

**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

**Розробка програмного забезпечення тренажеру з теми «Регулярні вирази:
синтаксис, сфера використання» дистанційного навчального курсу «Теорія
програмування»**

зі спеціальності 122 «Комп'ютерні науки»

Виконавець роботи Дашевський Василь Андрійович

_____ « ____ » _____ 2021р.
(підпис)

Науковий керівник к.ф.-м.н., доц. Черненко Оксана Олексіївна

_____ « ____ » _____ 2021р.
(підпис)

ПОЛТАВА 2021 р.

ЗМІСТ

ВСТУП	3
1. ПОСТАНОВКА ЗАДАЧІ.....	4
2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	6
2.1. Рекомендації щодо організації дистанційного навчання.....	6
2.2. Інструменти спілкування у дистанційному навчанні.....	8
3. ТЕОРЕТИЧНА ЧАСТИНА	12
3.1. Регулярні вирази: синтаксис, сфера використання	12
3.2. Алгоритмізація задачі за темою роботи	17
4. ПРАКТИЧНА ЧАСТИНА	25
4.1. Обґрунтування вибору програмних засобів для реалізації завдання роботи	25
4.2. Опис роботи програми.....	27
ВИСНОВКИ.....	31
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	33
ДОДАТОК А.....	34

ВСТУП

Розвиток сучасного суспільства характеризується процесом інформатизації і, у першу чергу, інформатизації освіти. У цих умовах наука й освіта повинні постійно вдосконалюватись. Найбільш перспективним напрямком інтенсифікації навчального процесу у вищих навчальних закладах є його комп'ютеризація.

Динамізм розвитку обчислювальної техніки, технологій програмування і телекомунікацій визначили значний прорив у сфері розробки різних комп'ютерних систем навчання, у тому числі комп'ютерних тренажерів, та їх застосування у підготовці студентів.

Метою роботи є розробка тренажеру з теми «Регулярні вирази: синтаксис, сфера використання» дистанційного навчального курсу «Теорія програмування».

Об'єктом розробки є процес дистанційного навчання математичним дисциплінам.

Предметом розробки є алгоритм роботи тренажеру з теми «Регулярні вирази: синтаксис, сфера використання».

Перелік використаних методів полягає в застосуванні регулярних виразів.

Тренажер готовий до використання в дистанційному курсі «Теорія програмування».

Робота складається з чотирьох розділів.

Обсяг пояснювальної записки: 34 стор., в т.ч. основна частина - 31 стор., джерела - 9 назв.

1. ПОСТАНОВКА ЗАДАЧІ

Визначимо основні закономірності розвитку комп'ютерних тренажерів:

- їх розвиток має відбуватися в тісному взаємозв'язку з усіма елементами системи навчальної (професійної) підготовки;
- існує першочерговий вплив законів і принципів педагогіки та психології навчання на створення і використання перспективних зразків;
- дидактичні та технічні вимоги до перспективних зразків повинні відповідати завданням навчання;
- рівень досконалості залежить від використання нових інформаційних технологій;

Основний принцип, що повинен бути покладений в основу створення перспективних комп'ютерних тренажерів – принцип відповідності засобів навчання змісту навчання і дидактичним завданням. Іншими принципами, які складають основу створення перспективних комп'ютерних тренажерів, є:

- принцип раціональності – раціонального розподілу завдань навчання за створюваними засобами навчання, тобто складність (не завжди технічна) розроблюваних засобів повинна відповідати значущості вирішуваних на них завдань;
- принцип узгодженості – усі засоби навчання повинні бути пов'язані між собою загальними й єдиними для цієї системи нормами оцінок, тобто перехід учня з одного технічного засобу навчання на інший повинен здійснюватися лише на основі оцінки його діяльності на попередньому засобі (засобах);
- принцип достатності – кожний технічний засіб системи має вирішувати певне коло завдань навчання, що не дублюють завдання, які вирішуються на інших засобах;

- принцип безперервності – кількість і технічні характеристики засобів, що складають систему засобів навчання, повинні вибиратися таким чином, щоб забезпечити безперервний процес навчання при повному завантаженні засобів, що його реалізують;

Кожен тренажер являє собою комплекс операцій, здійснюваних за допомогою програмних засобах. Інтерактивний тренажер складається з трьох етапів:

1. *Ознайомлювальний етап.* Користувач ознайомлюється з тренажером, його темою. Надається можливість переглянути теорію перед початком.

2. *Практичне заняття* розробляється за сценарієм і містить в собі низку дій, які повинен виконати користувач, щоб вирішити поставлене завдання. На цьому етапі користувач повинен зрозуміти: що йому потрібно зробити, щоб вирішити поставлене завдання.

3. *Тестування.* У режимі тестування користувач уже без підказок повинен виконати необхідні дії, а система автоматично здійснить оцінювання результату цих дій.

Тренажер – це самостійний програмний модуль, призначений для навчання, відпрацювання отриманих навичок та їх подальшого контролю. Тренажер надає можливість знайти індивідуальний підхід до кожного слухача: час проходження заняття залежить від нього, його можливостей із засвоєння матеріалів.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Рекомендації щодо організації дистанційного навчання

Дистанційне навчання – форма організації і реалізації освітнього процесу, за якою його учасники (об'єкт і суб'єкт навчання) здійснюють навчальну взаємодію принципово і переважно екстериторіально (тобто, на відстані, яка не дозволяє і не передбачає безпосередню навчальну взаємодію учасників віч-на-віч, інакше, коли учасники територіально знаходяться поза межами можливої безпосередньої навчальної взаємодії і коли у процесі навчання їх особиста присутність у певних навчальних приміщеннях навчального закладу не є обов'язковою).

Дистанційне навчання реалізовується шляхом застосування дистанційної форми як окремої форми навчання або використання технологій дистанційного навчання для забезпечення навчання в різних формах.

Науково-методичне забезпечення дистанційного навчання включає:

- методичні (теоретичні та практичні) рекомендації щодо розроблення та використання педагогічно-психологічних та інформаційно-комунікаційних технологій дистанційного навчання;
- критерії, засоби і системи контролю якості дистанційного навчання;
- змістовне, дидактичне та методичне наповнення веб-ресурсів (дистанційних курсів) навчального плану/навчальної програми підготовки.

Для забезпечення дистанційного навчання, необхідні веб-ресурси навчальних дисциплін (програм), які можуть містити:

- методичні рекомендації щодо їх використання, послідовності виконання завдань, особливостей контролю тощо;

- документи планування навчального процесу (навчальні програми, навчально-тематичні плани, розклади занять);
- відео- та аудіозаписи уроків, лабораторних, практичних занять тощо;
- мультимедійні матеріали;
- термінологічні словники;
- практичні завдання із методичними рекомендаціями щодо їх виконання;
- віртуальні лабораторні роботи із методичними рекомендаціями щодо їх виконання;
- віртуальні тренажери із методичними рекомендаціями щодо їх використання;
- пакети тестових завдань для проведення контрольних заходів, тестування із автоматизованою перевіркою результатів, тестування із перевіркою викладачем;
- ігрові інтерактивні завдання із методичними рекомендаціями щодо їх використання;
- електронні бібліотеки чи посилання на них;
- бібліографії;
- дистанційний курс, що об'єднує зазначені вище веб-ресурси навчальної дисципліни (програми) єдиним педагогічним сценарієм;
- інші ресурси навчального призначення.

Перелік веб-ресурсів навчальних дисциплін (програм), необхідних для забезпечення дистанційного навчання, визначається навчальним закладом залежно від профілю навчальної дисципліни.

