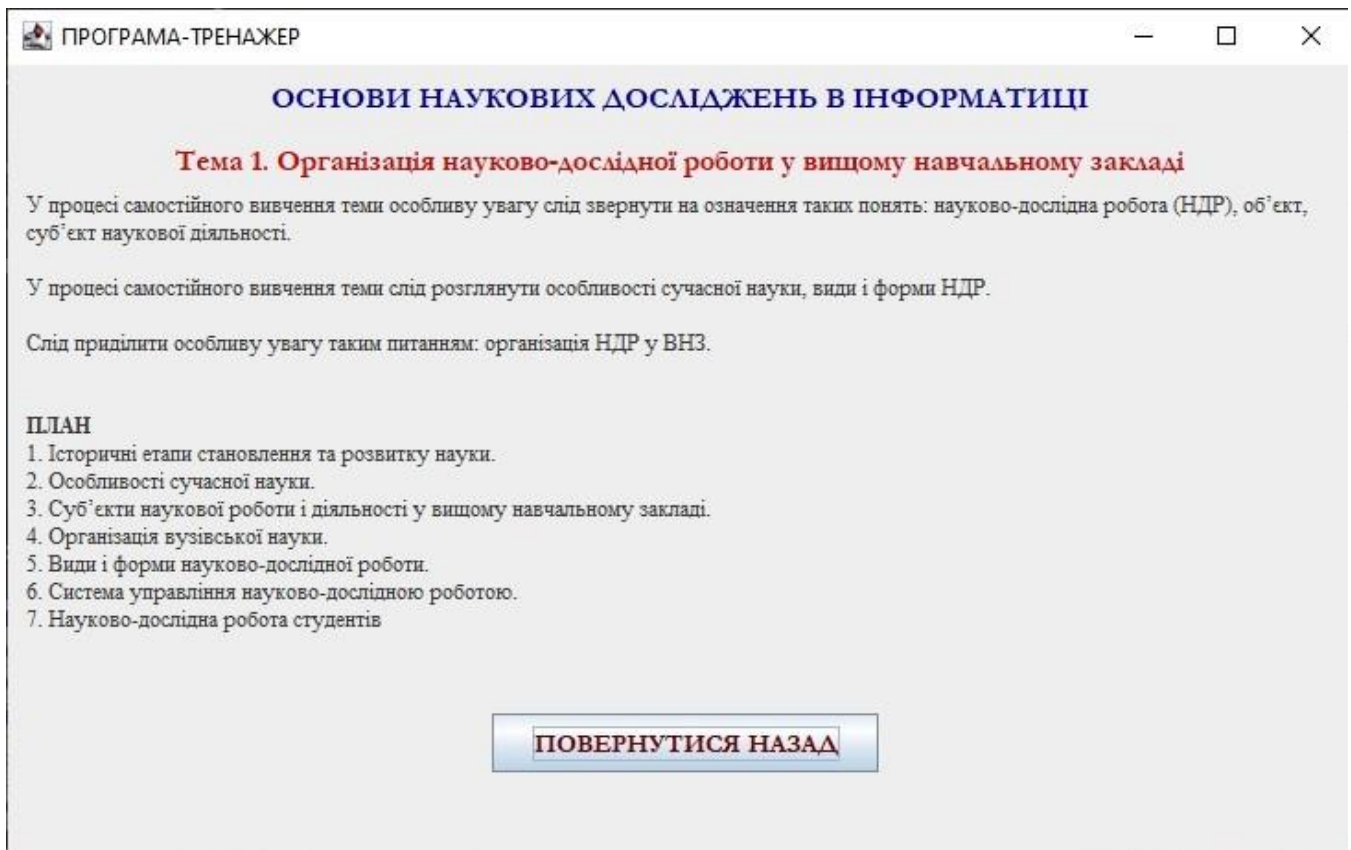


Рисунок 4.10 – Вікно ознайомлення з методичними рекомендаціями по Темі 1

**Частина коду вікна з методичними рекомендаціями по Темі 1:**

...

```
JLabel lblNewLabel = new JLabel("ОСНОВИ НАУКОВИХ ДОСЛІДЖЕНЬ В  
ІНФОРМАТИЦІ");
```

```
lblNewLabel.setBounds(0, 1, 731, 34);
```

```
lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
```

```
lblNewLabel.setForeground(new Color(0, 0, 153));
```

```
lblNewLabel.setFont(new Font("Garamond", Font.BOLD, 16));
```

```
frame.getContentPane().add(lblNewLabel);
```

```
JLabel lblNewLabel_1 = new JLabel("Тема 1. Організація науково-дослідної  
роботи у вищому навчальному закладі");
```

...

В даному вікні в елементі JLabel використано мову розмітки HTML для розміщення тексту з форматуванням.

