

**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

**Алгоритмізація та програмування тренажера «Регулярні операції над
мовами» дистанційного навчального курсу «Теорія програмування»**

зі спеціальності 122 «Комп'ютерні науки»

Виконавець роботи Дідик Віктор Васильович

_____ « _____ » _____ 2021р.
(підпис)

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

_____ « _____ » _____ 2021р.
(підпис)

ПОЛТАВА 2021 р.

ЗМІСТ

ВСТУП	3
1. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	5
2. ТЕОРЕТИЧНА ЧАСТИНА	8
2.1. Огляд матеріалу з теми.....	8
2.2. Алгоритмізація задачі за темою роботи	11
2.3. Розробка блок-схеми, яка підлягає програмуванню	18
3. ПРАКТИЧНА ЧАСТИНА	21
3.1. Обґрунтування вибору програмних засобів для реалізації завдання роботи.....	21
3.2. Опис процесу програмної реалізації	25
ВИСНОВКИ.....	28
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	30
ДОДАТОК А.....	31

ВСТУП

Будь-який математичний тренажер заснований на принципі формування практичних навичок з одночасною теоретичною підготовкою.

Психологами давно встановлено, що розвиток тактильної пам'яті сприяє розвитку пам'яті, наочно-образного мислення, уяви. А адже саме ці якості сприяють успішному освоєнню дисципліни.

Вважаємо наш тренажер перспективним, так він розвиває не тільки математичні навички, а й пам'ять, через тактильну взаємодію.

Актуальність проекту очевидна – за допомогою розробки нового інтерактивного тренажера підвищити якість успішності, створити свій проект навчання з захопленням.

Актуальність дослідження визначається необхідністю розробки засобів управління навчально-пізнавальною діяльністю студентів, що діагностують не тільки результат, але і весь процес діяльності випробуваного при виконанні завдань, що дають можливість викладачу спостерігати процес вирішення завдання.

Метою роботи є алгоритмізація та програмування тренажера «Регулярні операції над мовами» дистанційного навчального курсу «Теорія програмування».

Об'єктом розробки є математичний тренажер.

Предметом розробки є елементи тренажера з теми «Регулярні операції над мовами».

Перелік використаних методів:

- вивчення теоретичного матеріалу по різновидам тренажерів;
- розробка макета тренажера;
- апробація тренажера;
- аналіз отриманих даних.

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

Робота складається з трьох розділів.

Обсяг пояснювальної записки: 31 стор., в т.ч. основна частина - 29 стор., джерела - 8 назв.

1. ІНФОРМАЦІЙНИЙ ОГЛЯД

Інформаційно-комунікаційні технології або ІКТ — засоби, пов'язані зі створенням, збереженням, передачею, обробкою і управлінням інформації. Цей широко вживаний термін включає в себе всі технології, що використовуються для спілкування та роботи з інформацією.

Будь-яка педагогічна технологія - це інформаційна технологія, оскільки основу технологічного процесу навчання складає отримання і перетворення інформації.

Сучасні ІКТ в освітньо-інформаційному середовищі навчальних закладів мають виконувати такі функції:

- засобу навчання (застосування мультимедійних навчальних курсів);
- технічного засобу автоматизації процесу навчальної діяльності, що включена у пізнавальну, пошукову, дослідницьку, експериментальну роботу, який дозволяє мобільно працювати з текстом, графічним, звуковим або відеодокументом, якісно подавати інформацію, обробляти її;
- зразка сучасних інноваційних технологій, що розвиває навички оволодіння ними, дає знання про їхні назви і функціональне призначення, складові елементи;
- ефективного тренажера, що розвиває.

Використання комп'ютерних технологій - це не вплив моди, а необхідність, продиктована сьогодишнім рівнем розвитку освіти. Переваги використання ІКТ можна звести до двох груп: технічні і дидактичні.

Технічними перевагами є швидкість, маневреність, оперативність, можливість перегляду і прослуховування фрагментів і інші мультимедійні функції.

