

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

**ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ
ІНФОРМАТИКИ**

Допускається до захисту

Завідувач кафедри _____ О.О. Ємець
(підпис)

« _____ » _____ 2020 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО БАКАЛАВСЬКОЇ РОБОТИ**

на тему

**АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ ТРЕНАЖЕРА «ВИДАЛЕННЯ
ЛІВОЇ РЕКУРСІЇ» ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО КУРСУ «ТЕОРІЯ
ПРОГРАМУВАННЯ»**

зі спеціальності 122 «Комп'ютерні науки»

Виконавець роботи Князєв Олексій Миколайович

_____ « ____ » _____ 2020р.
(підпис)

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

_____ « ____ » _____ 2020р.
(підпис)

ПОЛТАВА 2020р.

ЗМІСТ

ВСТУП	3
1. ПОСТАНОВКА ЗАДАЧІ.....	5
2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	12
3. ТЕОРЕТИЧНА ЧАСТИНА	18
3.1. Ліворекурсивні нетермінали та видалення лівої рекурсії.....	18
3.2. Розробка алгоритму роботи тренажеру	24
3.3. Розробка блок-схеми, яка підлягає програмуванню	34
4. ПРАКТИЧНА ЧАСТИНА	36
4.1. Обґрунтування вибору програмних засобів.....	36
4.2. Розробка тренажеру	42
ВИСНОВКИ.....	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	45
ДОДАТОК А. КОД ПРОГРАМИ	47

ВСТУП

Інформаційні технології застосовуються зараз практично у всіх сферах життя людства. Ці технології докорінно змінили світ. Значні успіхи в телекомунікаціях, енергетиці, транспорті, медицині, освіті, космічних дослідженнях та інших сферах діяльності людства досягнуто завдяки використанню інформаційних технологій.

Однак таке розширення сфери їх застосувань має і негативну сторону. Людство стає все більш залежним від надійного і правильного функціонування комп'ютерних систем; помилки в їх роботі часом ведуть до величезних фінансових, а навіть і людських втрат. Дотепер фірми-розробники програмного забезпечення не дають гарантію на програмні продукти, які вони пропонують користувачу. Але такого фактично немає в жодній іншій сфері діяльності.

Будь-який легальний продавець якогось товару дає гарантії якості і готовий нести (пригадаємо повернення мільйонів автомобілів на заводи виробника у разі виявлення дефектів) фінансову і юридичну відповідальність за поставку недоброякісного продукту. Можна навести і безліч інших прикладів, коли погане програмне забезпечення призводило до знищення космічних ракет, аварій на заводах, гідроелектростанціях тощо. Про складний стан справ говорять і самі фахівці галузі комп'ютерних технологій.

У чому ж причини? Вони, безумовно, різноманітні і пов'язані з науковими, економічними, соціальними, психологічними та іншими проблемами створення і використання інформаційних технологій. Серед наукових однією з найважливіших є проблема створення ефективних методів розробки програмних систем. Такі методи мають базуватися на розвиненій теорії програмування, яка має на меті дослідження програм та методів їх аналізу і синтезу. Відзначимо, що тут термін «програма»

тлумачиться у широкому смислі, охоплюючи, зокрема, як програми конкретних мов програмування, так і великі програмні системи.

Метою роботи є розробка тренажера з теми «Видалення лівої рекурсії» дистанційного навчального курсу «Теорія програмування» за допомогою мови програмування Java.

Об'єктом розробки в даній роботі є процес дистанційного навчання математичним дисциплінам.

Предметом розробки є тренажер для закріплення знань з теми «Видалення лівої рекурсії» дистанційного навчального курсу «Теорія програмування».

Головне завдання – розробити алгоритм роботи тренажеру та його програмну реалізацію.

Перелік використаних методів – метод видалення лівої рекурсії.

Робота складається з чотирьох розділів. У першому розділі розглянуто постановку задачі тренажеру. У другому розділі описано інформаційний огляд. У третьому розділі ліворекурсивні нетермінальні та видалення лівої рекурсії, розробку алгоритму роботи тренажеру та розробку блок-схеми. У четвертому розділі – описано обґрунтування вибору програмних засобів, розробку тренажеру.

Обсяг пояснювальної записки: 49 стор., в т.ч. основна частина - 41 стор., джерела - 11 назв.

1. ПОСТАНОВКА ЗАДАЧІ

Як у прикладному, так і в фундаментальному плані існують проблеми, вирішення яких потребує об'єднання зусиль науковців та спеціалістів науково-промислового комплексу держави. Ці проблеми пов'язані, насамперед, із досягненням максимальної адекватності, комплексності і життєздатності запропонованих дидактичних, технологічних та технічних рішень, які використовуються в побудові сучасних тренажерів. Вирішення цих питань надасть можливість знайти відповідь на визначення питань побудови того чи іншого виду і типу тренажера.

Комп'ютерний тренажер – автоматизований апаратно-програмний функціонально орієнтований комплекс для навчання людини та відпрацювання певних навичок і вмінь. Тренажери в сучасному розумінні з'явилися, коли виникла необхідність масової підготовки фахівців для роботи або на однотипному обладнанні, або зі схожими робочими діями, а також для військових потреб. Останнім часом, у зв'язку з швидкою комп'ютеризацією світової спільноти, зі створенням найскладнішої техніки, експлуатація якої пов'язана з ризиком для життя не лише однієї людини, але й людства в цілому, виникла ціла індустрія – тренажерні технології, які визначаються як складні комплекси, системи моделювання і симуляції, комп'ютерні програми та фізичні моделі, спеціальні методики, що створюються для того, щоб підготувати особистість до прийняття якісних і швидких рішень.

У сучасних тренажерах та в програмах підготовки і навчання, на них заснованих, закладаються принципи розвитку практичних навичок з одночасною теоретичною підготовкою. Реалізація такого підходу стала можливою у зв'язку з бурхливим розвитком та здешевленням електронно-обчислювальної техніки і прогресом у сфері створення віртуальної реальності. На базі цих технологій розроблено численні тренажери для військового застосування, які надають змогу імітувати бойові дії з найвищою

детальністю в реальному часі, створено безліч додатків технології віртуальної реальності для медицини, що надають можливість проводити операції електронного пацієнту з високим ступенем достовірності, тощо. При цьому сфера застосування тренажерних технологій постійно розширюється.

Тренажерні технології виникли й отримали найбільший розвиток там, де помилки при навчанні на реальних об'єктах можуть призвести до надзвичайних наслідків, а їх усунення – до великих фінансових витрат: у військовій справі, медицині, ліквідації наслідків стихійного лиха, в атомній енергетиці, авіації та космосі. У загальному випадку тренажер являє собою програмно-апаратний комплекс, що має структуру, подану на рисунку.

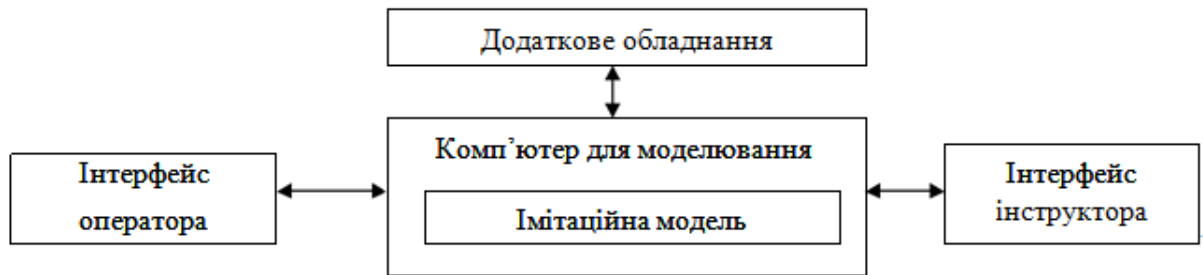


Рисунок 1.1 – Структура комп'ютерного тренажера

Визначимо деякі поняття, які застосовуються при аналізі комп'ютерного тренажера.

Моделюючий комп'ютер – може бути настільки ж простим, як і персональний комп'ютер або таким складним, як багатопроцесорний надсучасний. Він пов'язаний з інтерфейсом оператора через систему введення-виведення. Інтерфейс оператора може складатися як з панелей управління та контролю, так і відеотерміналів та розподіленої системи управління, що обслуговує відеотермінали. У більшості випадків фізичні властивості інтерфейсу оператора точно або в максимально наближеному ступені відповідають конкретному процесу моделювання.

Програмні моделі, що використовуються в імітаційному комп'ютері, реалістично відображають взаємодію компонентів і систем модельованого

процесу. Це найбільш важлива частина тренажерної системи. Від ступеня наближеності імітаційної моделі до реального об'єкта або ситуації залежить якість одержуваних навичок.

Інтерфейс оператора надає тому, хто навчається, можливість маніпулювати органами управління способом, наближеним або ідентичним тому, що використовується в моделюючому комп'ютері. Динамічний відгук тренажера повинен бути максимально наближеним до відгуку систем та компонентів реального об'єкта.

Інтерфейс інструктора надає змогу управляти роботою тренажера, вибирати сценарій тренування та початковий стан імітованого процесу, вводячи збої моделюючого процесу чи його компонентів або змінюючи зовнішні чинники. Частина функцій інструктора може автоматично виконувати і сама імітаційна модель.

Додаткове периферійне устаткування містить у собі принтери, панелі аварійної сигналізації та будь-яке інше обладнання, необхідне для підвищення реалістичності модельованого навколишнього оточення або документування процесу тренування.

Тренажери можуть об'єднуватися між собою в мережу для відпрацювання навичок взаємодії кількох осіб. При цьому може використовуватися загальний моделюючий комп'ютер із декількома інтерфейсами операторів або окремі моделюючі комп'ютери з пристроєм узгодження між ними.