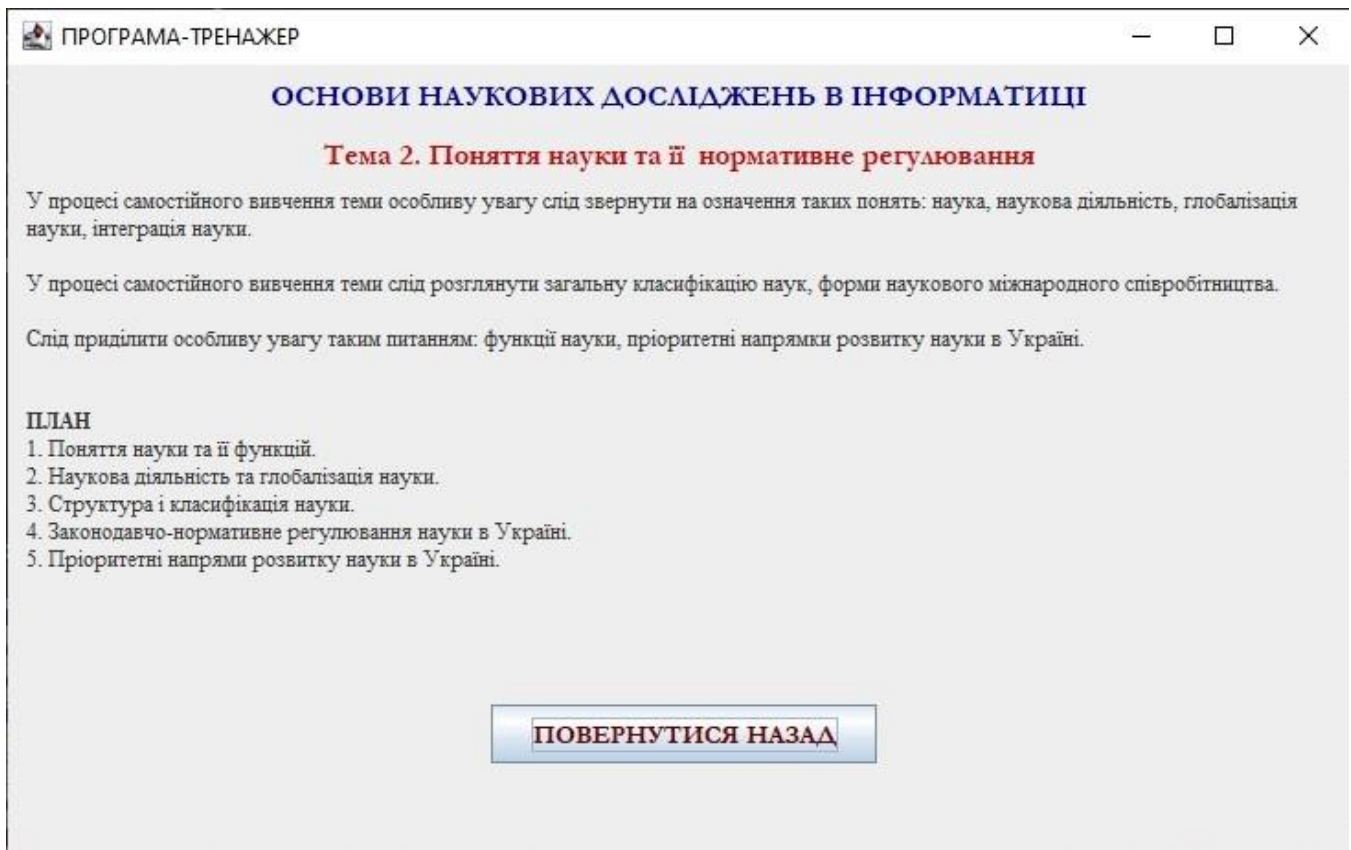


Рисунок 4.11 – Вікно ознайомлення з методичними рекомендаціями по Темі 2

**Частина коду вікна з методичними рекомендаціями по Темі 2:**

...

```
JLabel lblNewLabel_2 = new JLabel("<html> <p>У процесі самостійного  
вивчення теми особливу увагу слід звернути на означення таких понять: наука,  
наукова діяльність, глобалізація науки, інтеграція науки.<br> <br> У процесі  
самостійного вивчення теми слід розглянути загальну класифікацію наук, форми  
наукового міжнародного співробітництва.<br> <br> Слід приділити особливу увагу  
таким питанням: функції науки, пріоритетні напрямки розвитку науки в  
Україні.</p><br><br><b>ПЛАН</b><br><p>1. Поняття науки та її  
функцій.<br>\r\n2. Наукова діяльність та глобалізація науки.<br>\r\n3. Структура і  
класифікація науки.<br>\r\n4. Законодавчо-нормативне регулювання науки в  
Україні.<br>\r\n5. Пріоритетні напрями розвитку науки в Україні.</p></html>");  
lblNewLabel_2.setVerticalAlignment(SwingConstants.TOP);
```

...

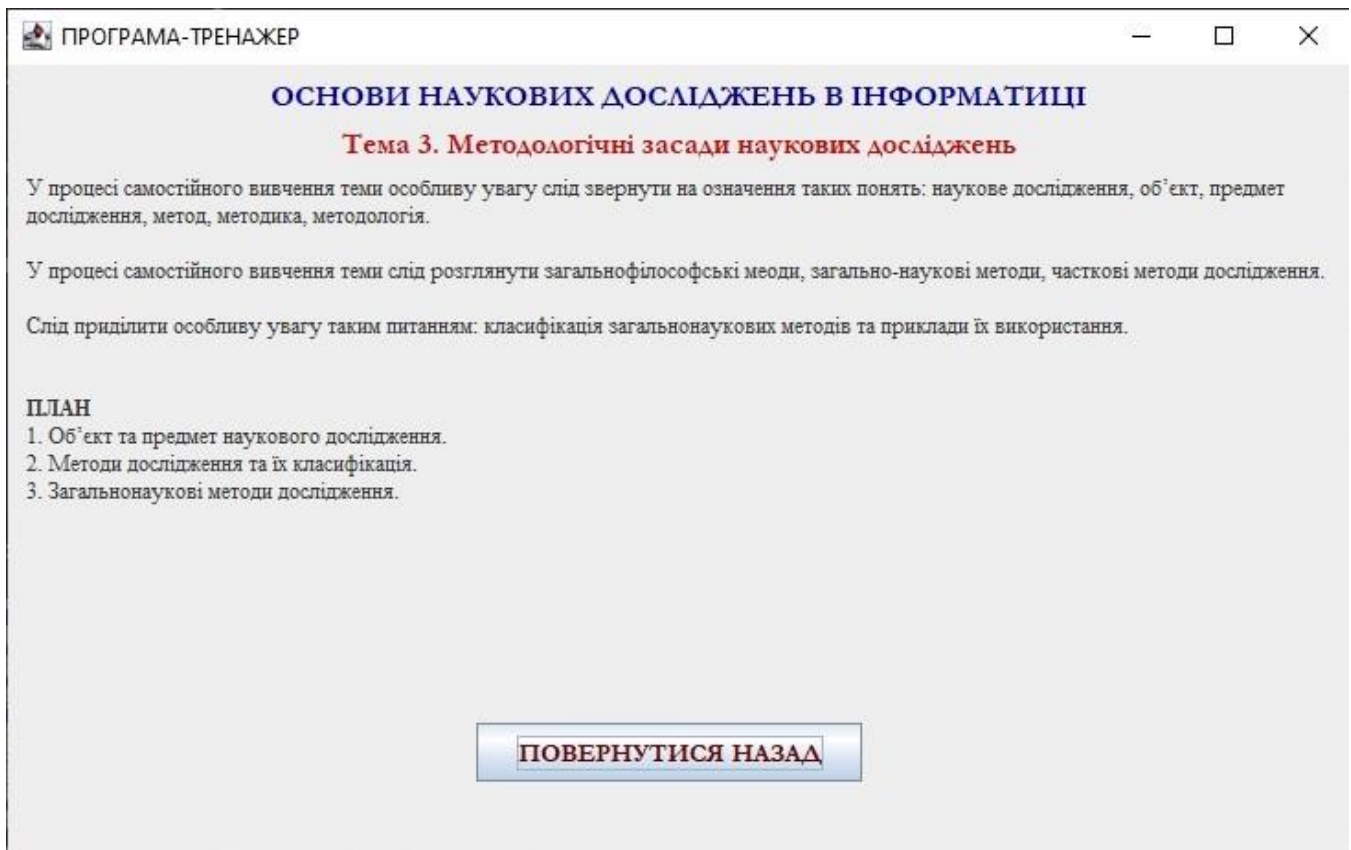


Рисунок 4.12 – Вікно ознайомлення з методичними рекомендаціями по Темі 3

***Частина коду вікна з методичними рекомендаціями по Темі 3:***

...

```
JLabel lblNewLabel_1 = new JLabel("Тема 3. Методологічні засади наукових  
досліджень");
```

```
lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
```

```
lblNewLabel_1.setForeground(new Color(204, 0, 0));
```

```
lblNewLabel_1.setFont(new Font("Garamond", Font.BOLD, 16));
```

```
lblNewLabel_1.setBounds(0, 26, 731, 34);
```

```
frame.getContentPane().add(lblNewLabel_1);
```

...