Інформаційні технології здатні:

- стимулювати пізнавальний інтерес,

- надати навчально-виховній роботі проблемний, творчий, дослідницький характер і розвивати самостійну діяльність,
- забезпечувати навички діяльності стосовно інформації, що міститься в навчальних предметах і освітніх областях,
- за допомогою реальних об'єктів (телевізор, магнітофон, комп'ютер) та інформаційних технологій (відеозапис, ЗМІ, Інтернет), формувати вміння самостійно шукати, аналізувати і відбирати необхідну інформацію, організовувати, перетворювати, зберігати і передавати її,

Використання ІКТ у роботі педагога дає можливість:

- самоосвіти педагога, підвищення його професійного рівня;
- навчатися і здобувати знання, вміння та якості, необхідні сучасній людині;
- отримувати найсучаснішу інформацію по темі, оновлювати навчальний та дидактичний матеріал;
- мати доступ до методичної бази розробок;
- спілкуватися з колегами на різних форумах;
- отримувати кваліфіковані консультації та поради експертів;
- публікувати свої матеріали;
- брати участь в обговоренні опублікованих матеріалів;
- брати участь в професійних конкурсах;
- обмінюватися досвідом з колегами з інших регіонів і країн.

Для ефективного впровадження ІКТ в освітньо-виховний процес вимагається володіння комп'ютерною грамотністю, яка передбачає:

- вміння писати та редагувати інформацію (текстову, графічну);
- користуватися комп'ютерною телекомунікаційною технологією;
- користуватися базами даних;
- роздруковувати інформацію на принтері;
- вміння скласти та надіслати лист через мережу Інтернет;

- вміння «перекачати» інформацію з мережі на жорсткий або гнучкий диск і навпаки, з жорсткого або гнучкого диска - в мережу;
- працювати в системах DOC і WINDOWS, користуючись редакторами WORD різної модифікації;
- входити в електронні конференції, розміщувати там власну інформацію і читати, «перекачувати» наявну в різних конференціях інформацію.

Систематичне включення інформаційно-комунікаційних технологій в освітній процес забезпечить формування і розвиток інформаційно-комунікаційної культури педагогічних працівників та вихованців [2].

2. ТЕОРЕТИЧНА ЧАСТИНА

2.1. Огляд матеріалу з теми

Над мовами можливі операції, що породжують нові мови. Означимо *регулярні операції* над мовами: об'єднання, катенацію та ітерацію. Нехай L_1 , L_2 , L позначають довільні мови в алфавіті X . Перелік основних операцій наводиться у табл. 2.1 [3].

Таблиця 2.1 – Операції над мовами

Операція	Визначення
Об'єднання L_1 і L_2 : $L_1 \cup L_2$	$L_1 \cup L_2 =$ $\{w \mid w \in L_1 \text{ або } w \in L_2\}$.
Катенація L_1 і L_2 : $L_1 L_2$	$L_1 L_2 = \{vw \mid v \in L_1, w \in L_2\}$.
Замикання Кліні, або ітерація L : L^* .	$L^* = \bigcup_{i=0}^{\infty} L^i = \{w^i \mid w \in L \text{ за } i \geq 0\}$ L^* означає 0 або більше конкатенацій L .
Позитивне замикання L : L^+ .	$L^+ = \bigcup_{i=1}^{\infty} L^i = \{w^i \mid w \in L \text{ за } i > 0\}$ L^+ означає одну або більше конкатенацій L .

Вираз $L_1 \cup L_2$ позначає *об'єднання* L_1 і L_2 – мову $\{w \mid w \in L_1 \text{ або } w \in L_2\}$. Наприклад, $\{a, ab\} \cup \{a, b, ba\} = \{a, b, ab, ba\}$.

Катенацією слів v і w називається дописування w після v : vw . Вираз L_1L_2 позначає **катенацію мов** – мову $\{vw \mid v \in L_1, w \in L_2\}$. Так, за $L_1 = \{a, bc\}, L_2 = \{x, y\}$ катенація $L_1L_2 = \{ax, bcx, ay, bcy\}$, за $L_1 = \{a, ab\}, L_2 = \{\varepsilon, b\}$ катенація $L_1L_2 = \{a, ab, abb\}$.