Окремо слід відзначити клас тренажерів, які не використовують спеціальну апаратну інтерфейсну частину. Це суто комп'ютерні тренажери (далі – комп'ютерні тренажери). Роль інтерфейсу в них виконують стандартні пристрої введення-виведення комп'ютера: клавіатура, миша, монітор. Використання таких тренажерів є доцільним у випадках, коли в модельованих об'єктах і ситуаціях немає необхідності у використанні спеціального обладнання. Прикладом можуть бути тренажери з прийняття

рішень та вироблення навичок поведінки, що не пов'язані безпосередньо з управлінням якимись пристроями.

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;
- створення принципово нових зразків має супроводжуватися розробкою методик їх застосування, а також змінами у програмах підготовки.

Аналіз ролі і місця комп'ютерних тренажерів у системі тактичної підготовки, чинників, які впливають на вдосконалення засобів навчання та закономірностей їх розвитку, надає можливість сформулювати найбільш загальні принципи створення перспективних зразків.

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 підготовки;

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. Цей принцип вимагає врахування специфіки діяльності фахівця в процесі навчання. При цьому найбільш складні для засвоєння елементи діяльності (дії, операції), які вимагають великих витрат часу, повинні відпрацьовуватися на пристроях з більшою пропускнуою здатністю;

принцип функціональної гнучкості – елементи тренажеру повинні бути спроектовані з урахуванням можливих змін характеру та змісту професійної діяльності, що відбуваються впродовж терміну експлуатації системи навчання; принцип наступності – при розробці кожного нового засобу, заміни існуючого необхідно враховувати досвід проектування й експлуатації створеного раніше. Новий засіб має без значних змін вписуватися в існуючу систему навчання як методично, так і конструктивно.

Заслуговує на увагу використання інтерактивних тренажерів. Інтерактивні тренажери надають можливість повною мірою зануритися у службовий процес, зімітувати різні випадки та варіанти розвитку подій. Це готове заняття, яке описує конкретне практичне завдання і надає змогу навчитися правильному її вирішенню. Робота проводиться відповідно до реальних подій, що точно відповідають практиці. Інтерактивний тренажер надає можливість домогтися вирішення низки важливих завдань, які не

можуть бути здійснені при використанні інших методів або програм. Кожен тренажер являє собою комплекс операцій, здійснюваних за допомогою програмних засобів. Інтерактивний тренажер складається з трьох етапів:

1. Ознайомлювальне заняття. На цьому занятті користувач знайомиться з основними елементами управління, узнає: навіщо вони потрібні й як ними користуватися. На підставі сценарію заняття розповідається: які саме завдання належить вирішувати користувачеві на тому чи іншому етапі роботи з інформаційною системою. Також робиться акцент на ключових діях, що вимагають від користувача підвищеної уваги та прийняття рішення.

2. Практичне заняття розробляється за сценарієм і містить в собі низку дій, які повинен виконати користувач, щоб вирішити поставлене завдання.

Кожна дія, яку повинен виконати користувач, пояснюється й озвучується. На цьому етапі користувач повинен зрозуміти: що, а головне – навіщо, 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.

Інтерактивний 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.

Інтерактивні тренажери (імітатори програмного забезпечення) можуть бути виконані у форматі окремого додатка (EXE для Microsoft Windows або APP для Apple OS X), SCORM – пакета для вбудовування в будь-яку систему, наприклад, дистанційного навчання, у вигляді окремої web-сторінки (з використанням Adobe Flash або HTML5). Імітація службової діяльності надає великі можливості щодо проведення практичних занять, формування компетенцій, навчання персоналу ДПСУ. Цей вид

навчання є особливо актуальним, коли потрібно навчити слухачів виконувати специфічні дії, які мають місце в оперативно-службовій діяльності. Інтерактивні тренажери надають змогу проводити практичні заняття не лише при очному навчанні, а й дистанційно.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

В сучасних соціально-економічних умовах актуальна концепція безперервного навчання, а диплом про вищу освіту не можна вважати однозначним свідомством наявності знань, необхідних для роботи з спеціальності. Тому для подальшого набуття знань, перепідготовки і підвищення кваліфікації можливість отримання якісної освіти дуже важлива. Оптимальною для цих цілей може стати дистанційна форма навчання.

Дистанційна освіта (ДО) на базі інформаційних технологій має інтернаціональний і глобальний характер. Студенти отримують можливість доступу до великого об'єму текстової, аудіо- та відеоінформації, проходження тестування та представлення виконаних робіт, спілкування між собою на значній відстані.

Дистанційна освіта — це міжнародний термін, який трактується як освіта на відстані та означає цілеспрямоване та методично організоване керівництво навчально-пізнавальною діяльністю осіб, які знаходяться на відстані від навчального закладу та не вступають у постійний контакт з викладацьким складом. На основі цього визначення ДО можна трактувати як самоосвіту, яка передбачає зворотній зв'язок із викладачем (навчальним закладом).

Важливою обставиною при впровадженні дистанційної форми освіти є визначення якості навчання. Під якістю освіти розуміється така організація взаємодії викладача і студентів, тобто навчального процесу, яка відповідає основним принципам концепції навчання, запитам сучасного суспільства та прогнозованій компетенції, якими має володіти випускник навчального закладу, щоб бути конкурентоспроможним фахівцем.

Для надання якісних освітніх послуг, впроваджуючи дистанційну форму навчання, необхідно враховувати специфіку навчального закладу, спеціальності, напрямку, тощо. Це стосується технічних спеціальностей,

освоєння яких потребує проведення великої кількості практичних та лабораторних робіт. Пошук шляхів модернізації методик викладання є запорукою врахування особливостей при впровадженні дистанційного навчання у технічному навчальному закладі. Автор має на меті звернути увагу на деякі організаційні аспекти розробки методичного та програмно-апаратного забезпечення, яке враховуватиме специфіку технічних спеціальностей та використовуватиме для побудови навчального контенту на-явні можливості технічного навчального закладу для підвищення якості надання освітніх послуг за дистанційною формою надання.

Впровадження дистанційного навчання (ДН) у світі набирає високих темпів. Підґрунтям активізації цього процесу є, крім явних переваг та привабливості дистанційної форми навчання, і експертні оцінки згідно з якими:

- у США більш 50 % компаній застосовують у своїй діяльності дистанційну форму навчання,
- все більше вищих навчальних закладів (ВНЗ) надають можливість отримати освіту за дистанційною формою,
- порівняння ефективності дистанційної і очної форм навчання на основі опитування викладачів ВНЗ США показало: 57 % викладачів вважають, що результати ДН не поступаються або перевищують результати традиційних занять, а 33,3 % опитаних вважають, що у найближчий час результати ДН перевищать результати аудиторного,
- економія: на прикладі розробленого компанією IBM та впровадженого дистанційного курсу вартість одного дня навчання скоротилася утричі при збільшенні навчального контенту у п'ять разів.

З точки зору учасників навчального процесу, перевагами дистанційної форми навчання є: географічні та часові переваги, можливість розширення контингенту студентів, розширення об'єму інформації та

підвищення інтенсивності навчання, оптимізація та автоматизація процесу передачі знань, тощо. Щодо недоліків, то це, перш за все, психологічний фактор переходу від традиційної до дистанційної форми навчання.

Розглянемо особливості розробки методичного забезпечення для надання освітніх послуг з технічних спеціальностей за дистанційною формою навчання. Система ДО слідом за розвиненими країнами отримала достатньо широке розповсюдження і в Україні. Разом із тим, якщо проаналізувати перелік спеціальностей, які пропонують дистанційну освіту, то це в основному менеджмент, юриспруденція, економіка, фінанси, тощо. Пояснюється це тим, що при викладанні технічних спеціальностей важливим є не стільки процес передачі освітньої інформації, скільки набуття цілого ряду практичних навичок при виконанні лабораторних та практичних робіт. Ця обставина викликає у професорсько-викладацького складу скептичне ставлення до самої ідеї впровадження дистанційної форми навчання у технічних ВНЗ.

Нові погляди на інженерну освіту, які полягають у розвитку творчого потенціалу майбутнього фахівця, спонукають шукати шляхи модернізації методик викладання в університеті за рахунок застосування інформаційних технологій у процесі навчання студентів. Поширюється тенденція розробки та впровадження у навчальний процес віртуальних тренажерів, лабораторій, тощо. Такий напрямок застосування інформаційних технологій у навчальному процесі актуальний при розробці дистанційних курсів, а також при вирішенні інших освітніх задач. На рис. 2.1 наведена сфера застосування віртуальних лабораторних тренажерів.