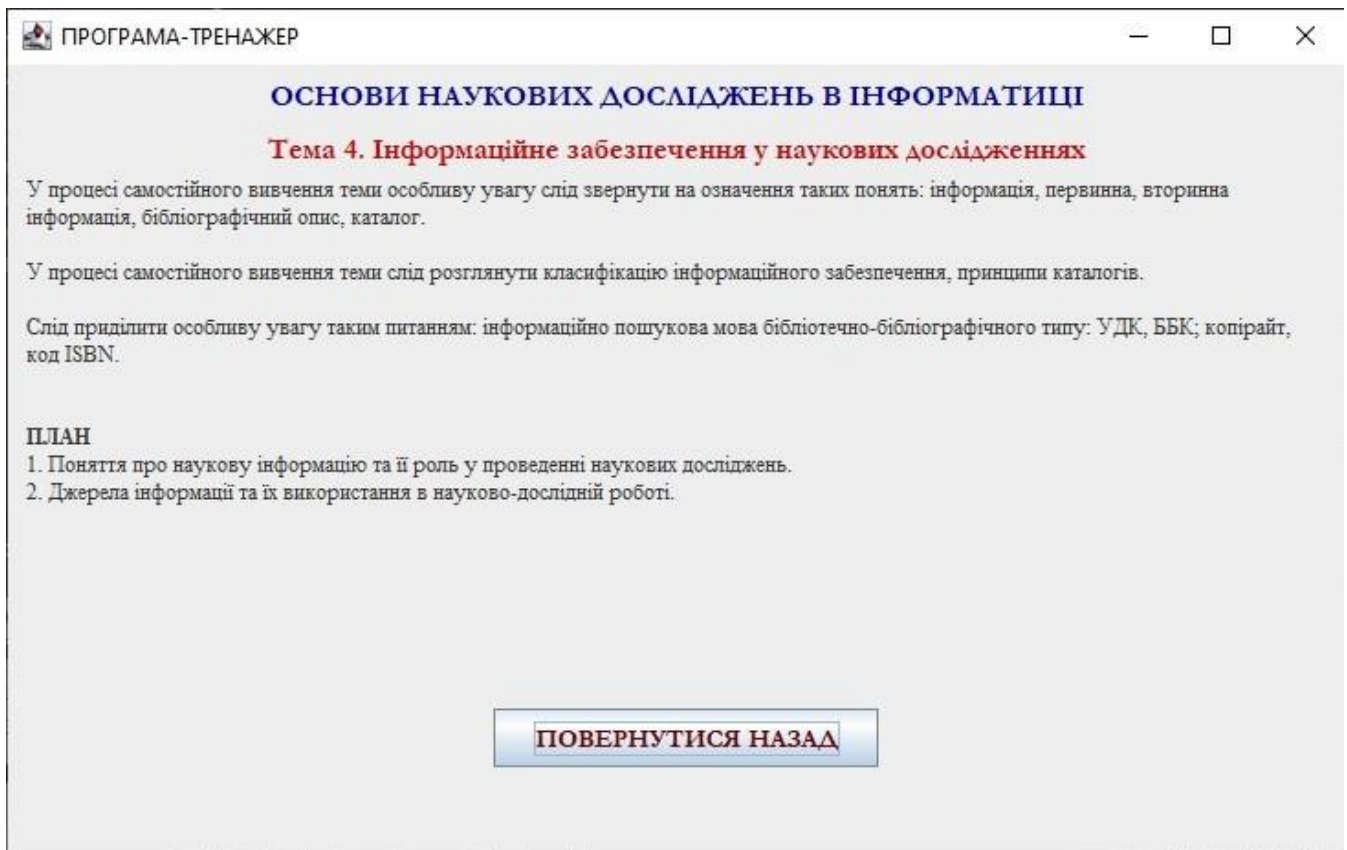


Рисунок 4.13 – Вікно ознайомлення з методичними рекомендаціями по Темі 4

***Частина коду вікна з методичними рекомендаціями по Темі 4:***

...

```
lblNewLabel.setForeground(new Color(0, 0, 153));
```

```
lblNewLabel.setFont(new Font("Garamond", Font.BOLD, 16));
```

```
lblNewLabel.setBounds(0, 0, 731, 34);
```

```
frame.getContentPane().add(lblNewLabel);
```

```
JLabel lblNewLabel_1 = new JLabel("Тема 4. Інформаційне забезпечення у  
наукових дослідженнях");
```

```
lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
```

```
lblNewLabel_1.setForeground(new Color(204, 0, 0));
```

```
lblNewLabel_1.setFont(new Font("Garamond", Font.BOLD, 16));
```

...

Після ознайомлення з методичними рекомендаціями по вивченню 1 модуля дистанційного навального курсу з «Основ наукових досліджень в інформатиці»,

користувач програми може розпочати тестування, натиснувши кнопку ПОЧАТИ ТЕСТУВАННЯ.

Кожне питання виводиться в окремому вікні (рис. 4.14), а результат відповіді записується програмою. Якщо відповідь невірна, то з'явиться вікно з теоретичною довідкою по даному питанню модуля (рис. 4.15) і здобувач освіти може повторити матеріал і спробувати відповісти правильно. Кількість спроб відповіді не обмежується але програма фіксує їх і зафіксує в кінцевий результат.