Від катенації походить **піднесення до степеня**: $L^0 = \{\varepsilon\}, L^i = L^{i-1}L$ за $i > 0$. Так, вираз $\{\varepsilon, a, aa\}^2$ задає мову $\{\varepsilon, a, aa, aaaa\}$.

Вираз L^* позначає **ітерацію** мови L – мову $\{w^i \mid w \in L \text{ за } i \geq 0\}$, тобто $\{\varepsilon\} \cup L \cup L^2 \cup \dots$. Зазначимо, що ітерація не подається жодним скінченним виразом з операціями катенації та \cup і тому *не є похідною від них*. Якщо в мові L є непорожнє слово, то мова L^* нескінченна. Наприклад, вираз $\{ab\}^*$ задає мову $\{\varepsilon, ab, abab, ababab, \dots\}, \{a, b\}\{a, b, 1\}^*$ – множину ідентифікаторів у алфавіті $\{a, b, 1\}$ [4].

Регулярні вирази й задані ними регулярні мови означимо індуктивно. Вирази \emptyset , ε та a при $a \in X$ є **регулярними** в алфавіті X і задають відповідно **регулярні мови** $\emptyset, \{\varepsilon\}, \{a\}$. Якщо r_1 і r_2 – регулярні вирази, що задають регулярні мови L_1 і L_2 , то вирази $(r_1), r_1 + r_2, r_1r_2, r_1^*$ є регулярними й задають відповідно регулярні мови $L_1, L_1 \cup L_2, L_1L_2, L_1^*$.

Очевидно, що кожна скінченна мова є регулярною, оскільки задається регулярним виразом як скінченне об'єднання одноелементних регулярних мов.

Множина регулярних мов, заданих усіма можливими регулярними виразами в алфавіті X , називається **класом регулярних мов над X** .

Регулярні операції застосовні до будь-яких мов, а не тільки до регулярних. За означенням, застосування їх до регулярних мов породжує регулярні мови.

Не всі мови є регулярними. Наприклад, "мова вкладених дужок", задана БНФ

$\langle \text{вкл} - \text{дуж} \rangle ::= () | (\langle \text{вкл} - \text{дуж} \rangle)$, є множиною $\{(^n)^n \mid n > 0\}$, яка не задається жодним скінченим регулярним виразом (доведення можна знайти в [AV]). Отже, розглянемо інші, потужніші засоби задання мов.

Нехай L – множина букв $\{A, \dots, z\}$, а D – множина $\{0, \dots, 9\}$.

Прикладами мов можуть бути:

$L \cup D$ – множина букв і цифр;

LD – множина рядків з букви і цифри;

L^4 – множина чотирибуквених рядків;

L^* – множина усіх рядків з букв, включаючи ε ;

$L(L \cup D)^*$ – множина усіх рядків з букв і цифр, що починаються з букви;

D^+ – множина усіх рядків з однієї або декількох цифр.

Найпростішу мову можна описати регулярним виразом, що будується із символів за допомогою конкатенації, об'єднання і повторення.

У той самий час мову $\{x^n y^n \mid n > 0\}$ неможливо описати регулярним виразом, оскільки в регулярних виразах немає засобів указати, що кількість x повинна дорівнювати кількості y .

Більш могутній механізм – використання продукцій, для даного випадку вони виглядають так:

$S \rightarrow xSy, S \rightarrow xy$.

Символ \rightarrow читається, як “може мати вигляд”.

Генерація рядків мови:

- S замінюється правою частиною продукції;
- якщо отриманий рядок не містить більше символів S , він є рядком мови.

Приклад послідовності рядків

S

xSy

$xxSyu$

$xxhuuu$.

Як правило, послідовність записують так: $S \Rightarrow xSy \Rightarrow xxSyu \Rightarrow xxhuuu$

Знак \Rightarrow читається як “породжує”.