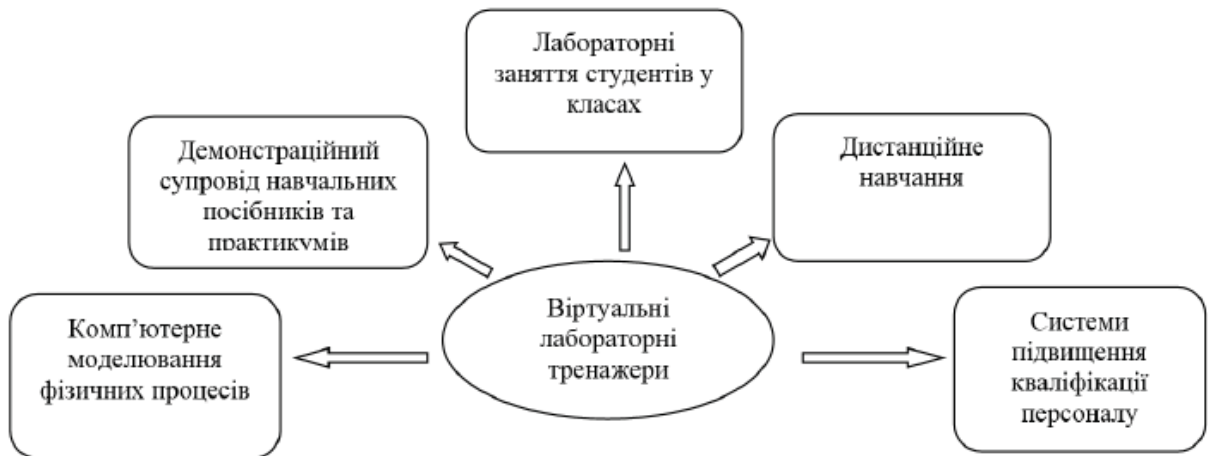


Рисунок 2.1 – Сфера застосування віртуальних лабораторних тренажерів

У цих тренажерах динаміка процесів реалізується за допомогою комп'ютерної анімації — комплексу методів відображення об'єктів у часі. Процеси формування понять за допомогою аналізу, порівняння, виділення істотних ознак і інших логічних операцій відтворюються фахівцем, який розробляє анімацію, в образній формі. Таким чином, динаміка комп'ютерної імітації не лише використовується для показу руху об'єкта, але і розкриває логіку руху думки від незнання до знання. На рис. 2 наведена принципова схема процесу навчання з використанням віртуального тренажера.

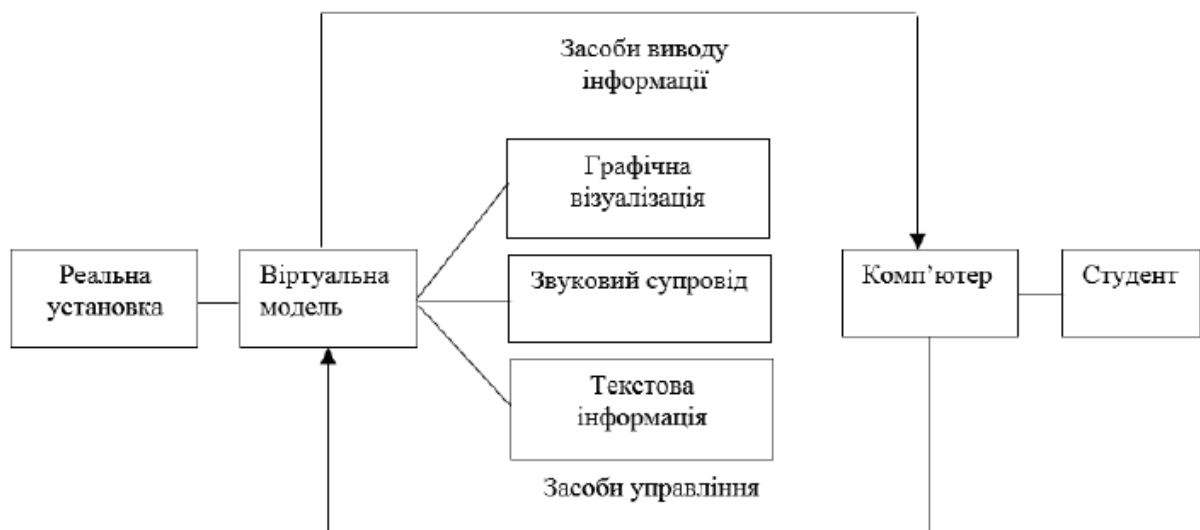


Рисунок 2.2 – Принцип дії віртуального тренажера

Практика застосування віртуальних тренажерів, електронних лабораторій, електронних практикумів для забезпечення надання якісних

освітніх послуг з технічних спеціальностей у системі ДН дозволяє вирішити проблему специфіки викладання технічних дисциплін.

Важливим є очікуваний результат такої роботи. З одного боку, навчальний заклад отримує можливість надання якісних освітніх послуг, отримуючи і використовуючи віртуальний інструментарій у навчальному процесі за дистанційною і не тільки формами навчання. З другого, — студенти, залучені до розробки інструментарію, матимуть можливість отримати науковий і практичний досвід, що, без сумніву, позитивно вплине на якість підготовки. Обидві обставини потенційно підвищують рейтинг університету, залучаючи абітурієнтів як денної, так і заочної та дистанційної форм навчання.

Організація розробки дистанційних курсів та навчального процесу з дисциплін за дистанційною формою навчання є ще одним дуже важливим нерозв'язаним питанням.

Цей аспект крім того, що також сприяє скептичному ставленню професорсько-викладацького складу університету до дистанційної форми навчання, у кінцевому результаті негативно позначається на якості наповнення навчального контенту та здійсненні навчального процесу.

Щодо розробки дистанційного курсу, то найбільш оптимальним був би наступний алгоритм дій. Викладач, використовуючи свій інтелект, знання, досвід, готує у певному вигляді комплект методичних матеріалів з дисципліни, маючи на цей вид діяльності внормований час (з урахуванням об'єму та складності дисципліни), та надає його для програмно-технічної обробки.

Такий підхід має наступні переваги. Викладач не витрачає час на технічну роботу, а зосереджується на науковій та методичній частині роботи над курсом. Технічну обробку текстів курсу виконує технічний персонал, який має досвід, навички, відслідковує новації програмного забезпечення та відповідає за структуру, дизайн і т.д. тексту, що дозволить представити освітній контент упорядкованим, витриманим у одному стилі, що дуже

важливо для комфортного сприйняття студентом методичного матеріалу за звичною формою. Супровід курсу, тобто зміни, правки тощо, мають відбуватись за тим же алгоритмом.

Для здійснення навчального процесу за дистанційною формою навчання прийнято надавати стільки ж часу, скільки і для заочної форми навчання.

Для надання якісних освітніх послуг за дистанційною формою слід враховувати реальні часові витрати на здійснення навчального процесу, що забезпечить гармонійність та комфорт навчання.

ДО розвивається колосальними темпами, цьому сприяє розвиток сучасних інформаційних та телекомунікаційних технологій. В той же час, дистанційні технології, які впроваджуються в освітній процес потребують ретельної дидактичної проробки, відпрацювання методик засвоєння знань, аналізу пріоритетів впливаючих факторів. Ефективність роботи студента та викладача у дистанційному середовищі безпосередньо залежить від науковості та методичної доцільності розроблених прийомів, форм, методів і технологій навчання.

Система ДН може і має посісти своє місце в системі освіти, оскільки при грамотній її організації вона може забезпечити якісну освіту, яка відповідатиме вимогам сучасного суспільства

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Ліворекурсивні нетермінали та видалення лівої рекурсії

Рекурсія є одним з фундаментальних методів програмування. При розробці програми часто необхідно звести початкові завдання до простіших.

Спосіб зведення завдання до нього ж самого, але із зміненими початковими даними, називається **рекурсією**.

Рекурсивний спуск — алгоритм синтаксичного аналізу, будується на основі взаємно рекурсивних процедур (або не рекурсивних еквівалентів), кожна із яких реалізує одну із продукцій граматики.

Обчислення факторіалу $n! = 1 \cdot 2 \cdot 3 \dots n$ зводиться до обчислення добутку цілих чисел від 1 до n : $0! = 1, 1! = 1, 4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$. Такий добуток можна легко обчислити за допомогою циклічних конструкцій, наприклад, оператора циклу `for`. Проте, $n!$ Рівне n , помноженому на добуток від 1 до $(n-1)$, а саме, $n! = (n-1)! \cdot n, 0! = 1$ за означенням $4! = 3! \cdot 4, 6! = 5! \cdot 6$.

В цьому випадку означуване поняття $n!$ описується через саме себе (n) і через простіше поняття $(n-1)!$. Щоб знайти $n!$, треба застосувати функцію факторіала до числа $(n-1)$, потім до $(n-2)$ і т.д. аж до 1. Це і означає, що таке означення факторіалу рекурсивне.

Зміст і потужність рекурсивного визначення, а також його головне призначення полягає в тому, що воно дозволяє за допомогою кінцевого виразу визначити нескінчену множину об'єктів.

Програми, в яких використовують рекурсивні процедури відрізняються простотою, наочністю і компактністю тексту, оскільки рекурсивна процедура указує, що потрібно робити, а нерекурсивна більше акцентує увагу на тому, як потрібно робити. Проте за цю простоту доводиться розплачуватися неекономним використанням оперативної пам'яті, оскільки при кожному рекурсивному виклику для локальних

змінних, а також для параметрів процедури, які передаються за значенням, виділяються нові комірки.

Рекурсивний спуск — алгоритм синтаксичного аналізу, будується на основі взаємно рекурсивних процедур (або не рекурсивних еквівалентів), кожна із яких реалізує одну із продукцій граматики.

Розбір рекурсивним спуском приховує небезпеку зациклювання. Це можливо, якщо продукція має вигляд $\langle \text{вир} \rangle \rightarrow \langle \text{вир} \rangle \langle \text{терм} \rangle$, через те, що нетермінал $\langle \text{вир} \rangle$ ліворекурсивний.

Ліворекурсивним назвемо нетермінал A , якщо в граматиці є продукція виду $A \rightarrow A\alpha$.

Якщо алгоритм обере цю продукцію, то, оскільки права частина починається з $\langle \text{вир} \rangle$, буде знову викликана процедура для нетермінала $\langle \text{вир} \rangle$. Поточний символ, що керує процесом розбору, може змінитися тільки, якщо буде виявлений його збіг з початковим терміналом у правій частині продукції. У нашому випадку продукція починається з нетермінала і поточний символ залишається незмінним, що викликає зациклювання.