Для забезпечення дистанційного навчання учнів, вихованців, студентів, слухачів навчальний заклад може створювати власні веб-ресурси

або використовувати інші веб-ресурси, що підлягають перевірці у цьому навчальному закладі [2].

2.2. Інструменти спілкування у дистанційному навчанні

Електронна пошта – це стандартний сервіс Інтернету, що забезпечує передавання повідомлень, як у формі звичайних текстів, які в інших формах (програмах, графіці, звуках, відео) у відкритому чи зашифрованому вигляді. У системі освіти електронна пошта використовується для організації спілкування викладача й студента, а також студентів між собою.

Форум – найпоширеніша форма спілкування викладача й студентів у дистанційному навчанні. Кожний форум присвячений будь-якій проблемі або темі. Модератор форуму (мережевий викладач) реалізує дискусію чи обговорення, стимулюючи питаннями, повідомленнями, новою цікавою інформацією. Програмне забезпечення форумів дозволяє приєднати різні файли певного розміру. Кілька форумів можна об'єднати в один великий. Наприклад, під час роботи малої групи студентів над проектом, створюються форуми для кожної окремої групи з метою спілкування під час проведення дослідження над вирішенням поставленого для даної групи завдання, потім – обговорення загальної проблеми проекту усіма учасниками навчального процесу (веб-конференція).

Чат – спілкування користувачів мережі в режимі реального часу, засіб оперативного спілкування людей через Інтернет. Є кілька різновидів чатів: текстовий, голосовий, аудіовідеочат. Найбільш поширений текстовий чат. Голосовий чат дозволяє спілкуватися за допомогою голосу, що під час вивчення іноземної мови у дистанційній формі є важливим моментом. В освітніх цілях за необхідності можна організувати спілкування в чатах з носіями мови. Це реальна можливість мовної практики, яка проводиться у рамках запропонованої для дискусії проблеми, сумісної проектної діяльності, обміну інформацією.

Відеоконференція – це конференція реального часу в on-line режимі. Вона проводиться у визначений день і в призначений час. Для якісного проведення відеоконференції, як і телеконференції, необхідна її чітка підготовка: створення програм (цим займається мережевий викладач), своєчасна інформація на сайті і розсилка за списком (виконує педагог-куратор). Відеоконференція – один із сучасних способів зв'язку, що дозволяє проводити заняття у «віддалених класах», коли студенти і викладач знаходяться на відстані. Отже, обговорення й прийняття рішень, дискусії, захист проектів відбуваються у режимі реального часу. Викладач і студенти можуть бачити один одного, викладач має можливість супроводжувати лекцію наочним матеріалом.

Блог. Це форма спілкування, яка нагадує форум, де право на публікацію належить одній особі чи групі людей. В Інтернет-курсі з мови технології блогу можна використовувати під час навчання у колективі. Наприклад, автор (один студент чи їх група) виконав певне завдання (вір, есе), яке розміщується на сайті свого мережевого щоденника (блогу), потім автор блогу дає можливість іншим студентам прочитати і прокоментувати розміщений матеріал.

Середовище Classroom дозволяє організувати онлайн навчання, використовуючи відео, текстову та графічну інформацію, різні додатки Google. Викладач має можливість контролювати, систематизувати, оцінювати діяльність, переглядати результати виконання вправ, застосовувати різні форми оцінювання.

Сервіс ZOOM.US – дозволяє безкоштовно використовувати для проведення онлайн-занять. Його можливості:

- спільне використання екрану;
- проведення інтегративних вебінарів;
- робота з Google-дискон, Dropbox і Box;
- групові чати для обміну текстом, зображеннями, аудіо і т. д.;
- відеозапис віртуальних зустрічей.

Соціальні мережі та Viber дозволяють створювати закриті групи, чати, обговорення тем, завдань, проблем, інформації. Наприклад, Facebook можна використовувати для отримання завдань студентами, проведення консультацій, взаємообміну довідковою інформацією, проходження тестів та іншим шляхом організації спільноти.

Розкривати нову тему, давати нові знання, провокувати думки можна також, використовуючи:

- тематичний контент Youtube (тематичне відео, короткі ролики, фільми- екранізації, освітні канали тощо);
- інфографіку;
- аудіо-книги або підкасти, інтерв'ю з цікавими людьми;
- youtube-канали викладачів-предметників;
- блоги вчителів (викладачів)-предметників (за категоріями) та ін.

Інтернет-навчання як основа безперервної освіти націлене на оволодіння студентами навичок самостійної освітньої роботи, на формування у них ключових компетентностей.

Для організації дистанційного та індивідуального навчання в закладах освіти можна також використовувати наступні веб-технології та програмні засоби.

1. Платформа Moodle – безкоштовна платформа з широкими можливостями кастомізації. Встановлюється тільки на свій сервер. Є безліч плагінів для розширення функціоналу. Вимагає навичок web-розробки для адміністрування.

2. Платформа Google Classroom – це інструмент, що пов'язує Google Docs, Google Drive і Gmail, допомагає створювати і впорядковувати завдання, виставляти оцінки, коментувати і організовувати ефективне спілкування зі студентами в режимі реального часу або в режимі дистанційного навчання. Завдяки платформі Classroom можна організувати проектну роботу, перевернуте навчання, дистанційне навчання.

3. Веб-додаток Edmodo – освітній сайт, який являє собою усічену соціальну мережу за типом Facebook, яка дозволяє спілкуватися викладачам та студентам.

4. Платформа LearningApps.org – онлайн-сервіс, який дозволяє створювати інтерактивні вправи. Він є конструктором для розробки різноманітних завдань з різних предметних галузей для використання і на заняттях, і позанавчальний час. Крім того, Learningapps.org надає можливість дистанційного навчання кожному викладачу, адже дозволяє створити набір класів у власному акаунті, ввести дані про студентів, їх профілі, задати пароль для входу та викладати вправи для виконання.

5. Платформа Learning.ua – містить онлайн тести, інтерактивні завдання з математики, української мови.

6. Платформа Stepik – безкоштовна освітня платформа та конструктор відкритих онлайн-курсів та уроків.

7. Хмарні сервіси (Office365, Google) для спільної роботи студентів та викладача [3].

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Регулярні вирази: синтаксис, сфера використання

Модуль `re` з'явився в Python 1.5 і працював з регулярними виразами стилю Perl. Старіші версії Пайтона поставлялись з модулем `regex`, який реалізовував регулярні вирази в стилі Emacs. Модуль `regex` був вилучений з Python 2.5.

Регулярні вирази (часті скорочення: RE, `regex`) це спеціальна малесенька мова програмування вбудована в Python за допомогою модуля `re`. За допомогою цієї мови можна описувати множини слів (рядків), які відповідають шаблону (регулярному виразу). Множинами можуть бути всі правильні адреси e-mail, команди TeX, чи будь-що що ви захочете. Потім можна отримувати відповіді на запитання "Чи відповідає цей рядок шаблону?", або "Чи зустрічається у цьому рядку послідовність символів що відповідає шаблону?". Також можна використовувати регулярні вирази, щоб робити заміни в рядку, чи розділяти її на складові різними способами.

Регулярні вирази компілюються в послідовність байткодів, які потім виконуються підпрограмою написаною на C.

Через те, що регулярні вирази - доволі мала та обмежена мова, тому не всі завдання з обробки рядків можна виконати з їх допомогою. Також є задачі, які робляться з регулярними виразами, але вони виходять занадто складними, і ефективніше написати код обробки рядка на Python [4].