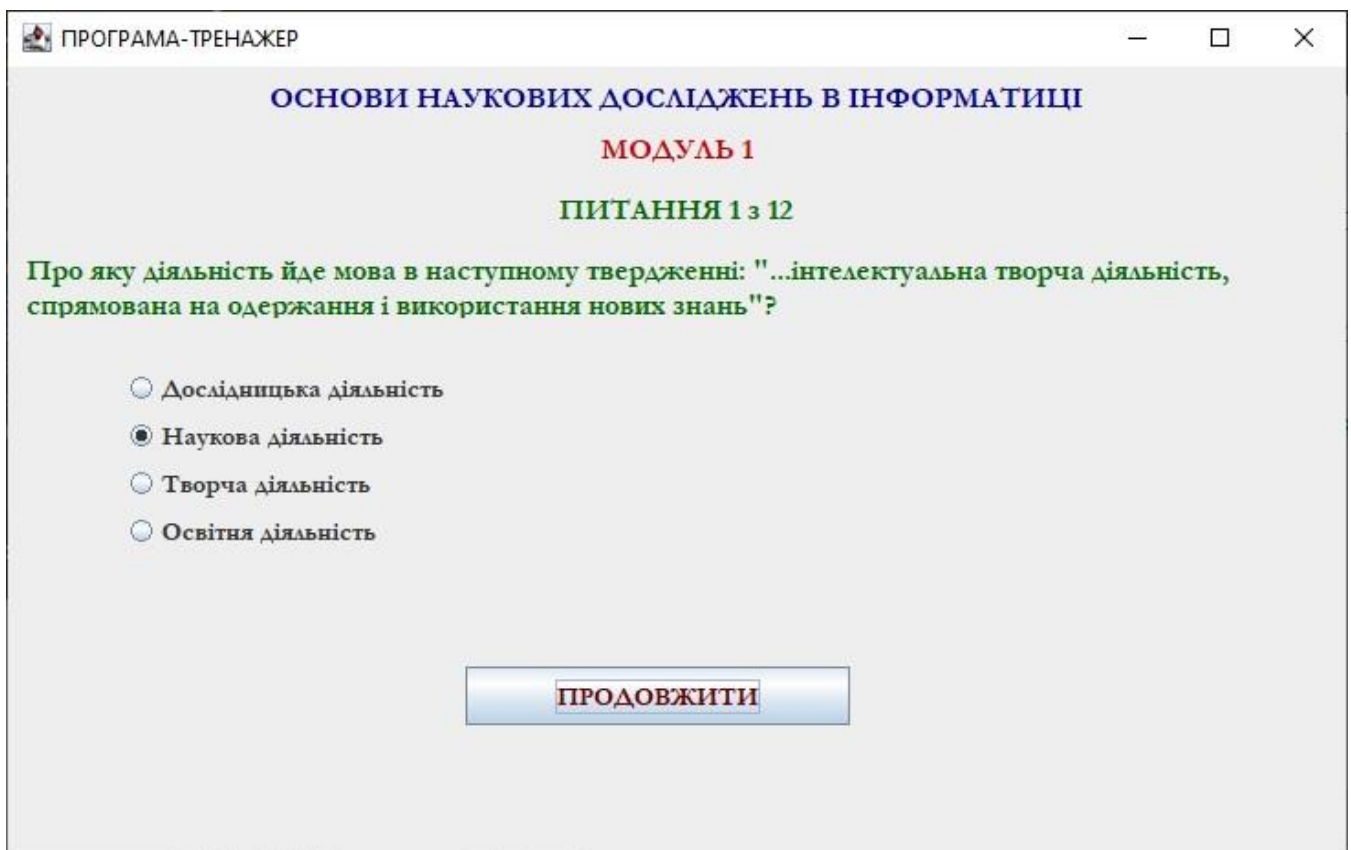


Рисунок 4.14 – Вікно з питанням тесту

***Частина коду вікна з питанням тесту:***

...

```
JLabel lblNewLabel_1_1_1 = new JLabel("<html><p>Про яку діяльність йде  
мова в наступному твердженні: \"...інтелектуальна творча діяльність, спрямована  
на одержання і використання нових знань\"?</p></html>");
```

```
lblNewLabel_1_1_1.setHorizontalAlignment(SwingConstants.LEFT);
```

...



Рисунок 4.15 – Вікно з теоретичною довідкою до питання

**Частина коду вікна з теоретичною довідкою:**

...

```
JLabel lblNewLabel_2 = new JLabel("Рис.1 Відповідність суб'єктів  
результату");
```

```
lblNewLabel_2.setFont(new Font("Garamond", Font.PLAIN, 12));
```

```
lblNewLabel_2.setIcon(new
```

```
ImageIcon("C:\\Users\\ВИКЛАДАЧ\\Desktop\\k1h1.jpg"));
```

```
lblNewLabel_2.setBounds(20, 188, 669, 189);
```

```
frame.getContentPane().add(lblNewLabel_2);
```

...

Починаючи з 9 питання, здобувачу освіти потрібно не обирати варіант відповіді, а вводити її самостійно (рис. 4.16). Так само як і при виборі варіанту якщо відповідь не вірна, то з'являється теоретична довідка до даної теми.

Рисунок 4.16 – Вікно з питанням і полем для введення відповіді

***Частина коду вікна з питанням тесту:***

...

JLabel lblNewLabel\_2 = **new** JLabel("Дайте письмову відповідь на запитання.  
Відповідь потрібно вводити використовуючи лише нижній регістр літер.");

lblNewLabel\_2.setHorizontalAlignment(SwingConstants.**CENTER**);

lblNewLabel\_2.setFont(**new** Font("Garamond", Font.**PLAIN**, 14));

lblNewLabel\_2.setBounds(10, 92, 711, 32);

frame.getContentPane().add(lblNewLabel\_2);

JLabel lblNewLabel\_2\_1 = **new** JLabel("ВАША ВІДПОВІДЬ:");

lblNewLabel\_2\_1.setHorizontalAlignment(SwingConstants.**CENTER**);

lblNewLabel\_2\_1.setFont(**new** Font("Garamond", Font.**PLAIN**, 14));

lblNewLabel\_2\_1.setBounds(70, 250, 238, 32);

frame.getContentPane().add(lblNewLabel\_2\_1);

...

## ВИСНОВКИ

Таким чином мету кваліфікаційної роботи було досягнуто в повній мірі. Згідно завдання було створено програмне забезпечення тренажеру за 1 модулем дистанційного навчального курсу «Основи наукових досліджень в інформатиці». Створення відбувалося на об'єктно-орієнтованій мові програмування Java з використанням інтегрованого середовища програмування Eclipse останньої версії 2022-12.

В першому розділі кваліфікаційної роботи було визначено завдання, етапи виконання та алгоритм роботи програми-тренажеру.

В другому розділі викладено інформаційну довідку про поняття програми-тренажеру та актуальність використання такого програмного забезпечення під час дистанційного навчання.

Третій розділ містить теоретичний опис методології створення програмного забезпечення, інформаційне наповнення програми-тренажеру, блок-схему її роботи, теоретичні основи мови програмування Java та інтегрованого середовища програмування Eclipse.

В останньому четвертому розділі викладено практичну реалізацію кваліфікаційної роботи у вигляді віконної програми, що побудована за допомогою вікон, та Java-код, за допомогою якого ці вікна запрограмовані.

Отже, програма-тренажер орієнтована на те, щоб здобувач освіти покращив свої знання з дистанційного навчального курсу «Основи наукових досліджень в інформатиці» за модулем 1. Користувач може неодноразово виконувати тестові завдання та читати теоретичний матеріал, а викладач, в свою чергу, може оцінити роботу учня отримавши результати його роботи з програмою.