Рекурсивні визначення – це такі визначення, в правій частині яких використовується посилання на поняття, що визначається. Такі визначення мають вид $x = \phi(x)$. Рекурсивне визначення можна тлумачити

- операційно, тобто вказати алгоритм, за яким можна обчислити рекурсивно визначений об'єкт;
- або денотаційно, тобто як рівняння, розв'язком якого є нерухомі точки (НТ) оператора ϕ .

Рекурсія широко використовується в мовах програмування. У таких випадках вона, як правило, визначається операційно, тобто вказується алгоритм, за яким можна обчислити рекурсивну процедуру або функцію. Традиційні проблеми, що розглядаються для такого роду рівнянь, є проблеми існування та опису всіх можливих розв'язків, зокрема, формулювання умов єдиності розв'язку. Існують різні методи розв'язку

рекурсивних рівнянь. Найчастіше використовують метод послідовних наближень.

Для того, щоб метод послідовних наближень міг бути застосований, необхідно задати відношення «наближеності» (це – частковий порядок), початкове наближення (у нашому випадку це буде найменший елемент), та гарантувати існування границі (повнота).

Теорії найменших нерухомих точок зазвичай використовується позначення виду $i \in \omega$, де ω – перший нескінченний ординал, тобто $\omega = \{0, 1, 2, \dots\}$ – множина натуральних чисел.

Приклад 1.

$$\begin{aligned} E &\rightarrow E + T \rightarrow + | T, \\ T &\rightarrow T * F | F, \\ F &\rightarrow (E) | id. \\ E &\rightarrow TE', \quad E' \rightarrow + TE' | \varepsilon, \\ T &\rightarrow FT', \quad T' \rightarrow * FT' | \varepsilon, \\ F &\rightarrow (E) | id. \\ A &\rightarrow A\alpha_1 | A\alpha_2 | \dots | A\alpha_m | \beta_1 | \dots | \beta_n \end{aligned}$$

де β_i не починається з A . Потім заміняємо ці A -продукції на

$$\begin{aligned} A &\rightarrow \beta_1 A' | \beta_2 A' | \dots | \beta_n A', \\ A' &\rightarrow \alpha_1 A' | \alpha_2 A' | \dots | \alpha_m A' | \varepsilon. \end{aligned}$$

Нетермінал A породжує ті самі рядки, що і раніше, але тепер немає лівої рекурсії. За допомогою цієї процедури видаляються всі безпосередні ліві рекурсії, але не видаляється ліва рекурсія, що включає два або більше кроків породження.

Задача 1.

$$\begin{aligned} \text{Задана граMATика } G_1 &= (\{ a, \dots, z, 0, \dots, 9 \}, \{ I, L, D \}, \\ &\{ I \otimes L | IL | ID, L \otimes a | \dots | z, D \otimes 0 | \dots | 9 \}, I), \end{aligned}$$

що породжує мові ідентифікаторів.

Видалити безпосередню ліву рекурсію.

$$I \rightarrow IL | ID | L$$

Видалимо рекурсію за формулами

$$I \rightarrow LI'$$

$$I' \rightarrow LT' | DT' | \varepsilon$$

Нові продукції

$$I \rightarrow LI'$$

$$I' \rightarrow LT' | DT' | \varepsilon$$

$$L \rightarrow a | \dots | z, D \rightarrow 0 | \dots | 9.$$

Задача 2.

Розглянемо граматику з продукціями:

$$A \rightarrow Ac | Sd | \varepsilon.$$

$$S \rightarrow Aa | b ;$$

Необхідно:

Видалити безпосередню ліву рекурсію.

За алгоритмом видалити всі ліві рекурсії з граматики.

Безпосередньо:

$$A \rightarrow Ac | Sd | \varepsilon, \text{ тоді}$$

$$A \rightarrow SdA' | \varepsilon A'$$

$$A' \rightarrow cA' | \varepsilon$$

$$S \rightarrow Aa | b$$

За алгоритмом перепозначимо нетермінали:

$$A = A1, \quad A' = A2, \quad S = A3, \text{ тоді}$$

$$A1 = A3dA2 | A2$$

$$A2 = cA2 | \varepsilon$$

$$A3 = A1a | b$$

Видаляємо безпосередньо рекурсію:

$$A_1 \rightarrow A_1 a d A_2 \mid b d A_2 \mid A_2.$$

Замінімо:

$$A_1 \rightarrow b d A_2 A_1' \mid A_2 A_1'$$

$$A_1 \rightarrow a d A_2 A_1' \mid \varepsilon.$$

Перепозначимо назад:

$$A \rightarrow b d A' A_4 \mid A' A_4$$

$$A_4 \rightarrow a d A' A_4 \mid \varepsilon$$

$$A' \rightarrow c A' \mid \varepsilon$$

Предиктивний синтаксичний аналіз – це рекурсивний синтаксичний аналіз без повернень. Це можливо, якщо з граматики видалена ліва рекурсія і вона лівофакторизована.

Рекурсія може бути прихованою (неявною). Для ілюстрації розглянемо парадокс Рассела з теорії множин. А саме, множина x називається нормальною, якщо $x \notin x$.

Позначимо через N множину усіх нормальних множин, тобто $N = \{ x \mid x \notin x \}$. Парадокс виявляється, якщо запитати, чи є N нормальною множиною? Отримуємо, що $N \in N$ тоді і тільки тоді, коли $N \notin N$. Тут рекурсія виступає неявно, бо в визначенні N (неявно) припускається, що N може бути елементом N .

Ліва факторизація

Лівою факторизацією називається перетворення граматики у форму, придатну для предиктивного аналізу.

Основна ідея лівої факторизації полягає у тому, що, коли незрозуміло, яку з двох альтернативних продукцій потрібно використовувати для розгортання не терміналу A , потрібно переробити A -продукції так, щоб

відкласти рішення доти, поки із вхідного потоку не буде прочитано достатньо символів для правильного вибору.

Приклад 2.

Якщо є дві продукції

$$\text{stmt} \rightarrow \text{ifexp } r \text{ then stmt else stmt} \mid \text{ifexp then stmt},$$

то, після знаходження у вхідному потоці *if*, ми не можемо відразу вибрати ні одну з них.

У загальному випадку, якщо $A \rightarrow \alpha\beta_1 \mid \alpha\beta_2$ – дві *A*-продукції і вхідний потік починається з непорожнього рядка, виведеного з α , ми не можемо сказати, буде використовуватися перша чи друга продукція. Однак можна відкласти рішення, розширивши *A* до $\alpha A'$. У цьому випадку, після розгляду вхідного потоку, що виводиться з α , ми працюємо з A' , розширюючи його до β_1 або до β_2 . Таким чином, лівофакторизовані продукції наберуть вигляду:

$$A \rightarrow \alpha A', A' \rightarrow \beta_1 \mid \beta_2.$$

Крок 1. Для кожного *A* шукаємо найдовший префікс α , спільний для двох або більше його альтернатив.

Крок 2. Якщо $\alpha \neq \varepsilon$, тобто існує нетривіальний спільний префікс, заміняємо всі *A*-продукції $\alpha\beta_1 \mid \alpha\beta_2 \mid \dots \mid \alpha\beta_n$ $\gamma A \rightarrow \beta_1 \beta_2 \dots \beta_n$, де γ представляє всі альтернативи, які не починаються з α , продукціями (A' – новий нетермінал):

$$A \rightarrow \alpha \gamma', \quad A \rightarrow \beta_1 \beta_2 \dots \beta_n A'.$$

Повторно застосовуємо це перетворення, поки ніякі дві альтернативи не будуть мати спільний префікс.

$$G = (\{a, b, i, t, e\}, \{E, S\}, P, S)$$

з продукціями:

$$S \rightarrow iEtS \mid iEtSeS \mid a,$$

$$E \rightarrow b.$$

$$S \rightarrow iEtSS' \mid a,$$

$$S' \rightarrow eS \mid \varepsilon,$$

$$E \rightarrow b.$$

3.2. Розробка алгоритму роботи тренажеру

Крок 1.

Користувачу відображається питання: «Що таке рекурсія?».

Наводяться варіанти відповіді, слід вибрати лише один:

А) Спосіб зведення завдання до нього ж самого, але із зміненими початковими даними.

Б) Спосіб зведення завдання до нього ж самого, але із не зміненими початковими даними.

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь.

Правильний варіант – А».

Крок 2.

Користувачу відображається питання: «Рекурсивний спуск – ...».

Наводяться варіанти відповіді, слід вибрати лише один:

А) схема синтаксичного аналізу, будується на основі взаємно рекурсивних процедур (або не рекурсивних еквівалентів), кожна із яких реалізує одну із продукцій граматики.

Б) алгоритм синтаксичного аналізу, будується на основі взаємно рекурсивних процедур (або не рекурсивних еквівалентів), кожна із яких реалізує одну із продукцій граматики.

В) метод синтаксичного аналізу, будується на основі взаємно рекурсивних процедур (або не рекурсивних еквівалентів), кожна із яких реалізує одну із продукцій граматики.

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь.

Правильний варіант – Б».

Крок 3.

Користувачу відображається питання: «Ліва факторизація ...».

Наводяться варіанти відповіді, слід вибрати лише один:

А) називається перетворення граматики у форму, придатну для предиктивного аналізу.

Б) це спосіб зведення завдання до нього ж самого, але із зміненими початковими даними.

В) це поточний символ, що керує процесом розбору.

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь.

Правильний варіант –А».

Крок 4.