Синтаксис регулярних виразів залежить від інтерпретатора, що використовується для їхньої обробки. Однак, із незначними відхиленнями, майже всі поширені інтерпретатори регулярних виразів мають спільні правила.

Найпростішим регулярним виразом, з якого формуються складні, є звичайний символ.

Більшість символів у регулярному виразі представляють самі себе, за винятком спеціальних символів (метасимволів) `[] \ ^ $. | ? * + () { }`, яким може передувати символ `\` (зворотна коса риска), котрий робить метасимволи «екранованими», «захищеними» для представлення, відображення їх самих як символів тексту. Можна екранувати деяку послідовність символів, розмістивши її між `\Q` і `\E`.

Аналогічно можуть бути представлені інші спеціальні символи (набір символів, що вимагають екранування, може відрізнитися залежно від конкретної реалізації). Частина символів, які в тій або іншій реалізації не вимагають екранування (наприклад, кутові дужки `<` `>`), можуть бути екрановані з міркувань зручності читання.

Залежно від інтерпретатора регулярних виразів, метасимволи «`?`», «`+`», «`{}`», «`|`», «`(`», та «`)`» можуть втрачати своє спеціальне значення, замість цього слід вживати «`\?`», «`\+`», «`\{}`», «`\|`», «`\(`», та «`\)`».

Метасимволу `.` (крапка) відповідає довільний символ, окрім символу нового рядка (в деяких реалізаціях).

Набір символів у квадратних дужках `[]` називається символьним класом і дозволяє вказати інтерпретаторові регулярних виразів, що на даному місці в рядку може стояти один із перерахованих символів.

Зокрема, `[абв]` задає можливість появи в тексті одного із трьох зазначених символів, а `[1234567890]` задає відповідність одній із цифр. Можливе зазначення діапазонів символів: наприклад, `[0-9]`.

Для включення усіх символів українського алфавіту можна використовувати `[Є-ЯГ]`, `[a-ïr]`.

Якщо потрібно вказати символи, які не входять у зазначений набір, то використовують символ `^` усередині квадратних дужок, наприклад, `[^0-9]` означає будь-який символ, крім цифр.

Деякі символьні класи можна замінити спеціальними метасимволами:

Таблиця 3.1 – Спеціальні метасимволи

Символ	Опис
<code>\d</code>	Відповідає цифрі. Еквівалентно <code>[0-9]</code>
<code>\D</code>	Відповідає нецифровому символу. Еквівалентно <code>[^0-9]</code>
<code>\s</code>	Відповідає будь-якому пробільному символу. Еквівалентно <code>[\f\n\r\t\v]</code>
<code>\S</code>	Відповідає будь-якому непробільному символу. Еквівалентно <code>[^\f\n\r\t\v]</code>
<code>\w</code>	Відповідає будь-якому літерному символу, цифровому й знаку підкреслення. Еквівалентно <code>[:word:]</code>
<code>\W</code>	Відповідає будь-якому символу, крім літерного символу, цифрового або підкреслення. Еквівалентно <code>[^[:word:]]</code>

Наступні символи дозволяють позиціонувати регулярний вираз щодо елементів тексту: початку й кінця рядка, меж слова.

Таблиця 3.2 – Позиція всередині рядка

Представлення	Позиція	Приклад	Відповідність
<code>^</code>	Початок рядка	<code>^a</code>	<code>aaa aaa</code>
<code>\$</code>	Кінець рядка	<code>a\$</code>	<code>aaa aaa</code>
<code>\b</code>	Межа слова	<code>a\b</code>	<code>aaa aaa</code>
		<code>\ba</code>	<code>aaa aaa</code>
<code>\B</code>	Не межа слова	<code>\Ba\B</code>	<code>aaa aaa</code>
<code>\G</code>	Попередній успішний пошук	<code>\Ga</code>	<code>aaa aaa</code> (пошук зупинився на 4-й позиції — там, де не знайшлося а)

Квантифікатор після символу, символного класу або групи визначає, скільки разів попередній вираз може зустрічатися. Варто враховувати, що квантифікатор може стосуватися більш ніж до одного символу в регулярному виразі, тільки якщо це символний клас або група.

Таблиця 3.3 – Квантифікація (пошук послідовностей)

Представлення	Кількість повторень	Еквівалент	Приклад	Відповідність
{ <i>n</i> }	Рівно <i>n</i> разів		colou{3}r	colouuur
{ <i>m,n</i> }	Від <i>m</i> до <i>n</i> включно		colou{2,4}r	colouur, colouuur, colouuuur
{ <i>m</i> ,}	Не менше <i>m</i>		colou{2,}r	colouur, colouuur, colouuuur і т.д.
{, <i>n</i> }	Не більше <i>n</i>		colou{,3}r	color, colour, colouur, colouuuur
*	Нуль або більше	{0,}	colou*r	color, colour, colouur і т.д.
+	Одне або більше	{1,}	colou+r	colour, colouur і т.д. (але не color)
?	Нуль або одне	{0,1}	colou?r	color, colour

Круглі дужки використовуються для визначення області дії й пріоритету операцій. Шаблон усередині групи обробляється як єдине ціле й може бути квантифікованим. Наприклад, вираз (тр[ау] м-?)* знайде послідовність виду трумтрам-трум-трамтрум.

Одне із застосувань групування — повторне використання раніше знайдених груп символів (*підрядків, блоків, позначених підвиразів*). При обробці виразу підрядки, що знайдені за шаблоном усередині групи, зберігаються в окремій області пам'яті й отримують номер, починаючи з

одиниці. Кожному підрядку відповідає пара дужок у регулярному виразі. Квантифікація групи не впливає на збережений результат, тобто зберігається лише перше входження. Зазвичай підтримується до 9 нумерованих підрядків із номерами від 1 до 9, але деякі інтерпретатори дозволяють працювати з більшою кількістю. Згодом у межах даного регулярного виразу можна використати позначення від $\backslash 1$ до $\backslash 9$ для перевірки на збіг із раніше знайденим підрядком.

Наприклад, регулярний вираз $(ta|tu)\backslash 1$ знайде рядок та-та або ту-ту, але пропустить рядок та-ту.

Також раніше знайдені підрядки можна використовувати при заміні за регулярним виразом. У такому разі в текст, що заміщає, вставляються ті ж позначення, що й у межах самого виразу [5].

Вивчимо найпростіші регулярні вирази. Найпростіша задача, яку вони виконують – пошук відповідностей шаблону.

Нехай A – алфавіт.

Клас регулярних виразів визначається такими правилами:

- 1) ε є регулярним виразом;
- 2) якщо $a \in A$, то a – регулярний вираз;
- 3) якщо w – регулярний вираз, то w^* і (w) – регулярні вирази;
- 4) якщо w_1 і w_2 – регулярні вирази, то w_1w_2 , w_1/w_2 – регулярні вирази.

У виразі w_1w_2 використовується операція катенації, у виразі w_1/w_2 – операція об'єднання.

Приклади.

$$L(a^*) = \{a, aa, aaa, \dots\}.$$

$$L(a/b/c) = \{a, b, c\}.$$

$$L(ab^*c) = \{ac, abc, abbc, \dots\}.$$

$$L(ab^+c) = \{abc, abbc, \dots\}.$$

$$L(a(b/c)d) = \{abd, acd\}.$$

Зайві дужки (необов'язково усі) можуть бути видалені, якщо прийняти такі співвідношення: операція ітерація “*” має вищий пріоритет, катенація – другий за значенням і “|” – нижчий.