## ВИКОРИСТАНІ ДЖЕРЕЛА

1. Інформаційні технології і засоби навчання 3б. наук. пр. Київ: Атіка, 2005. 272 с.
2. Колчук Т. В. Принципи розробки навчальних матеріалів дистанційного курсу. *Theory and methods of e-learning*. 2014. Т. 2. С. 291–296. URL: <https://doi.org/10.55056/e-learn.v2i1.288> (дата звернення: 12.08.2022).
3. Яновський А. Інформаційно-освітнє середовище в умовах дистанційного навчання. *Humanities science current issues*. 2020. Т. 4, № 30. С. 310–315. URL: <https://doi.org/10.24919/2308-4863.4/30.212627> (дата звернення: 19.10.2022).
5. Доценко Н. Застосування навчальних комп'ютерних інтерактивних тренажерів здобувачами вищої освіти інженерних спеціальностей в умовах інформаційно-освітнього середовища. *Педагогічні науки: теорія, історія, інноваційні технології*. 2018. № 2(76). С. 118–128. URL: <https://doi.org/10.24139/2312-5993/2018.02/118-128> (дата звернення: 18.05.2022).
6. Учасники проектів Вікімедіа. Java – вікіпедія. Вікіпедія. URL: <https://uk.wikipedia.org/wiki/Java#Література> (дата звернення: 10.11.2022).
7. Учасники проектів Вікімедіа. Eclipse – вікіпедія. *Вікіпедія*. URL: <https://uk.wikipedia.org/wiki/Eclipse> (дата звернення: 10.11.2022).
8. Учасники проектів Вікімедіа. Освоюємо java – вікіпідручник. *Вікіпідручник*. URL: [https://uk.wikibooks.org/wiki/Освоюємо\\_Java](https://uk.wikibooks.org/wiki/Освоюємо_Java) (дата звернення: 24.11.2022).
9. Копитко М. Ф., Іванків К. С. Основи програмування мовою Java: тексти лекцій. Львів: Вид. центр ЛНУ ім. Ів. Франка, 2002. 83 с.
10. Методологія та організація наукових досліджень: конспект лекцій для студ. спец. 8.050201 "Менеджмент організацій" усіх форм навчання. Суми: СумДУ, 2009. 111 с.
11. Головний центр дистанційного навчання. URL: <http://www2.el.puet.edu.ua/zo/course/view.php?id=1449> (дата звернення: 10.01.2023).

12. Романчиков В. І. Основи наукових досліджень. Навчальний посібник / В. І. Романчиков. – К.: Центр учбової літератури, 2007. – 254 с.
13. Burnette E. Eclipse IDE: Pocket Guide. O'Reilly Media, 2005.
14. Шавтикова Л. М., Левченко А. В., Байрамкулов И. Х. Линейное программирование на языке Java. *Тенденции развития науки и образования*. 2021. Т. 80, № 3. С. 9–11. URL: <https://doi.org/10.18411/trnio-12-2021-109> (дата звернення: 09.01.2023).
15. Вовдюк Л. В. Дистанційне навчання. *Безпека життєдіяльності*. 2018. № 3, берез. С. 16–17.

## ДОДАТОК А. ПРОГРАМА

### КОД ПЕРШОГО ВІКНА ДЛЯ РЕЄСТРАЦІЇ

```
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import javax.swing.JOptionPane;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import java.awt.Color;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.SwingConstants;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.event.ChangeListener;
import javax.swing.event.ChangeEvent;
public class Register extends JDialog {
    private final JPanel contentPanel = new JPanel();
    private JTextField txtimya;
    private JTextField txtprizv;
    private JButton okButton;
    public static void main(String[] args) {
        try {
            Register dialog = new Register();
```

```

dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
dialog.setVisible(true);
} catch (Exception e) {
e.printStackTrace();
}
}

public Register() {
setBackground(new Color(255, 204, 204));
setTitle("ПРОГРАМА-ПРЕНАЖЕР");
setBounds(100, 100, 450, 300);
getContentPane().setLayout(new BorderLayout());
contentPanel.setBorder(new EmptyBorder(5, 5, 5, 5));
getContentPane().add(contentPanel, BorderLayout.CENTER);
contentPanel.setLayout(null);

JLabel lblNewLabel = new JLabel("ОСНОВИ НАУКОВИХ ДОСЛІДЖЕНЬ В
ІНФОРМАТИЦІ");
lblNewLabel.setForeground(new Color(0, 0, 204));
lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel.setFont(new Font("Garamond", Font.BOLD, 14));
lblNewLabel.setBounds(10, 11, 414, 26);
contentPanel.add(lblNewLabel);

JLabel lblNewLabel_1 = new JLabel("Введіть своє ім'я:");
lblNewLabel_1.setHorizontalAlignment(SwingConstants.LEFT);
lblNewLabel_1.setForeground(new Color(0, 102, 0));
lblNewLabel_1.setFont(new Font("Garamond", Font.BOLD, 14));
lblNewLabel_1.setBounds(10, 76, 146, 26);
contentPanel.add(lblNewLabel_1);

JLabel lblNewLabel_1_1 = new JLabel("Введіть своє прізвище:");

```

```

lblNewLabel_1_1.setHorizontalAlignment(SwingConstants.LEFT);
lblNewLabel_1_1.setForeground(new Color(0, 102, 0));
lblNewLabel_1_1.setFont(new Font("Garamond", Font.BOLD, 14));
lblNewLabel_1_1.setBounds(10, 113, 160, 26);
contentPanel.add(lblNewLabel_1_1);
txtimya = new JTextField();
txtimya.setHorizontalAlignment(SwingConstants.LEFT);
txtimya.setFont(new Font("Garamond", Font.BOLD, 14));
txtimya.setBounds(194, 80, 160, 20);
contentPanel.add(txtimya);
txtimya.setColumns(10);
txtprizv = new JTextField();
txtprizv.setHorizontalAlignment(SwingConstants.LEFT);
txtprizv.setFont(new Font("Garamond", Font.BOLD, 14));
txtprizv.setColumns(10);
txtprizv.setBounds(194, 117, 160, 20);
contentPanel.add(txtprizv);
JLabel lblNewLabel_1_1_1 = new JLabel("Щоб продовжити, натисніть кнопку  
\"BXID\");
lblNewLabel_1_1_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_1_1_1.setForeground(new Color(153, 0, 0));
lblNewLabel_1_1_1.setFont(new Font("Garamond", Font.BOLD, 14));
lblNewLabel_1_1_1.setBounds(10, 171, 414, 26);
contentPanel.add(lblNewLabel_1_1_1);
{
JPanel buttonPane = new JPanel();
buttonPane.setLayout(new FlowLayout(FlowLayout.RIGHT));
getContentPane().add(buttonPane, BorderLayout.SOUTH);

```

```

{
JPanel panel = new JPanel();
buttonPane.add(panel);
okButton = new JButton("ВХІД");
panel.add(okButton);
okButton.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
String imya, prizv;
imya = txtimya.getText();
prizv = txtprizv.getText();
System.exit (0);
}
});
okButton.setActionCommand("ВХІД");
getRootPane().setDefaultButton(okButton);
}
{
JButton cancelButton = new JButton("СКАСУВАТИ");
cancelButton.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
txtimya.setText(null);
txtprizv.setText(null);
}
});
cancelButton.setActionCommand("СКАСУВАТИ");
buttonPane.add(cancelButton);
}

```

```

}
}
protected JTextField JTextField(String string) {
return null;
}
public JButton getOkButton() {
return okButton;
}
}

```

## ДРУГЕ ВІКНО ПОЧАТОК

```

import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.SwingConstants;
import java.awt.Color;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.AbstractAction;
import javax.swing.Action;
import javax.swing.JPanel;
public class Pochatok {
private JFrame frame;
private final Action action = new SwingAction();
public static void main(String[] args) {