Користувачу відображається питання: «Рекурсія є:».

Наводяться варіанти відповіді, слід вибрати лише один:

А) Значення яке можна обчислити.

Б) Одним з фундаментальних методів програмування.

В) Комплексний текст.

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь.

Правильний варіант –А».

Крок 5.

Користувачу відображається питання: «Основна ідея лівої факторизації ...».

Наводяться варіанти відповіді, слід вибрати лише один:

А) переходить у новий стан, який може і не відрізнятися від поточного

Б) передається за значенням, виділяються нові комірочки

В) коли незрозуміло, яку з двох альтернативних продукцій потрібно використовувати для розгортання не термінала А, потрібно переробити А-

продукції так, щоб відкласти рішення доти, поки із вхідного потоку не буде прочитано достатньо символів для правильного вибору.

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь».

Правильний варіант – В».

Крок 6.

Користувачу відображається питання: «Алгоритм – це ...».

Наводяться варіанти відповіді, слід вибрати лише один:

А) список деталізованих інструкцій, що реалізують процес обчислення, який, починаючи з початкового стану, відбувається через послідовність логічних станів, яка завершується кінцевим станом.

Б) набір інструкцій, які описують порядок дій виконавця, щоб досягти результату розв'язання задачі за скінченну кількість дій; система правил виконання дискретного процесу, яка досягає поставленої мети за скінченний час.

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь».

Правильний варіант –Б».

Крок 7.

Користувачу відображається питання: «Предиктивний синтаксичний аналіз – ...».

Наводяться варіанти відповіді, слід вибрати лише один:

А) це рекурсивний синтаксичний аналіз без повернень.

Б) це алгоритм синтаксичного аналізу

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь».

Правильний варіант – А».

Крок 8.

Користувачу відображається умова прикладу: «

Приклад 1

$$E \rightarrow E + T \rightarrow + | T ,$$

$$T \rightarrow T * F | F ,$$

$$F \rightarrow (E) | id .$$

$$E \rightarrow TE' , \quad E' \rightarrow + TE' | \varepsilon ,$$

$$T \rightarrow FT' , \quad T' \rightarrow * FT' | \varepsilon ,$$

$$F \rightarrow (E) | id .$$

$$A \rightarrow A\alpha_1 | A\alpha_2 | \dots | A\alpha_m | \beta_1 | \dots | \beta_n$$

де β_i не починається з A . »

A також питання: «Потім заміняємо ці A -продукції на ...».

Наводяться варіанти відповіді, можна вибрати декілька:

А) $A \rightarrow \beta_1 A' | \beta_2 A' | \dots | \beta_n A'$.

Б) $A \rightarrow \alpha_1 A' | \alpha_2 A' | \dots | \alpha_n A'$.

В) $A' \rightarrow \alpha_1 A' | \alpha_2 A' | \dots | \alpha_m A' | \varepsilon$.

Г) $A' \rightarrow \beta_1 A' | \beta_2 A' | \dots | \beta_m A' | \varepsilon$.

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь.

Правильна відповідь – А і В».

Крок 9.

Користувачу відображається результат: «Нетермінал A породжує ті самі рядки, що і раніше, але тепер немає лівої рекурсії. За допомогою цієї процедури видаляються всі безпосередні ліві рекурсії, але не видаляється ліва рекурсія, що включає два або більше кроків породження.»

Крок 10.

Користувачу відображається умова задачі: «

Задача 1.

Задана граMATика $G_1 = (\{ a, \dots, z, 0, \dots, 9 \}, \{ I, L, D \},$

$$\{ I \rightarrow L \mid IL \mid ID, L \rightarrow a \mid \dots \mid z, D \rightarrow 0 \mid \dots \mid 9 \}, I \},$$

що породжує мові ідентифікаторів.

Видалити безпосередню ліву рекурсію.

$$I \rightarrow IL \mid ID \mid L$$

Видалимо рекурсію за формулами

$$I \rightarrow LI'$$

$$I' \rightarrow LT' \mid DT' \mid \varepsilon.$$

А також питання: «Нові продукції: ...».

Наводяться варіанти відповіді, можна вибрати декілька:

А) $I \rightarrow a \mid \dots \mid z, D \rightarrow 0 \mid \dots \mid 9.$

Б) $I \rightarrow LI'.$

В) $I' \rightarrow LT' \mid DT' \mid \varepsilon.$

Г) $L \rightarrow a \mid \dots \mid z, D \rightarrow 0 \mid \dots \mid 9.$

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь.

Правильна відповідь – Б, В, Г».

Крок 11.

Користувачу відображається умова задачі: «

Задача 2.

Розглянемо граматику з продукціями:

$$A \rightarrow Ac \mid Sd \mid \varepsilon.$$

$$S \rightarrow Aa \mid b;$$

Необхідно:

Видалити безпосередню ліву рекурсію.

За алгоритмом видалити всі ліві рекурсії з граматики.»

А також питання: «Безпосередньо: ...».

Наводяться варіанти відповіді, слід вибрати лише один:

А) $A \rightarrow Ac \mid Sd \mid \varepsilon.$

Б) $A \rightarrow SdA' \mid \varepsilon A'.$

$$\text{В) } A' \rightarrow cA' \mid \varepsilon.$$

$$\text{Г) } S \rightarrow Aa \mid b.$$

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь.

Правильний варіант – А».

Крок 12.

Користувачу відображається питання: «Безпосередньо:

$$A \rightarrow Ac \mid Sd \mid \varepsilon, \text{ тоді ...}».$$

Наводяться варіанти відповіді, слід вибрати лише один:

$$\text{А) } A \rightarrow Ac \mid Sd \mid \varepsilon.$$

$$\text{Б) } A \rightarrow SdA' \mid \varepsilon A'.$$

$$\text{В) } A' \rightarrow cA' \mid \varepsilon.$$

$$\text{Г) } S \rightarrow Aa \mid b.$$

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь.

Правильний варіант – Б».

Крок 13.

Користувачу відображається питання: «Безпосередньо:

$$A \rightarrow Ac \mid Sd \mid \varepsilon,$$

$$A \rightarrow SdA' \mid \varepsilon A', \text{ тоді ...}».$$

Наводяться варіанти відповіді, слід вибрати лише один:

$$\text{А) } A \rightarrow Ac \mid Sd \mid \varepsilon.$$

$$\text{Б) } A \rightarrow SdA' \mid \varepsilon A'.$$

$$\text{В) } A' \rightarrow cA' \mid \varepsilon.$$

$$\text{Г) } S \rightarrow Aa \mid b.$$

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь.

Правильний варіант – В».

Крок 14.

Користувачу відображається питання: «Безпосередньо:

$$A \rightarrow Ac \mid Sd \mid \varepsilon,$$

$$A \rightarrow SdA' \mid \varepsilon A',$$

$$A' \rightarrow cA' \mid \varepsilon, \text{ тоді ...}.$$

Наводяться варіанти відповіді, слід вибрати лише один:

А) $A \rightarrow Ac \mid Sd \mid \varepsilon.$

Б) $A \rightarrow SdA' \mid \varepsilon A'.$

В) $A' \rightarrow cA' \mid \varepsilon.$

Г) $S \rightarrow Aa \mid b.$

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь.

Правильний варіант – Г».

Крок 15.

Користувачу відображається питання: «За алгоритмом перепозначимо нетермінали:

$$A=A1, \quad A'=A2, \quad S=A3, \text{ тоді ...}.$$

Наводяться варіанти відповіді, слід вибрати лише один:

А) $A1 = A3dA2 \mid A2.$

Б) $A1 = c A2 \mid \varepsilon.$

В) $A1 = A1a \mid b.$

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь.

Правильний варіант – А».

Крок 16.

Користувачу відображається питання: «За алгоритмом перепозначимо нетермінали:

$A=A1, \quad A'=A2, \quad S=A3, \text{ тоді ...}.$

Наводяться варіанти відповіді, слід вибрати лише один:

А) $A2 = A3dA2 \mid A2.$

Б) $A2 = c A2 \mid \epsilon.$

В) $A2 = A1a \mid b.$

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь.

Правильний варіант – Б».

Крок 17.

Користувачу відображається питання: «За алгоритмом перепозначимо нетермінали:

$A=A1, \quad A'=A2, \quad S=A3, \text{ тоді ...}.$

Наводяться варіанти відповіді, слід вибрати лише один:

А) $A3 = A3dA2 \mid A2.$

Б) $A3 = c A2 \mid \epsilon.$

В) $A3 = A1a \mid b.$

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь.

Правильний варіант – В».

Крок 18.

Користувачу відображається питання: «Видаляємо пезпосередньо рекурсію:

$A1 \rightarrow A1adA2 \mid bdA2 \mid A2.$

Замінімо: ...».

Наводяться варіанти відповіді, можна вибрати декілька:

А) $A1 \rightarrow bdA2 A1' \mid A2 A1'.$

Б) $A1 \rightarrow adA2 A1' \mid \epsilon.$

В) $A1' \rightarrow adA2 A1 \mid \epsilon.$

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь.

Правильна відповідь – А, Б».

Крок 19.

Користувачу відображається питання: «Перепозначимо назад: ...».

Наводяться варіанти відповіді, можна вибрати декілька:

А) $A \rightarrow bdA'A4 \mid A'A4$.

Б) $A4 \rightarrow adA' A4 \mid \varepsilon$.

В) $A' \rightarrow cA' \mid \varepsilon$.

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь.

Правильна відповідь – всі правильні».

Крок 20.

Користувачу відображається умова прикладу: «

Приклад 2.