Приклад виконання завдання. Написати регулярний вираз для мови, яка являє собою множину ланцюжків у алфавіті $\{a,b,c\}$, що містять хоча б один символ a і хоча б один символ b .

Потрібний вираз може бути представлений, наприклад, у такій формі [4]:

$$((a/b/c)^* a (a/b/c)^* b (a/b/c)^*) / ((a/b/c)^* b (a/b/c)^* a (a/b/c)^*) .$$

Наведемо декілька словесних прикладів регулярних виразів: «три цифри» (такому шаблону відповідають рядки «121», «424», але не «23абв»), «дві цифри, після яких стоїть крапка, за якою йдуть кілька цифр, проте не менше однієї» (цьому шаблону відповідають «11.2», «11.4а», але не «11. »). Регулярні вирази подібні до арифметичних у тому, що формуються з менших за допомогою операторів [5].

3.2. Алгоритмізація задачі за темою роботи

На стартовій сторінці виводиться інформація про тренажер:

- ПІБ розробника;
- група студента;
- тема тренажеру.

Також присутня сторінка з теоретичним матеріалом.

Після переходу до виконання тренажеру виводиться перший крок.

Крок 1. Відображається питання і варіанти відповіді: «Найпростішим регулярним виразом, з якого формуються складні, є:»

- сукупність символів;
- звичайний символ;
- звичайний вираз.

Якщо вибрано правильний варіант (2), то перехід на наступний крок. В іншому разі виводиться повідомлення про помилку: «Помилка! Відповідь неправильна».

Крок 2. Відображається питання і варіанти відповіді: «Більшість символів у регулярному виразі представляють самі себе, за винятком спеціальних символів (метасимволів):»

- `[]\^$.|?*+(){};`
- `[]^$.?*+(){};`
- `\^$.|?*+() .`

Якщо вибрано правильний варіант (1), то перехід на наступний крок. В іншому разі виводиться повідомлення про помилку: «Помилка! Відповідь неправильна».

Крок 3. Відображається питання і потрібно заповнити пусті комірки: «Залежно від інтерпретатора регулярних виразів, метасимволи «?», «+», «{», «|», «(», та «)» можуть втрачати своє спеціальне значення, замість цього слід вживати:»

- `«__?», «__+», «__ {», «__|», «__ (», «__)».`

Якщо вибрано правильний варіант (\), то перехід на наступний крок. В іншому разі виводиться повідомлення про помилку: «Помилка! Відповідь неправильна».

Крок 4. Відображається питання і варіанти відповіді: «Якщо потрібно вказати символи, які не входять у зазначений набір, то усередині квадратних дужок використовують символ:»

- `?`;
- `\`;
- `^`.

Якщо вибрано правильний варіант (3), то перехід на наступний крок. В іншому разі виводиться повідомлення про помилку: «Помилка! Відповідь неправильна».

Крок 5. Відображається питання і потрібно встановити відповідність:
«Деякі символні класи можна замінити спеціальними метасимволами:»

Символ	Опис
\d	Відповідає цифрі. Еквівалентно [0-9]
\D	Відповідає нецифровому символу. Еквівалентно [^0-9]
\s	Відповідає будь-якому пробільному символу. Еквівалентно [\f\n\r\t\v]
\S	Відповідає будь-якому непробільному символу. Еквівалентно [^\f\n\r\t\v]
\w	Відповідає будь-якому літерному символу, цифровому й знаку підкреслення. Еквівалентно [[:word:]]
\W	Відповідає будь-якому символу, крім літерного символу, цифрового або підкреслення. Еквівалентно [^[:word:]]

Якщо вибрано все правильно, то перехід на наступний крок. В іншому разі виводиться повідомлення про помилку: «Помилка! Відповідь неправильна».

Крок 6. Відображається питання і потрібно встановити відповідність:
«Наступні символи дозволяють позиціонувати регулярний вираз щодо елементів тексту: початку й кінця рядка, меж слова:»

Представлення	Позиція
^	Початок рядка
\$	Кінець рядка
\b	Межа слова
\B	Не межа слова
\G	Попередній успішний пошук

Якщо вибрано все правильно, то перехід на наступний крок. В іншому разі виводиться повідомлення про помилку: «Помилка! Відповідь неправильна».

Крок 7. Відображається питання і потрібно встановити відповідність: «Для кожного прикладу встановіть його відповідність:»

Приклад	Відповідність
$\wedge a$	aaa aaa
$a\$$	aaa aaa
$a\backslash b$	aaa aaa
$\backslash ba$	aaa aaa
$\backslash Ba\backslash B$	aaa aaa
$\backslash Ga$	aaa aaa

Якщо вибрано все правильно, то перехід на наступний крок. В іншому разі виводиться повідомлення про помилку: «Помилка! Відповідь неправильна».

Крок 8. Відображається питання і потрібно встановити відповідність: «Квантифікатор після символу, символного класу або групи визначає, скільки разів попередній вираз може зустрічатися. Варто враховувати, що квантифікатор може стосуватися більш ніж до одного символу в регулярному виразі, тільки якщо це символний клас або група:»

Представлення	Кількість повторень
$\{n\}$	Рівно n разів
$\{m,n\}$	Від m до n включно
$\{m,\}$	Не менше m
$\{,n\}$	Не більше n

Якщо вибрано все правильно, то перехід на наступний крок. В іншому разі виводиться повідомлення про помилку: «Помилка! Відповідь неправильна».

Крок 9. Відображається питання і потрібно встановити відповідність: «Аналогічно для наступних представлень:»

Представлення	Кількість повторень	Еквівалент
*	Нуль або більше	{0,}
+	Одне або більше	{1,}
?	Нуль або одне	{0,1}

Якщо вибрано все правильно, то перехід на наступний крок. В іншому разі виводиться повідомлення про помилку: «Помилка! Відповідь неправильна».

Крок 10. Відображається питання і потрібно встановити відповідність: «Для кожного представлення вкажіть його приклад:»

Представлення	Приклад
{n}	colou{3}r
{m,n}	colou{2,4}r
{m,}	colou{2,}r
{,n}	colou{,3}r

Якщо вибрано все правильно, то перехід на наступний крок. В іншому разі виводиться повідомлення про помилку: «Помилка! Відповідь неправильна».

Крок 11. Відображається питання і потрібно встановити відповідність: «Для кожного прикладу вкажіть його відповідність:»

Представлення	Приклад	Відповідність
{n}	colou{3}r	colouuuur

$\{m,n\}$	colou{2,4}r	colouur, colouuur, colouuuur
$\{m,\}$	colou{2,}r	colouur, colouuur, colouuuur і т.д.
$\{,n\}$	colou{,3}r	color, colour, colouur, colouuur

Якщо вибрано все правильно, то перехід на наступний крок. В іншому разі виводиться повідомлення про помилку: «Помилка! Відповідь неправильна».

Крок 12. Відображається питання і потрібно встановити відповідність: «Для кожного представлення вкажіть його приклад:»

Представлення	Приклад
*	colou*r
+	colou+r
?	colou?r

Якщо вибрано все правильно, то перехід на наступний крок. В іншому разі виводиться повідомлення про помилку: «Помилка! Відповідь неправильна».