```

```

EventQueue.invokeLater(new Runnable() {
    public void run() {
        try {
            Pochatok window = new Pochatok();
            window.frame.setVisible(true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});
}

public Pochatok() {
    initialize();
}

private void initialize() {
    frame = new JFrame();
    frame.setTitle("ПРОГРАМА-ПРЕХАЖЕР");
    frame.setBounds(100, 100, 747, 470);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    JLabel lblNewLabel = new JLabel("ОСНОВИ НАУКОВИХ ДОСЛІДЖЕНЬ В  
ІНФОРМАТИЦІ");
    lblNewLabel.setForeground(new Color(0, 0, 153));
    lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
    lblNewLabel.setFont(new Font("Garamond", Font.BOLD, 16));
    lblNewLabel.setBounds(10, 11, 711, 32);
    frame.getContentPane().add(lblNewLabel);

    JLabel lblNewLabel_1 = new JLabel("МОДУЛЬ 1");

```

```

lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_1.setForeground(new Color(204, 0, 0));
lblNewLabel_1.setFont(new Font("Garamond", Font.BOLD, 16));
lblNewLabel_1.setBounds(10, 54, 711, 32);
frame.getContentPane().add(lblNewLabel_1);

JLabel lblNewLabel_1_1 = new JLabel("Тема 1. Організація науково-дослідної
роботи у вищому навчальному закладі");
lblNewLabel_1_1.setHorizontalAlignment(SwingConstants.LEFT);
lblNewLabel_1_1.setForeground(new Color(0, 102, 0));
lblNewLabel_1_1.setFont(new Font("Garamond", Font.BOLD, 16));
lblNewLabel_1_1.setBounds(10, 97, 711, 32);
frame.getContentPane().add(lblNewLabel_1_1);

JLabel lblNewLabel_1_1_1 = new JLabel("Тема 2. Поняття науки та її нормативне
регулювання");
lblNewLabel_1_1_1.setHorizontalAlignment(SwingConstants.LEFT);
lblNewLabel_1_1_1.setForeground(new Color(0, 102, 0));
lblNewLabel_1_1_1.setFont(new Font("Garamond", Font.BOLD, 16));
lblNewLabel_1_1_1.setBounds(10, 140, 711, 32);
frame.getContentPane().add(lblNewLabel_1_1_1);

JLabel lblNewLabel_1_1_1_1 = new JLabel("Тема 3. Методологічні засади наукових
досліджень");
lblNewLabel_1_1_1_1.setHorizontalAlignment(SwingConstants.LEFT);
lblNewLabel_1_1_1_1.setForeground(new Color(0, 102, 0));
lblNewLabel_1_1_1_1.setFont(new Font("Garamond", Font.BOLD, 16));
lblNewLabel_1_1_1_1.setBounds(10, 183, 711, 32);
frame.getContentPane().add(lblNewLabel_1_1_1_1);

JLabel lblNewLabel_1_1_1_1_1 = new JLabel("Тема 4. Інформаційне забезпечення у
наукових дослідженнях");
lblNewLabel_1_1_1_1_1.setHorizontalAlignment(SwingConstants.LEFT);

```

```

lblNewLabel_1_1_1_1_1.setForeground(new Color(0, 102, 0));
lblNewLabel_1_1_1_1_1.setFont(new Font("Garamond", Font.BOLD, 16));
lblNewLabel_1_1_1_1_1.setBounds(10, 226, 711, 32);
frame.getContentPane().add(lblNewLabel_1_1_1_1_1);
JButton btnNewButton_1 = new JButton("ТЕМА 2");
btnNewButton_1.setForeground(new Color(255, 51, 0));
btnNewButton_1.setFont(new Font("Garamond", Font.BOLD, 14));
btnNewButton_1.setBounds(238, 323, 116, 32);
frame.getContentPane().add(btnNewButton_1);
JButton btnNewButton_2 = new JButton("ТЕМА 3");
btnNewButton_2.setForeground(new Color(0, 51, 255));
btnNewButton_2.setFont(new Font("Garamond", Font.BOLD, 14));
btnNewButton_2.setBounds(407, 323, 116, 32);
frame.getContentPane().add(btnNewButton_2);
JButton btnNewButton_2_1 = new JButton("ТЕМА 4");
btnNewButton_2_1.setForeground(new Color(0, 51, 0));
btnNewButton_2_1.setFont(new Font("Garamond", Font.BOLD, 14));
btnNewButton_2_1.setBounds(573, 323, 116, 32);
frame.getContentPane().add(btnNewButton_2_1);
JLabel lblNewLabel_1_1_1_2 = new JLabel("ТЕОРЕТИЧНА ДОБІДКА");
lblNewLabel_1_1_1_2.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_1_1_1_2.setForeground(new Color(153, 0, 0));
lblNewLabel_1_1_1_2.setFont(new Font("Garamond", Font.BOLD, 14));
lblNewLabel_1_1_1_2.setBounds(159, 286, 414, 26);
frame.getContentPane().add(lblNewLabel_1_1_1_2);
JButton btnNewButton_1_1 = new JButton("ПОЧАТИ ТЕСТУВАННЯ");
btnNewButton_1_1.addActionListener(new ActionListener() {

```

```

public void actionPerformed(ActionEvent e) {
}

});

btnNewButton_1_1.setForeground(new Color(102, 0, 0));
btnNewButton_1_1.setFont(new Font("Garamond", Font.BOLD, 14));
btnNewButton_1_1.setBounds(258, 388, 210, 32);
frame.getContentPane().add(btnNewButton_1_1);
JButton btnNewButton_1_2 = new JButton("ТЕМА 1");
btnNewButton_1_2.setForeground(new Color(186, 85, 211));
btnNewButton_1_2.setFont(new Font("Garamond", Font.BOLD, 14));
btnNewButton_1_2.setBounds(67, 323, 116, 32);
frame.getContentPane().add(btnNewButton_1_2);
}
}

```

## ОПИС ТЕМИ 1

```

import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JLabel;
import java.awt.Font;
import java.awt.Color;
import javax.swing.SwingConstants;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import javax.swing.JButton;
public class Tema1 {
private JFrame frame;

```

```

public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                Tema1 window = new Tema1();
                window.frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

public Tema1() {
    initialize();
}

private void initialize() {
    frame = new JFrame();
    frame.setTitle("ПРОГРАМА-ТЕМНАЖЕР");
    frame.setBounds(100, 100, 747, 470);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    JLabel lblNewLabel = new JLabel("ОСНОВИ НАУКОВИХ ДОСЛІДЖЕНЬ В  
ІНФОРМАТИЦІ");
    lblNewLabel.setBounds(0, 1, 731, 34);
    lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
    lblNewLabel.setForeground(new Color(0, 0, 153));
    lblNewLabel.setFont(new Font("Garamond", Font.BOLD, 16));
    frame.getContentPane().add(lblNewLabel);
}