Якщо є дві продукції

$stmt \rightarrow if\ expr\ r\ then\ stmt\ else\ stmt \mid if\ expr\ then\ stmt,$

то, після знаходження у вхідному потоці if , ми не можемо відразу вибрати ні одну з них.

У загальному випадку, якщо $A \rightarrow \alpha\beta_1 \mid \alpha\beta_2$ – дві A -продукції і вхідний потік починається з непорожнього рядка, виведеного з α , ми не можемо сказати, буде використовуватися перша чи друга продукція. Однак можна відкласти рішення, розширивши A до $\alpha A'$. У цьому випадку, після розгляду вхідного потоку, що виводиться з α , ми працюємо з A' , розширюючи його до β_1 або до β_2 . Таким чином, лівофакторизовані продукції наберуть вигляду:

$A \rightarrow \alpha A', A' \rightarrow \beta_1 \mid \beta_2$ »

А також питання: «Крок 1. ...».

Наводяться варіанти відповіді, слід вибрати лише один:

А) Для кожного A шукаємо найдовший префікс α , спільний для двох або більше його альтернатив.

Б) Якщо $a \neq \varepsilon$, тобто існує нетривіальний спільний префікс, заміняємо всі A -продукції $\alpha\beta \mid \alpha\beta \mid \dots \alpha\beta \mid \gamma \mid A \rightarrow 1 \ 2 \ n$, де γ представляє всі альтернативи, які не починаються з α , продукціями (A' – новий нетермінал):
 $A \ A \rightarrow \alpha \ \gamma'$, $A \ \beta \ \beta \ \beta \ 1 \ 2 \ n \ ' \rightarrow$.
 Повторно застосовуємо це перетворення, поки ніякі дві альтернативи не будуть мати спільний префікс.

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь».

Правильний варіант – А».

Крок 21.

Користувачу відображається питання: «Крок 2. ...».

Наводяться варіанти відповіді, слід вибрати лише один:

А) Для кожного A шукаємо найдовший префікс α , спільний для двох або більше його альтернатив.

Б) Якщо $a \neq \varepsilon$, тобто існує нетривіальний спільний префікс, заміняємо всі A -продукції $\alpha\beta \mid \alpha\beta \mid \dots \alpha\beta \mid \gamma \mid A \rightarrow 1 \ 2 \ n$, де γ представляє всі альтернативи, які не починаються з α , продукціями (A' – новий нетермінал):
 $A \ A \rightarrow \alpha \ \gamma'$, $A \ \beta \ \beta \ \beta \ 1 \ 2 \ n \ ' \rightarrow$.
 Повторно застосовуємо це перетворення, поки ніякі дві альтернативи не будуть мати спільний префікс.

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку: «Помилка! Перевірте свою відповідь».

Правильний варіант – Б».

Крок 22.

Користувачу відображається результат: «

$$G = (\{a,b,i,t,e\}, \{E, S\}, P, S)$$

з продукціями:

$$S \rightarrow iEtS \mid iEtSeS \mid a ,$$

$$E \rightarrow b.$$

$$S \rightarrow iEtSS' \mid a ,$$

$$S' \rightarrow eS \mid \varepsilon ,$$

$$E \rightarrow b.»$$

3.3. Розробка блок-схеми, яка підлягає програмуванню

На рисунку 3.1 зображено блок-схему алгоритму роботи тренажера.

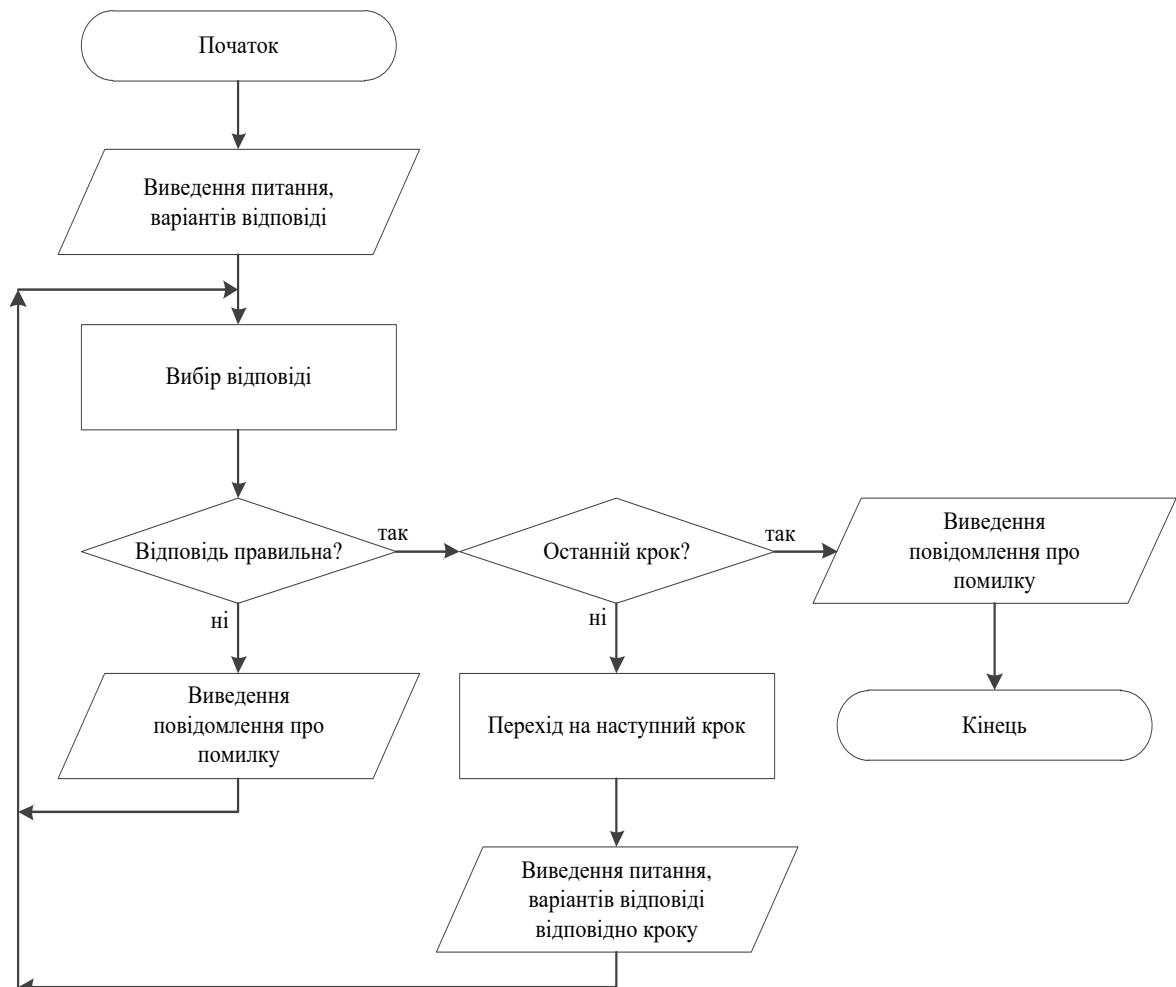


Рисунок 3.1 – Блок-схема алгоритму роботи тренажеру

4. ПРАКТИЧНА ЧАСТИНА

4.1. Обґрунтування вибору програмних засобів

У 1991 році, коли група інженерів з компанії Sun під керівництвом Патріка Нотона і члена ради директорів Джеймса Гослінга зайнялася розробкою невеликої мови, який можна було б використовувати для програмування побутових пристроїв, наприклад, контролерів для перемикачів каналів кабельного телебачення.

Оскільки такі пристрої не споживають багато енергії і не мають великих мікросхем пам'яті, мова повинна бути маленькою і генерувати дуже компактні програми. Крім того, оскільки різні виробники можуть вибирати різні центральні процесори, було важливо не загрузнути в якійсь одній архітектурі комп'ютерів. Проект отримав кодову назву «Green».

Прагнучи винайти невеликий, компактний і машинезалежний код, розробники відродили модель, використану при реалізації перших версій мови Pascal зорі ери персональних комп'ютерів. Ніклаус Вірт, творець мови Pascal, свого часу розробив машинезалежну мову, що генерує проміжний код для якоїсь гіпотетичної машини. Ця мова став комерційним продуктом під назвою UCSD Pascal. (Такі гіпотетичні машини часто називаються віртуальними – наприклад, віртуальна машина мови Java, або JVM.)

Цей проміжний код можна виконувати на будь-якій машині, що має відповідний інтерпретатор. Інженери, що працювали над проектом «Green», також використовували віртуальну машину, що вирішило їх основну проблему.

Отож основною причиною виникнення Java була потреба в мові програмування, яка б не залежала від платформи (тобто від архітектури) і яку можна було б використовувати для створення програмного забезпечення, яке вбудовується в різноманітні побутові електронні прилади, такі як мобільні засоби зв'язку, пристрої дистанційного керування тощо.

JRE (Java Runtime Environment)

Для запуску відкомпільованих програм на Java достатньо мати встановлений набір JRE, що реалізує т. з. віртуальну машину Java (JVM). Для кожної операційної системи створюється власна версія JVM. Один раз створена і відкомпільована, Java-програма гарантовано виконуватиметься на кожній ОС із встановленою JVM. При роботі програми віртуальна машина виконує її т. з. байт-код – проміжний код, який утворюється в результаті компіляції початкового тексту програми. За рахунок цього реалізується концепція кросплатформеності – «написав будь-де, запускай всюди».

JDK (Java Development Kit)

У той час, як для виконання Java-застосунків і аплетів достатньо JRE, для створення програм існує спеціальний пакет JDK, який містить необхідні програми для компіляції, відлагодження і запуску Java-програми, а також набір додаткових утиліт. При встановленні JDK також встановлюються необхідні компоненти віртуальної машини (за необхідності).