Крок 13. Відображається питання і потрібно встановити відповідність: «Для кожного прикладу вкажіть його відповідність:»

Представлення	Приклад	Відповідність
*	colou*r	color, colour, colouur і т.д.
+	colou+r	colour, colouur і т.д. (але не color)
?	colou?r	color, colour

Якщо вибрано все правильно, то перехід на наступний крок. В іншому разі виводиться повідомлення про помилку: «Помилка! Відповідь неправильна».

Крок 14. Відображається питання і потрібно заповнити пусті комірки: «Круглі дужки використовуються для визначення області дії й пріоритету операцій. Шаблон усередині групи обробляється як єдине ціле й може бути квантифікованим. Наприклад, вираз $(tr[ay] m-?)^*$ знайде послідовність виду:»

- _____.

Якщо вказано правильний варіант (трумтрам-трум-трамтрум), то перехід на наступний крок. В іншому разі виводиться повідомлення про помилку: «Помилка! Відповідь неправильна».

Крок 15. Відображається питання і варіанти відповіді: «Одне із застосувань групування — повторне використання раніше знайдених груп символів (*підрядків, блоків, позначених підвиразів*). При обробці виразу підрядки, що знайдені за шаблоном усередині групи, зберігаються в окремій області пам'яті й отримують номер, починаючи з одиниці. Наприклад, регулярний вираз $(ta|tu)-\backslash 1$ знайде рядок:»

- та-та;
- та-ту;
- ту-ту.

Якщо вибрано правильний варіант (1 і 3), то перехід на наступний крок. В іншому разі виводиться повідомлення про помилку: «Помилка! Відповідь неправильна».

Крок 16. Відображається питання і потрібно заповнити пусті комірки: «Написати регулярний вираз для мови, яка являє собою множину ланцюжків у алфавіті $\{a,b,c\}$, що містять хоча б один символ a і хоча б один символ b .»

- (_____).

Якщо вказано правильний варіант

$$((a/b/c)*a(a/b/c)*b(a/b/c)*) / ((a/b/c)*b(a/b/c)*a(a/b/c)*),$$

то перехід на наступний крок. В іншому разі виводиться повідомлення про помилку: «Помилка! Відповідь неправильна».

Крок 17. Відображається питання і варіанти відповіді: «Наведемо декілька словесних прикладів регулярних виразів: «три цифри» (такому шаблону відповідають рядки):»

- «121»;
- «424»;
- «23абв».

Якщо вибрано правильний варіант (1 і 2), то перехід на наступний крок. В іншому разі виводиться повідомлення про помилку: «Помилка! Відповідь неправильна».

Крок 18. Відображається питання і варіанти відповіді: «Наведемо декілька словесних прикладів регулярних виразів: «дві цифри, після яких стоїть крапка, за якою йдуть кілька цифр, проте не менше однієї» (цьому шаблону відповідають):»

- «11.»;
- «11.2»;
- «11.4а».

Якщо вибрано правильний варіант (2 і 3), то перехід на наступний крок. В іншому разі виводиться повідомлення про помилку: «Помилка! Відповідь неправильна».

Крок 19. Відображається повідомлення про завершення: «Вітаємо! Ви пройшли тренажер з теми «Регулярні вирази: синтаксис, сфера використання». Бажаєте закріпити матеріал і розпочати спочатку?». Якщо так – перехід на крок 1, якщо ні – кінець роботи тренажеру.

4. ПРАКТИЧНА ЧАСТИНА

4.1. Обґрунтування вибору програмних засобів для реалізації завдання роботи

Якщо б мову Java використовували для створення машинно незалежних застосувань, то і цього було б достатньо для її успіху в програмістів. Однак 1993 р. компанія Sun звернула увагу на зростання популярності Internet і почала доробляти мову Java.

Серед початківців розповсюджена хибна думка, що оскільки мову Java використовують для створення Web-сторінок, то її можна вважати деяким вдосконаленням HTML (Hypertext Markup Language – мова розмітки гіпертексту). Насправді це не так. HTML є засобом логічної організації інформації і створення гіпер-текстових зв'язків з відповідними даними. Вона дає змогу читати документи не тільки зверху вниз, а й у будь-якому іншому порядку, проте ніколи не була мовою програмування. Єдиний зв'язок між HTML і Java – це наявний в HTML дескриптор APPLET, за допомогою якого викликається для виконання аплет Java.

Компанія JavaSoft, створена розробниками мови Java, пропонує безкоштовно набір засобів для програмістів мовою Java JDK (Java Development Kit) за адресою [http:// java.sun.com/products/jdk/](http://java.sun.com/products/jdk/). JDK містить все необхідне для створення програм: базові функції мови, інтерфейс прикладного програмування (API) з наборами пакетів й основні інструменти. Більшість версій JDK містять сім інструментів розробки на Java:

- 1) компілятор (javac);
- 2) генератор документації (javadoc);
- 3) генератор файлів заголовків і заглушок мови C++ для Java (javah);
- 4) інтерпретатор (java);
- 5) програму перегляду аплетів (appletviewer);

- б)реасемблер файлів класів;
- 7)відлагоджувач програм (jdb).

Процес встановлення SDKналічує три етапи:1)отримання SDK(переважно з Internet); 2)встановлення SDK; 3)перевірка конфігурації.При перевірціконфігурації необхідно визначити правильне встановлення змінних оточення PATH, яка вказує на каталог з інструментами SDK, та CLASSPATH, яка задає шлях до каталогів класів Java(як готових, так і власних).

Конкурент компанії Microsoftна ринку продуктів з Javaкомпанія Symantесвоїм важливим козирем вважає те, що її Visual Cafe цілковито відповідає стандартіві Java. Компанія Symantесвизначає Visual J++як дещо підправлене та адаптоване середо-вище C++від Microsoft. Фірма Borlandтакож намагається не відставати від конку-рентів і пропонує своє середовище JBuilder. Детально познайо-митися з JBuilderможна на Web-сервері за адресою [http:// www.borland.com/jbuilder/](http://www.borland.com/jbuilder/) [7].

Розглянемо причини використання Java:

Причина #1 Java ще не скоро втратить актуальність.

Від моменту появи 1995 року, від початку глобальної комп'ютеризації і до сьогодні — мова програмування Java стабільно користується попитом на ринку. З того часу Java довела свою затребуваність.

До речі! Мову програмування C# Microsoft свого часу розробив як покращений варіант Java.

Причина #2 Java вирішує широке коло задач.

Завдяки простоті та надійності Java використовують у різних сферах життя: для розробки ПЗ в державній сфері, в науці, освіті, сфері охорони здоров'я, в приватному секторі при створенні програм для трейдингу, серверні додатки для банкінгу та для багатьох інших корпоративних і ентерпрайз цілей.

Причина #3 Java продовжує активно розвиватися.

Java 8 привнесла в світ Java концепцію функціонального програмування. Воно додало тієї самої гнучкості, за відсутність якої раніше критикували цю мову і протиставляли інші мови програмування.

Причина #4 Java мова програмування для розробки додатків під Android.

Завдяки змінам у Java 5 Java 6 версіях зросла її продуктивність, і це одна з причин чому Google обрала мову програмування Java основною для розробки додатків під Android

Причина #5 Гарантоване працевлаштування, все тільки залежить від вас.

Стосовно працевлаштування, то дана мова програмування перевершить всі показники у порівнянні з іншими мовами. Будь який програміст або тестувальник спроможний отримати тонну вакансій і подальшу роботу, вивчаючи мову програмування Java.

4.2. Опис роботи програми

Після відкриття програми відображається початкове вікно (рис. 4.1).