```

```

JLabel lblNewLabel_1 = new JLabel("Тема 1. Організація науково-дослідної роботи
у вищому навчальному закладі");

lblNewLabel_1.setBounds(0, 35, 731, 34);

lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);

lblNewLabel_1.setForeground(new Color(204, 0, 0));

lblNewLabel_1.setFont(new Font("Garamond", Font.BOLD, 16));

frame.getContentPane().add(lblNewLabel_1);

JLabel lblNewLabel_2 = new JLabel("<html> <p>У процесі самостійного вивчення
теми особливу увагу слід звернути на означення таких понять: науково-дослідна
робота (НДР), об'єкт, <br>суб'єкт наукової діяльності.<br> <br> У процесі
самостійного вивчення теми слід розглянути особливості сучасної науки, види і
форми НДР.<br> <br> Слід приділити особливу увагу таким питанням: організація
НДР у ВНЗ.</p><br><br><b>ПЛАН</b><br><p>1. Історичні етапи становлення та
розвитку науки.<br>\r\n2. Особливості сучасної науки.<br>\r\n3. Суб'єкти наукової
роботи і діяльності у вищому навчальному закладі.<br>\r\n4. Організація вузівської
науки.<br>\r\n5. Види і форми науково-дослідної роботи.<br>\r\n6. Система
управління науково-дослідною роботою.<br>\r\n7. Науково-дослідна робота
студентів</p></html>");

lblNewLabel_2.setLabelFor(lblNewLabel_2);

lblNewLabel_2.setBounds(10, 68, 711, 249);

lblNewLabel_2.setFont(new Font("Times New Roman", Font.PLAIN, 12));

lblNewLabel_2.setVerticalAlignment(SwingConstants.TOP);

frame.getContentPane().add(lblNewLabel_2);

JButton btnNewButton_1_1 = new JButton("ПОВЕРНУТИСЯ НАЗАД");

btnNewButton_1_1.setForeground(new Color(102, 0, 0));

btnNewButton_1_1.setFont(new Font("Garamond", Font.BOLD, 14));

btnNewButton_1_1.setBounds(263, 352, 210, 32);

frame.getContentPane().add(btnNewButton_1_1);

}

}

```

```
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JLabel;
import java.awt.Font;
import java.awt.Color;
import javax.swing.SwingConstants;
import javax.swing.JButton;
public class Tema2 {
    private JFrame frame;
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Tema2 window = new Tema2();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
    public Tema2() {
        initialize();
    }
    private void initialize() {
        frame = new JFrame();
        frame.setTitle("ПРОГРАМА-ПРЕХАЖЕР");
```

```

frame.setBounds(100, 100, 747, 470);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.getContentPane().setLayout(null);

JLabel lblNewLabel = new JLabel("ОСНОВИ НАУКОВИХ ДОСЛІДЖЕНЬ В  
ІНФОРМАТИЦІ");
lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel.setForeground(new Color(0, 0, 153));
lblNewLabel.setFont(new Font("Garamond", Font.BOLD, 16));
lblNewLabel.setBounds(0, 0, 731, 34);
frame.getContentPane().add(lblNewLabel);

JLabel lblNewLabel_1 = new JLabel("Тема 2. Поняття науки та її нормативне  
регулювання");
lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_1.setForeground(new Color(204, 0, 0));
lblNewLabel_1.setFont(new Font("Garamond", Font.BOLD, 16));
lblNewLabel_1.setBounds(0, 32, 731, 34);
frame.getContentPane().add(lblNewLabel_1);

JLabel lblNewLabel_2 = new JLabel("<html> <p>У процесі самостійного вивчення  
теми особливу увагу слід звернути на означення таких понять: наука, наукова  
діяльність, глобалізація науки, інтеграція науки.<br> <br> У процесі самостійного  
вивчення теми слід розглянути загальну класифікацію наук, форми наукового  
міжнародного співробітництва.<br> <br> Слід приділити особливу увагу таким  
питанням: функції науки, пріоритетні напрямки розвитку науки в  
Україні.</p><br><br><b>ПЛАН</b><br><p>1. Поняття науки та її  
функцій.<br>\r\n2. Наукова діяльність та глобалізація науки.<br>\r\n3. Структура і  
класифікація науки.<br>\r\n4. Законодавчо-нормативне регулювання науки в  
Україні.<br>\r\n5. Пріоритетні напрями розвитку науки в Україні.</p></html>");
lblNewLabel_2.setVerticalAlignment(SwingConstants.TOP);
lblNewLabel_2.setFont(new Font("Times New Roman", Font.PLAIN, 12));
lblNewLabel_2.setBounds(10, 66, 711, 249);
frame.getContentPane().add(lblNewLabel_2);

JButton btnNewButton_1_1 = new JButton("ПОВЕРНУТИСЯ НАЗАД");

```

```

btnNewButton_1_1.setForeground(new Color(102, 0, 0));
btnNewButton_1_1.setFont(new Font("Garamond", Font.BOLD, 14));
btnNewButton_1_1.setBounds(263, 348, 210, 32);
frame.getContentPane().add(btnNewButton_1_1);
}
}

```

### ОПИС ТЕМИ 3

```

import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JLabel;
import java.awt.Font;
import java.awt.Color;
import javax.swing.SwingConstants;
import javax.swing.JButton;
public class Tema3 {
    private JFrame frame;
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Tema3 window = new Tema3();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}

```

```

});
}

public Tema3() {
    initialize();
}

private void initialize() {
    frame = new JFrame();
    frame.setTitle("ПРОГРАМА-ПРЕНАЖЕР");
    frame.setBounds(100, 100, 747, 470);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    JLabel lblNewLabel = new JLabel("ОСНОВИ НАУКОВИХ ДОСЛІДЖЕНЬ В  
ІНФОРМАТИЦІ");
    lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
    lblNewLabel.setForeground(new Color(0, 0, 153));
    lblNewLabel.setFont(new Font("Garamond", Font.BOLD, 16));
    lblNewLabel.setBounds(0, 0, 731, 34);
    frame.getContentPane().add(lblNewLabel);

    JLabel lblNewLabel_1 = new JLabel("Тема 3. Методологічні засади наукових  
досліджень");
    lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
    lblNewLabel_1.setForeground(new Color(204, 0, 0));
    lblNewLabel_1.setFont(new Font("Garamond", Font.BOLD, 16));
    lblNewLabel_1.setBounds(0, 26, 731, 34);
    frame.getContentPane().add(lblNewLabel_1);

    JLabel lblNewLabel_2 = new JLabel("<html> <p>У процесі самостійного вивчення  
теми особливу увагу слід звернути на означення таких понять: наукове  
дослідження, об'єкт, предмет дослідження, метод, методика, методологія.<br> <br>  
У процесі самостійного вивчення теми слід розглянути загальнофілософські меоди,  
загально-наукові методи, часткові методи дослідження.<br> <br> Слід приділити

```

особливу увагу таким питанням: класифікація загальнонаукових методів та приклади їх використання.

**ПЛАН**

1. Об'єкт та предмет наукового дослідження.

2. Методи дослідження та їх класифікація.