Для компілювання Java-програми призначено файл `javac.exe`, який міститься у папці `BIN`, що в папці встановлення JDK. Якщо шлях до місця встановлення JDK змінювали при інсталяції, то компілятор знаходиться у `C:\Program Files\Java\jdk1.8.0_45\bin`.

IDE NetBeans (<http://netbeans.org/>) є однією з найпопулярніших відкритих інструментальних платформ, орієнтованих на розробників ПЗ на Java та інших мовах програмування. До завантаження на сайті доступні різні пакети. Для розробників-початківців підійде базовий набір NetBeans з комплектацією Java SE. Він має модульну структуру, тому у подальшому його можна доповнювати новими інструментами окремо. Пакет передбачає попереднє встановлення Java SDK. Зауважимо, що сам NetBeans також написаний на Java.

Інсталяція IDE NetBeans

Рекомендується розміщати папку встановлення в кореневій папці диску C: (рис. 4.1). Це не є обов'язковим, але дозволить спростити конфігурування системи, особливо при встановленні 32- розрядної версії у 64-розрядній операційній системі.

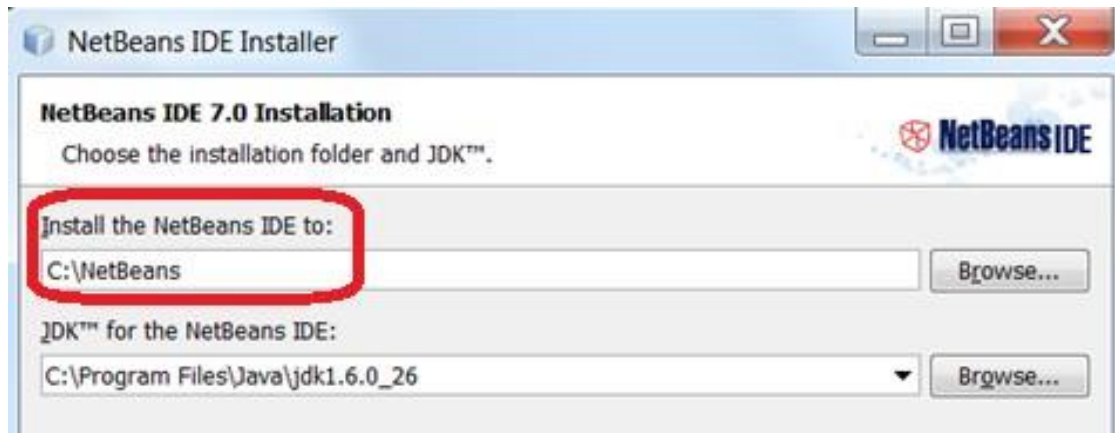


Рисунок 4.1 – Зміна розміщення папки IDE

При першому запуску NetBeans з'явиться стартове привітальне вікно з пропозицією ознайомитися з можливостями платформи та електронними ресурсами співтовариства.

Робота в середовищі NetBeans

Для постійної роботи з IDE необхідно вказати папку, де будуть розміщені програмні проекти розробника.

Проект – пакет файлів, який уособлює програмний продукт і містить файли різного типу:

- тексти програмних модулів;
- відкомпільовані модулі;
- конфігураційні файли проекту;
- файли ресурсів (мультимедіа, файли даних тощо).

IDE типу NetBeans передбачають організацію навіть простої програми у формі проекту.

Створимо новий проект: п.м. File → NewProject... → Java Application (рис. 4.2).

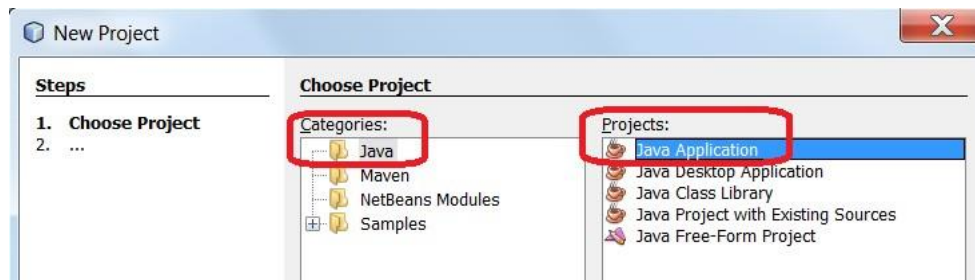


Рисунок 4.2 – Створення проекту Java-застосунку

У момент створення нового проекту пропонується назва проекту і його розміщення за замовченням (рис. 4.3).

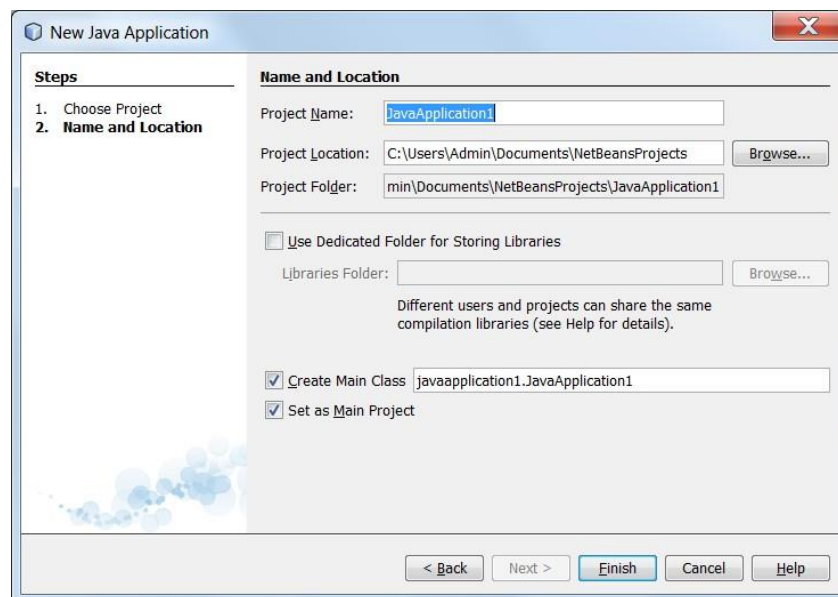


Рисунок 4.3 – Назва і шлях до проекту за замовченням

Шлях до папки розміщення проекту при підтвердженні може бути досить довгим. У випадку встановлення пакету у Windows 7 шлях матиме вигляд типу:

`C:\Users\Admin\Documents\NetBeansProjects\JavaApplication1`

де Admin – ім'я користувача ОС Windows, JavaApplication1 – ім'я проекту (і назва відповідної папки на диску, яка буде створена) за замовченням. Якщо вас це не влаштовує, місце розміщення проектів можна змінити, скориставшись кнопкою Browse..., або відредагувавши вміст поля Project location.

З огляду на ряд причин, доцільно встановити інше розміщення всіх наступних проектів. Нехай, наприклад, всі проекти користувача будуть міститись у папці `C:\NetBeans\projects\`. Створимо папку `projects` у папці `C:\NetBeans\` і вкажемо її при створенні проекту. Назву проекту, зрозуміло, можна змінити. Нехай це буде `MyFirstProject` (рис. 4.4). При створенні наступних проектів місцем їх розміщення буде пропонуватись `C:\NetBeans\projects\`.

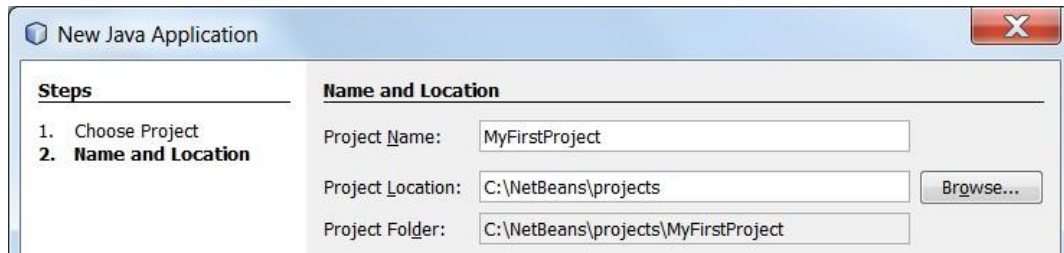


Рисунок 4.4 – Нові назва і розміщення проекту

Після підтвердження відкриється робоча область IDE з допоміжними вікнами навігатора проекту (зліва вгорі), навігатора логічної структури (ієрархії вкладених блоків) зліва внизу, вікна з текстом програми (справа вгорі) та ін. Набір і розміщення видимих вікон IDE можна змінювати за допомогою пункту меню `Window` та перетягуванням мишкою.

Зверніть увагу на відповідність назв елементів у навігаторі проекту, назв ідентифікаторів у тексті програми (рис. 4.5) та структури папок і назв файлів проекту на диску (рис. 4.6). Файли `.java` з текстом програми будуть розміщені у папці `SRC` у папці проекту.