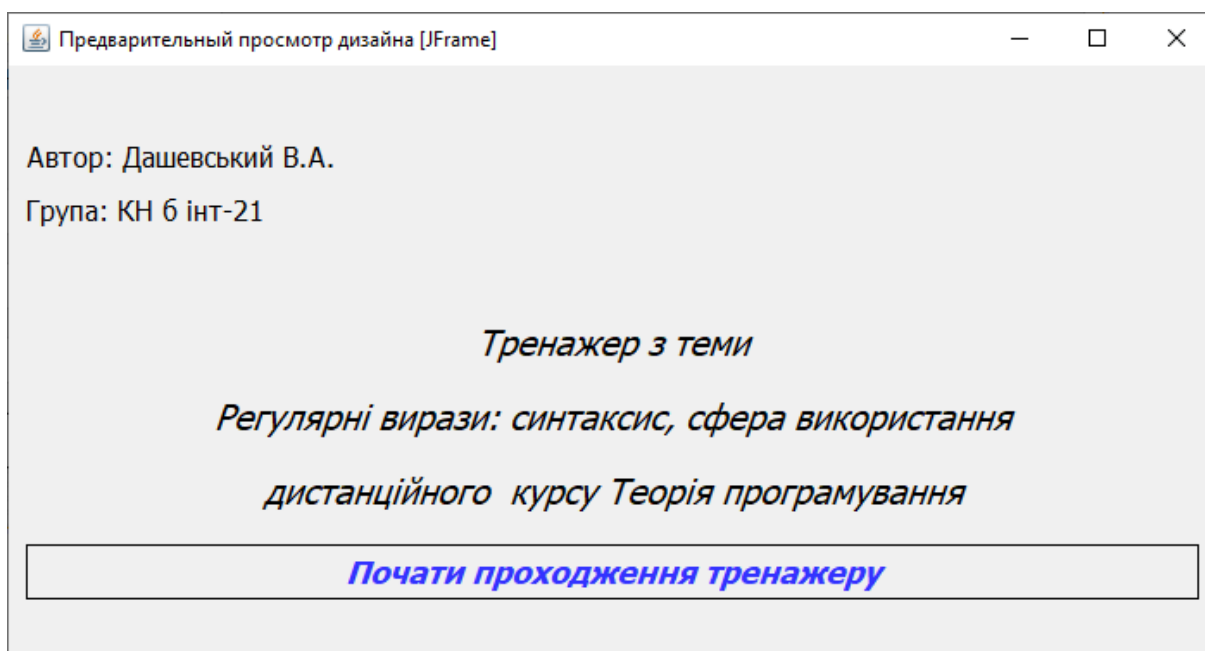


Рисунок 4.1 – Початкове вікно

Щоб почати проходження потрібно натиснути «Почати проходження тренажеру». Виводиться перший крок, потрібно обрати відповідь (рис. 4.2).

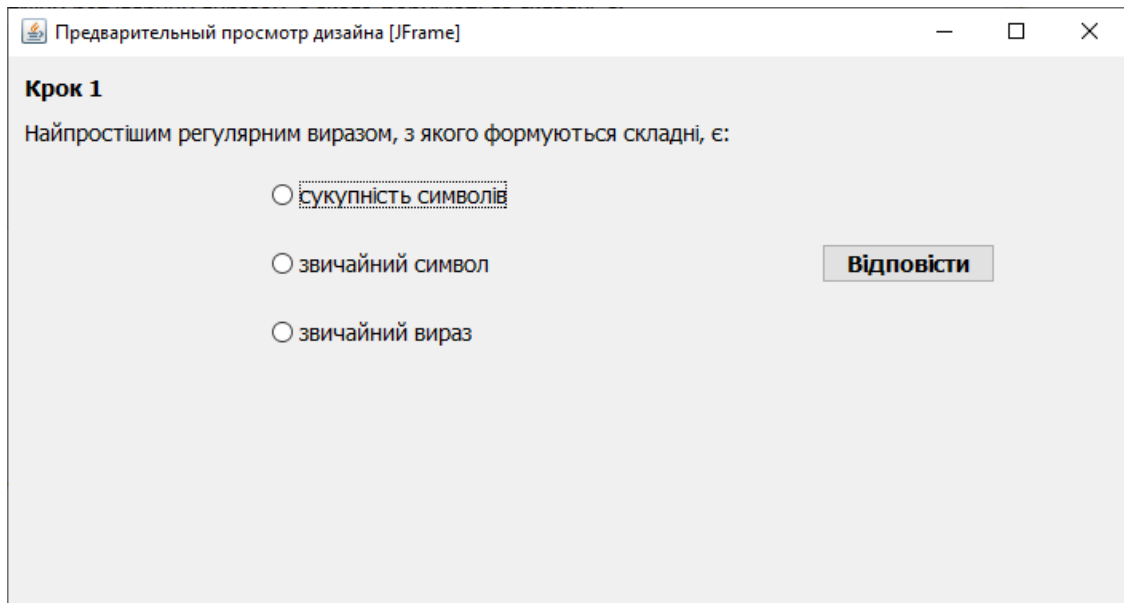


Рисунок 4.2 – Перший крок

Після правильної відповіді відображається наступний крок (рис. 4.3).

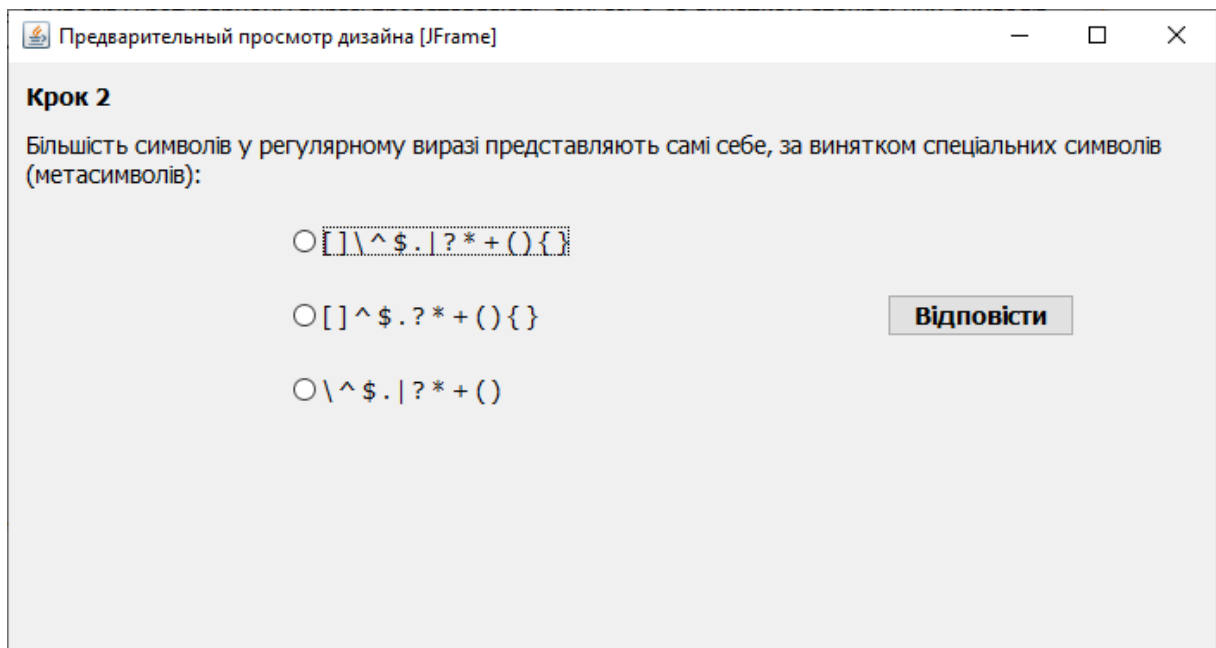


Рисунок 4.3 – Другий крок

На третьому кроці вже необхідно вводити відповіді в пусті поля (рис. 4.4). Також, як приклад, крок 5 (рис. 4.5).