3. Загальнонаукові методи дослідження.

```
lblNewLabel_2.setVerticalAlignment(SwingConstants.TOP);
lblNewLabel_2.setFont(new Font("Times New Roman", Font.PLAIN, 12));
lblNewLabel_2.setBounds(10, 59, 711, 249);
frame.getContentPane().add(lblNewLabel_2);
JButton btnNewButton_1_1 = new JButton("ПОБЕРНУТИСЯ НАЗАД");
btnNewButton_1_1.setForeground(new Color(102, 0, 0));
btnNewButton_1_1.setFont(new Font("Garamond", Font.BOLD, 14));
btnNewButton_1_1.setBounds(255, 358, 210, 32);
frame.getContentPane().add(btnNewButton_1_1);
}
}
```

#### ОПИС ТЕМИ 4

```
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JLabel;
import java.awt.Font;
import java.awt.Color;
import javax.swing.SwingConstants;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
public class Tema4 {
    private JFrame frame;
    public static void main(String[] args) {
```

```

EventQueue.invokeLater(new Runnable() {
    public void run() {
        try {
            Tema4 window = new Tema4();
            window.frame.setVisible(true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});
}

public Tema4() {
    initialize();
}

private void initialize() {
    frame = new JFrame();
    frame.setTitle("ПРОГРАМА-ПРЕХАЖЕР");
    frame.setBounds(100, 100, 747, 470);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);

    JLabel lblNewLabel = new JLabel("ОСНОВИ НАУКОВИХ ДОСЛІДЖЕНЬ В  
ІНФОРМАТИЦІ");
    lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
    lblNewLabel.setForeground(new Color(0, 0, 153));
    lblNewLabel.setFont(new Font("Garamond", Font.BOLD, 16));
    lblNewLabel.setBounds(0, 0, 731, 34);
    frame.getContentPane().add(lblNewLabel);

```

```

JLabel lblNewLabel_1 = new JLabel("Тема 4. Інформаційне забезпечення у наукових дослідженнях");

lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);

lblNewLabel_1.setForeground(new Color(204, 0, 0));

lblNewLabel_1.setFont(new Font("Garamond", Font.BOLD, 16));

lblNewLabel_1.setBounds(0, 29, 731, 34);

frame.getContentPane().add(lblNewLabel_1);

JLabel lblNewLabel_2 = new JLabel("<html> <p>У процесі самостійного вивчення теми особливу увагу слід звернути на означення таких понять: інформація, первинна, вторинна інформація, бібліографічний опис, каталог.<br> <br> У процесі самостійного вивчення теми слід розглянути класифікацію інформаційного забезпечення, принципи каталогів.<br> <br> Слід приділити особливу увагу таким питанням: інформаційно пошукова мова бібліотечно-бібліографічного типу: УДК, ББК; копірайт, код ISBN.</p><br><br><b>ПЛАН</b><br><p>\r\n1. Поняття про наукову інформацію та її роль у проведенні наукових досліджень.<br>\r\n2. Джерела інформації та їх використання в науково-дослідній роботі.</p></html>");

lblNewLabel_2.setVerticalAlignment(SwingConstants.TOP);

lblNewLabel_2.setFont(new Font("Times New Roman", Font.PLAIN, 12));

lblNewLabel_2.setBounds(10, 60, 711, 249);

frame.getContentPane().add(lblNewLabel_2);

JButton btnNewButton_1_1 = new JButton("ПОВЕРНУТИСЯ НАЗАД");

btnNewButton_1_1.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {

        new Pochatok();

        setVisible(false);

    }

    private void setVisible(boolean e) {

    }

});

btnNewButton_1_1.setForeground(new Color(102, 0, 0));

btnNewButton_1_1.setFont(new Font("Garamond", Font.BOLD, 14));

```

```

btnNewButton_1_1.setBounds(265, 351, 210, 32);
frame.getContentPane().add(btnNewButton_1_1);
}
}

```

## ПЕРШЕ ПИТАННЯ

```

import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.Insets;
import java.awt.Label;
import java.awt.event.ActionEvent;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JRadioButton;
import javax.swing.SwingConstants;
import java.awt.event.ActionListener;
public class Put1 {
    private JFrame frame;
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Put1 window = new Put1();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
    public Put1() {
        initialize();
    }
    private void initialize() {
        frame = new JFrame();
        frame.setTitle("ПРОГРАМА-ТРЕНАЖЕР");
        frame.setBounds(100, 100, 747, 470);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().setLayout(null);
        JLabel lblNewLabel = new JLabel("ОСНОВИ НАУКОВИХ ДОСЛІДЖЕНЬ В ІНФОРМАТИЦІ");
    }
}

```

```

lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel.setForeground(new Color(0, 0, 153));
lblNewLabel.setFont(new Font("Garamond", Font.BOLD, 16));
lblNewLabel.setBounds(0, 0, 731, 34);
frame.getContentPane().add(lblNewLabel);
JLabel lblNewLabel_1 = new JLabel("МОДУЛЬ 1");
lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_1.setForeground(new Color(204, 0, 0));
lblNewLabel_1.setFont(new Font("Garamond", Font.BOLD, 16));
lblNewLabel_1.setBounds(10, 29, 711, 32);
frame.getContentPane().add(lblNewLabel_1);
JLabel lblNewLabel_1_1 = new JLabel("ПИТАННЯ 1 з 12");
lblNewLabel_1_1.setHorizontalAlignment(SwingConstants.CENTER);
lblNewLabel_1_1.setForeground(new Color(0, 102, 0));
lblNewLabel_1_1.setFont(new Font("Garamond", Font.BOLD, 16));
lblNewLabel_1_1.setBounds(10, 62, 711, 32);
frame.getContentPane().add(lblNewLabel_1_1);
JLabel lblNewLabel_1_1_1 = new JLabel("<html><p>Про яку діяльність йде мова в  
наступному твердженні: \"...інтелектуальна творча діяльність, спрямована на  
одержання і використання нових знань\"?</p></html>");
lblNewLabel_1_1_1.setHorizontalAlignment(SwingConstants.LEFT);
lblNewLabel_1_1_1.setForeground(new Color(0, 102, 0));
lblNewLabel_1_1_1.setFont(new Font("Garamond", Font.BOLD, 16));
lblNewLabel_1_1_1.setBounds(10, 105, 711, 32);
frame.getContentPane().add(lblNewLabel_1_1_1);
JRadioButton rdb1 = new JRadioButton("Дослідницька діяльність");
rdb1.setFont(new Font("Garamond", Font.BOLD, 14));
rdb1.setBounds(63, 164, 187, 23);
frame.getContentPane().add(rdb1);
JRadioButton rdb2 = new JRadioButton("Наукова діяльність");
rdb2.setFont(new Font("Garamond", Font.BOLD, 14));
rdb2.setBounds(63, 190, 187, 23);
frame.getContentPane().add(rdb2);
JRadioButton rdb3 = new JRadioButton("Творча діяльність");
rdb3.setFont(new Font("Garamond", Font.BOLD, 14));
rdb3.setBounds(63, 216, 187, 23);
frame.getContentPane().add(rdb3);
JRadioButton rdb4 = new JRadioButton("Освітня діяльність");
rdb4.setFont(new Font("Garamond", Font.BOLD, 14));
rdb4.setBounds(63, 242, 187, 23);
frame.getContentPane().add(rdb4);
JButton btNext = new JButton("ПРОДОВЖИТИ");
btNext.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

```

```
});  
btNext.setForeground(new Color(102, 0, 0));  
btNext.setFont(new Font("Garamond", Font.BOLD, 14));  
btNext.setBounds(250, 327, 210, 32);  
frame.getContentPane().add(btNext);  
}}
```