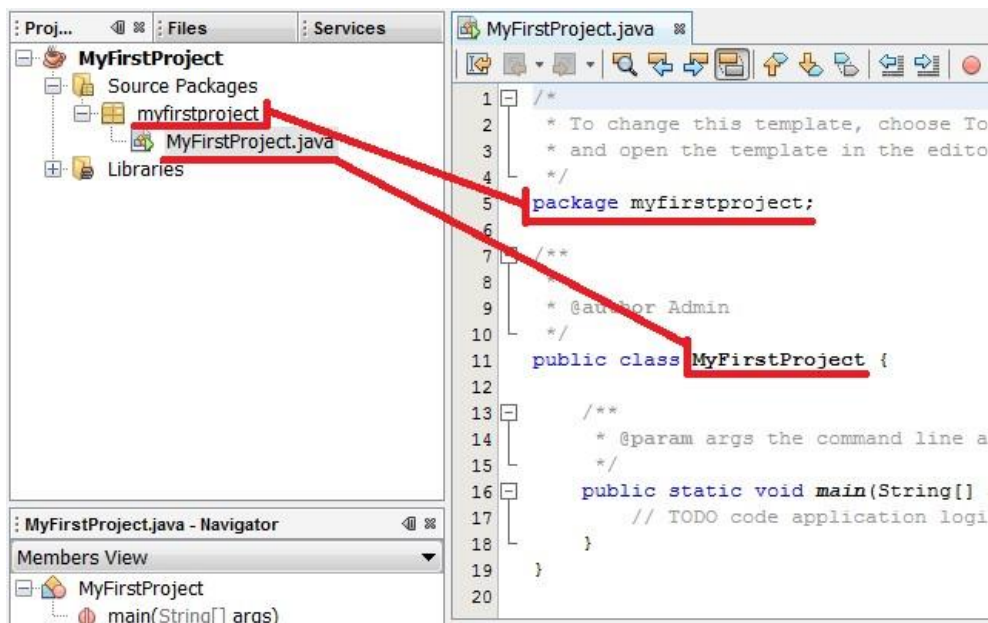


Рисунок 4.5 – Назва пакету і класу в проєкті

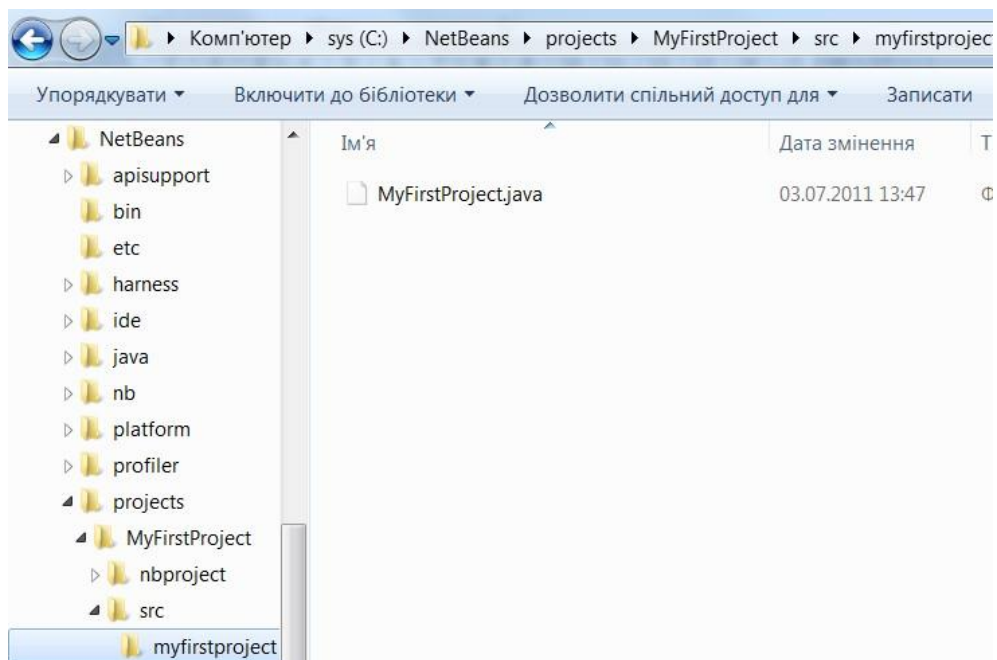


Рисунок 4.6 – Структура папок проєкту і назви файлів

Змінимо текст основної програми на наступний:

Лістинг 1. Програма у середовищі NetBeans

```
package myfirstproject;
```

```
public class pr2_2 {
```

```
    public static void main(String[] args) {
```

```
        System.out.println("Привіт!");
```

```

    }
}

```

Оскільки назва головного класу програми `pr2_2` не співпадає з назвою файлу `MyFirstProject.java`, NetBeans видасть підказку про синтаксичну помилку, суть якої висвітиться при наведенні курсору на червоний маркер на номері рядка програми (рис. 4.7).

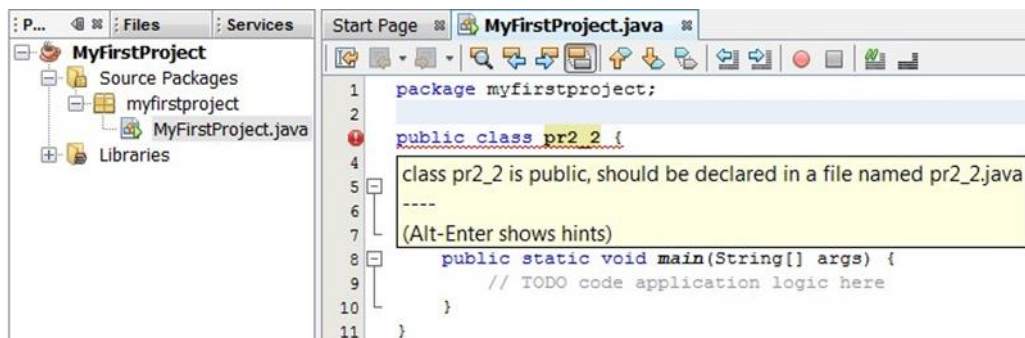


Рисунок 4.7 – Синтаксична помилка: невідповідність імені файлу

Для виправлення помилки слід або змінити ім'я класу (з урахуванням регістру!), або перейменувати файл безпосередньо в навігаторі зліва від вікна з текстом програми. Для виконання останнього зручно скористатися контекстною (правою) кнопкою мишки, клікнувши на імені файлу, а потім – використати п.м. Properties або Refactor → Rename.

Для компіляції і запуску програми скористайтесь п.м. Run → Run Main Project або клавішею F6, чи кнопкою на панелі інструментів.

4.2. Розробка тренажера

Створено зміню `int a`, що вказує поточний крок.

```
int a=0;
```

Саме питання і варіанти відповіді виводяться наступним чином:

```
System.out.println("Що таке рекурсія:");
```

```
System.out.println("А) Спосіб зведення завдання до нього ж самого,  
але із зміненими початковими даними");
```

```
System.out.println("Б) Спосіб зведення завдання до нього ж самого,  
але із не зміненими початковими даними");
```

```
System.out.print("Виберіть правильну відповідь: ");
```

Потім відбувається перевірка вибраної відповіді. Якщо правильно, то перехід до наступного.

```
String b=Dip.readLine();
```

```
if (b.equals("A")){
```

```
  a=a+1;
```

```
}
```

ВИСНОВКИ

Метою виконаної роботи є розроблений тренажер з теми «Видалення лівої рекурсії» дистанційного навчального курсу «Теорія програмування» за допомогою мови програмування Java.

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;
- принцип узгодженості;
- принцип достатності;
- принцип безперервності;
- принцип функціональної гнучкості.

Заслуговує на увагу використання інтерактивних тренажерів. Інтерактивний тренажер складається з трьох етапів:

1. Ознайомлювальне заняття.
2. Практичне заняття.
3. Тестування.

В алгоритмі користувачу відображається умова задачі, а також питання. Наводяться варіанти відповіді.

Якщо відповідь правильна, тоді виконується перехід на наступний крок. Якщо ні, то виводиться повідомлення про помилку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ємець О. О. Методичні рекомендації до виконання бакалаврської роботи для студентів за освітньою програмою «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки та інформаційні технології» галузь знань - 12 «Інформаційні технології» / О.О.(Олег) Ємець. – Полтава : РВВ ПУЕТ, 2017. – 71 с.
2. Гуревич, Р. С. Інформаційно-телекомунікаційні технології в навчальному процесі та наукових дослідженнях / Р. С. Гуревич, М. Ю. Кадемія. – К. : Освіта України, 2006. – 390 с.
3. Матвієвський, О. М. Методичний підхід до обґрунтування характеристик тренажних засобів і систем / О. М. Матвієвський, О. В. Герасименко, Ю. М. Щєбланін // Наука і оборона. – 2005. – № 1. – С. 59–65.
4. Бабій М.С. Теорія програмування: Навчальний посібник [Електронний ресурс] / М.С. Бабій, О.П. Чекалов.– Суми: Вид-во СумДУ, 2009. – 181 с.
5. Зыков С.В. Современные языки программирования. Ч.1. Функциональный подход к программированию / С.В. Зыков. – М.: МИФИ, 2003. – 230 с.
6. Бондаренко М.Ф. Комп'ютерна дискретна математика: Підручник / М.Ф. Бондаренко, Н.В. Білоус, А.Г. Руткас. – Харків: «Компанія СМІТ», 2004. – 408 с.
7. Барендрегт Х. Ламбда-исчисление. Его синтаксис и семантика / Х. Барендрегт. – М.: Мир, 1985. – 606 с.
8. Гросс М. Теория формальных грамматик / М. Гросс, А. Лантен. – М.: Мир, 1971. – 294 с.
9. Пентус А. Е. Теория формальных языков: Учебное пособие / А. Е. Пентус, М.Р. Пентус.– М.: Изд-во ЦПИ при механико-математическом ф-те МГУ, 2004. – 80 с.

10. Черненко О.О. Електронний навчально-методичний посібник для самостійного вивчення навчальної дисципліни «Теорія програмування» для студентів напрямку 6.040302 «Інформатика» – Режим доступу: <http://tprogr.ho.ua>.
11. 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. КОД ПРОГРАМИ