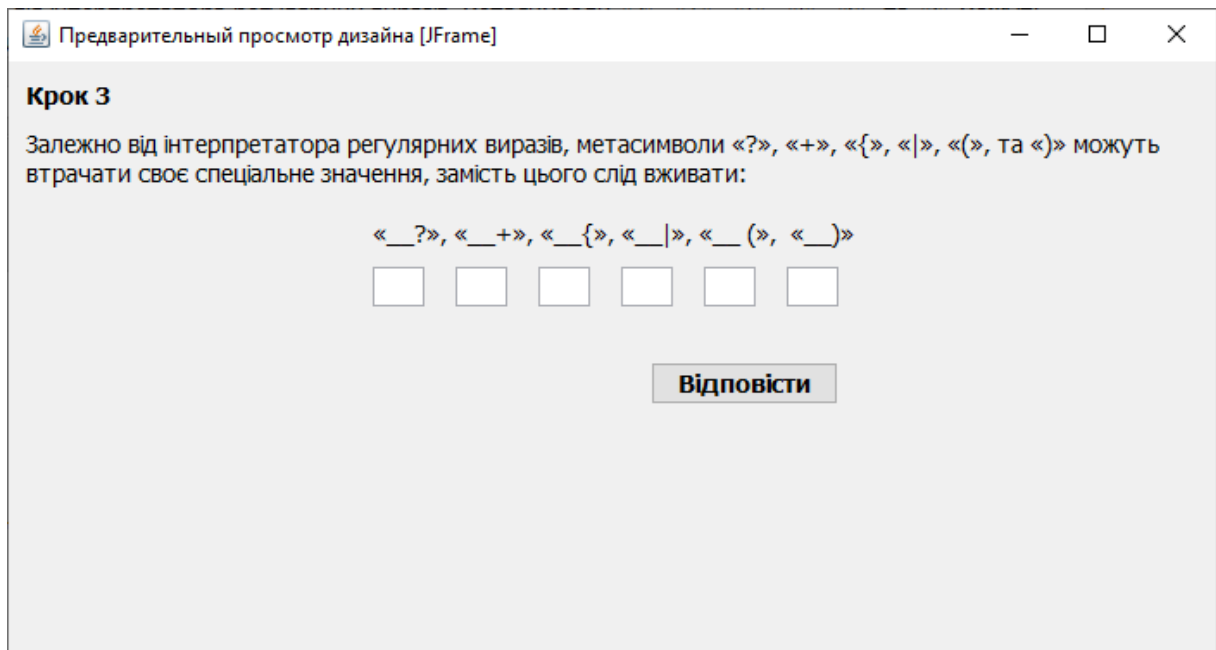


Рисунок 4.4 – Третій крок

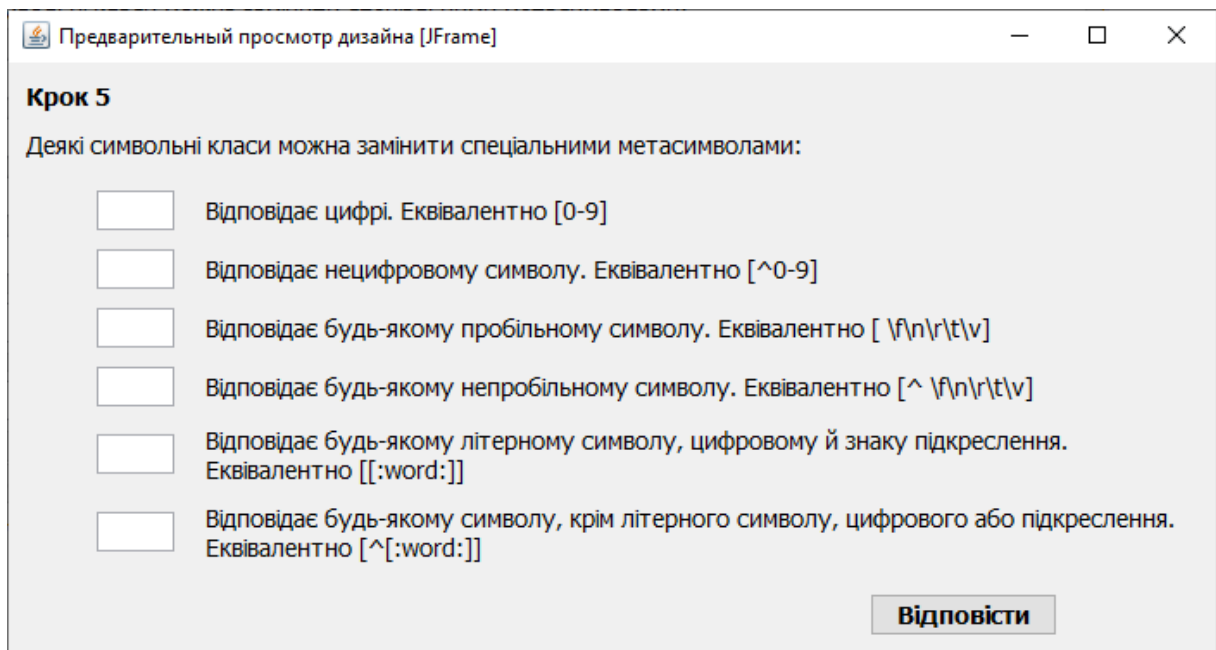


Рисунок 4.5 – П'ятий крок

Деякі кроки вимагають встановити відповідність (рис. 4.6, 4.7).

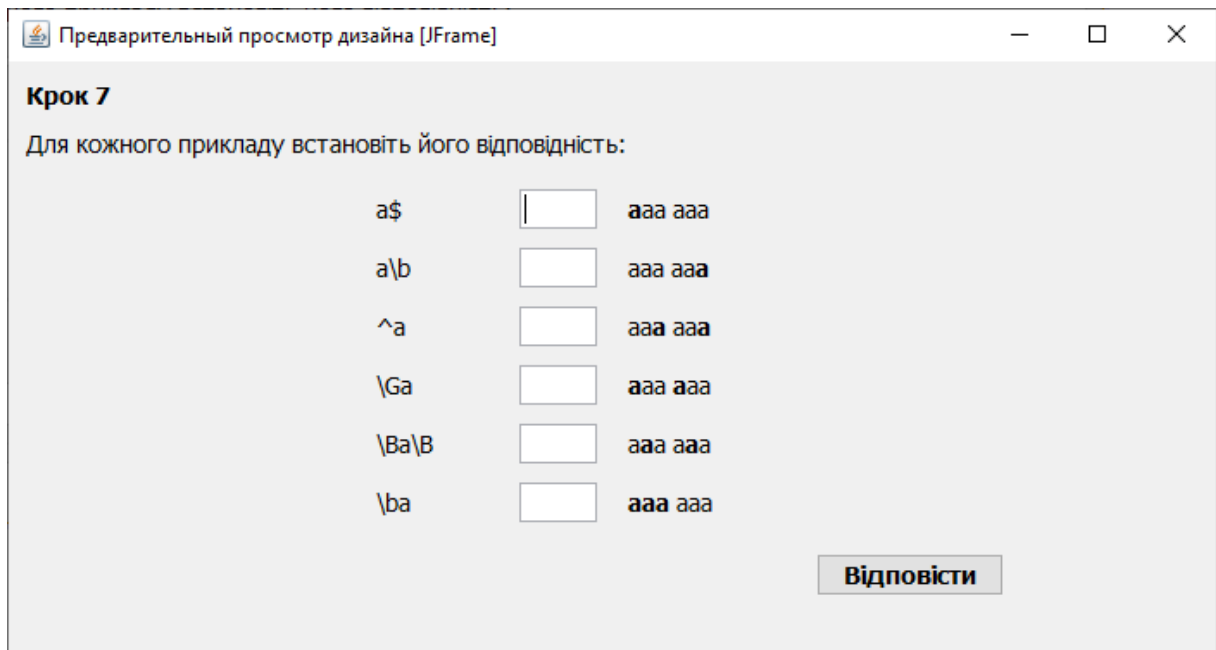


Рисунок 4.6 – Сьомий крок

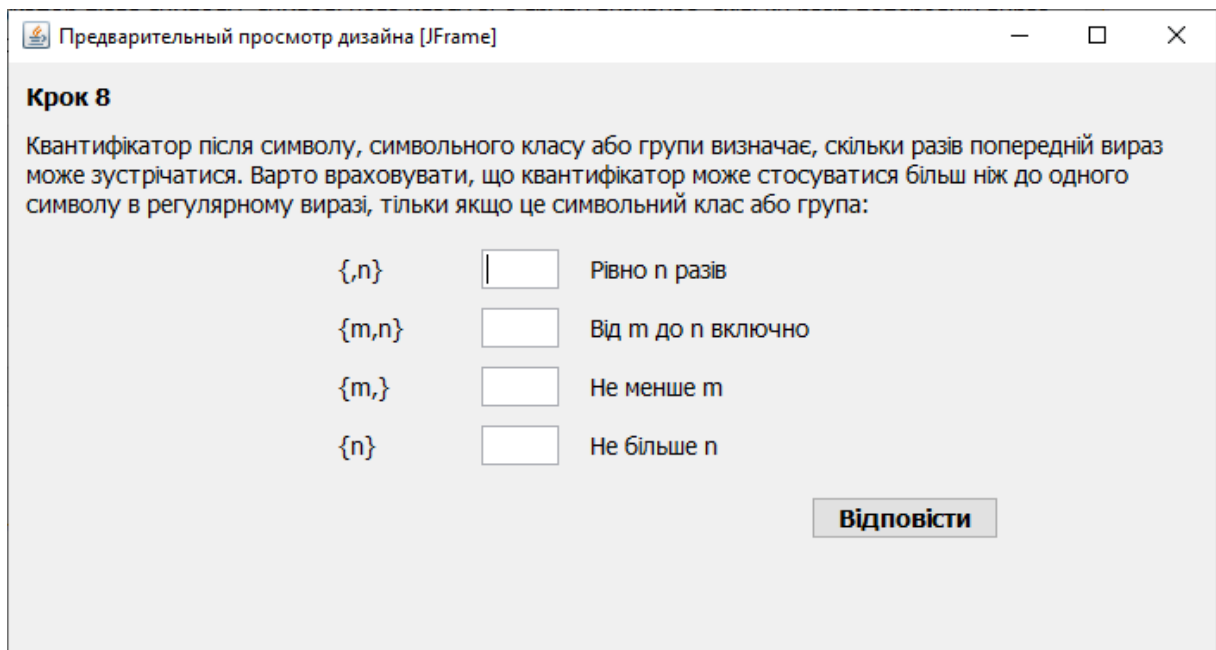


Рисунок 4.7 – Восьмий крок

ВИСНОВКИ

Віртуальні тренажери дозволяють значно скоротити час засвоєння матеріалу, а отже прискорити процес навчання. Студенти зможуть наочно розібрати досліджувану тему стільки разів, скільки їм потрібно для набуття навичок.

Існує ряд тем, які викликають значні проблеми у студентів, що часто пов'язано з труднощами наочної демонстрації. В таких випадках доцільно використання тренажерів. Наприклад, при вивченні теми “Табличний процесор Microsoft Excel” дуже важливо навчити студентів створювати та копіювати формули.

Основний принцип, що повинен бути покладений в основу створення комп'ютерних тренажерів – принцип відповідності засобів навчання змісту навчання і дидактичним завданням

Пропонується розробити тренажер, що складається з трьох етапів:

1. *Ознайомлювальний етап.* Користувач ознайомлюється з тренажером, його темою. Надається можливість переглянути теорію перед початком.
2. *Практичне заняття* розробляється за сценарієм і містить в собі низку дій, які повинен виконати користувач, щоб вирішити поставлене завдання.
3. *Тестування.* У режимі тестування користувач уже без підказок повинен виконати необхідні дії, а система автоматично здійснить оцінювання результату цих дій.

Також було розглянуто поняття регулярних виразів:

- Синтаксис
 - Представлення символів
 - Позиція всередині рядка
 - Квантифікація (пошук послідовностей)
 - Групування

- Різне

- Приклади

В алгоритмі описано наступну структуру роботи.

На стартовій сторінці виводиться інформація про тренажер:

- ПІБ розробника;
- група студента;
- тема тренажеру.

Також присутня сторінка з теоретичним матеріалом.

Після переходу до виконання тренажеру виводиться перший крок.

Після проходження тренажеру відображається повідомлення про завершення: «Вітаємо! Ви пройшли тренажер з теми «Регулярні вирази: синтаксис, сфера використання». Бажаєте закріпити матеріал і розпочати спочатку?». Якщо так – перехід на крок 1, якщо ні – кінець роботи тренажеру.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ємець О. О. Методичні рекомендації щодо оформлення пояснювальних записок до курсових проектів (робіт) для студентів за освітньою програмою «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки та інформаційні технології», «Комп'ютерні науки» галузь знань - 12 «Інформаційні технології» / О. О. Ємець – Полтава : РВВ ПУЕТ, 2017. – 69с.
2. Черненко О.О. Електронний навчально-методичний посібник для самостійного вивчення навчальної дисципліни «Теорія програмування» для студентів напряму 6.040302 «Інформатика»
3. Про затвердження Положення про дистанційне навчання МОН України; Наказ, Положення від 25.04.2013 № 466 / Зареєстровано: Мін'юст України від 30.04.2013 № 703/23235 з0703-13, Чинний, Редакція від 16.10.2020, 28 кб (4 редакції). – Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0703-13#Text>
4. Методичні рекомендації по дистанційному навчанні – Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0703-13#Text>
5. Черненко О.О. Електронний навчально-методичний посібник для самостійного вивчення навчальної дисципліни «Теорія програмування» для студентів напряму 6.040302 «Інформатика» – Режим доступу: <http://tprogr.ho.ua>.
6. Регулярний вираз / Матеріал з Вікіпедії — вільної енциклопедії. – Режим доступу: https://uk.wikipedia.org/wiki/Регулярний_вираз
7. Копитко М.Ф. Основи програмування мовою Java: Тексти лекцій. / М.Ф. Копитко, К.С. Іванків – Львів: Видавничий центр ЛНУ ім. Івана Франка, 2002.– 83 с.
8. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання: ДСТУ 7.1-2006. – [Чинний від 2007-07-01]. – К. : Держспоживстандарт України, 2007. – 47 с.

ДОДАТОК А