Послідовність кроків генерації рядка із застосуванням продукцій називається **породженням** (derivation) рядка [3].

2.2. Алгоритмізація задачі за темою роботи

В алгоритмі всі правильні відповіді позначено знаком «+» або вказується правильна відповідь у дужках.

Крок 0. Виводиться тема тренажера, прізвище та ім'я розробника, академічна група. Пропонується перейти до тестів.

Крок 1. Виводиться завдання (в дужках вказано пусті комірки, що потрібно заповнити):

Вираз (1) позначає **об'єднання** L_1 і L_2 – мову (2).

Доступні варіанти для (1):

- L_1L_2
- $L_1 \cup L_2$ (+)

Доступні варіанти для (2):

- $\{w \mid w \in L_1 \text{ або } w \in L_2\}$ (+)
- $\{vw \mid v \in L_1, w \in L_2\}$

При помилці вказується відповідне повідомлення: «Відповідь неправильна. За потреби зверніться до довідки». Якщо все вказано правильно, то відображається наступний крок.

Крок 2. Вказується завдання (в дужках вказано пусті комірки, що потрібно заповнити):

Наприклад, $\{a, ab\} \cup \{a, b, ba\} = (1)$.

Комірка(-и) заповнюється вручну. ($\{a, b, ab, ba\}$)

При помилці вказується відповідне повідомлення: «Відповідь неправильна. За потреби зверніться до довідки». Якщо все вказано правильно, то відображається наступний крок.

Крок 3. Виводиться завдання (в дужках вказано пусті комірки, що потрібно заповнити):

Катенацією слів v і w називається дописування w після v : vw . Вираз (1) позначає **катенацію мов** – мову (2).

Доступні варіанти для (1):

- L_1L_2 (+)
- $L_1 \cup L_2$

Доступні варіанти для (2):

- $\{w \mid w \in L_1 \text{ або } w \in L_2\}$
- $\{vw \mid v \in L_1, w \in L_2\}$ (+)

При помилці вказується відповідне повідомлення: «Відповідь неправильна. За потреби зверніться до довідки». Якщо все вказано правильно, то відображається наступний крок.

Крок 4. Вказується завдання (в дужках вказано пусті комірки, що потрібно заповнити):

Так, за $L_1 = \{a, bc\}, L_2 = \{x, y\}$ катенація $L_1L_2 = (1)$.

Комірка(-и) заповнюється вручну. ($\{ax, bcx, ay, bcy\}$)

При помилці вказується відповідне повідомлення: «Відповідь неправильна. За потреби зверніться до довідки». Якщо все вказано правильно, то відображається наступний крок.

Крок 5. Вказується завдання (в дужках вказано пусті комірки, що потрібно заповнити):

Так, за $L_1 = \{a, ab\}, L_2 = \{\varepsilon, b\}$ катенація $L_1L_2 = (1)$.

Комірка(-и) заповнюється вручну. ($\{a, ab, abb\}$)

При помилці вказується відповідне повідомлення: «Відповідь неправильна. За потреби зверніться до довідки». Якщо все вказано правильно, то відображається наступний крок.

Крок 6. Виводиться завдання (в дужках вказано пусті комірки, що потрібно заповнити):

Від катенації походить *піднесення до степеня*: $L^0 = (1), L^i = (2)$ за (3).

Доступні варіанти для (1) і (2):

- $\{\varepsilon\}$ (+ для 1)
- L
- $L^{i-1}L$ (+ для 2)

Доступні варіанти для (3):

- $i > 0$ (+)
- $i \geq 0$
- $i = 0$

При помилці вказується відповідне повідомлення: «Відповідь неправильна. За потреби зверніться до довідки». Якщо все вказано правильно, то відображається наступний крок.

Крок 7. Вказується завдання (в дужках вказано пусті комірки, що потрібно заповнити):

Так, вираз $\{\varepsilon, a, aa\}^2$ задає мову (1).

Комірка(-и) заповнюється вручну. Замість символу «ε» слід вказувати літеру «e». ($\{\varepsilon, a, aa, aaa, aaaa\}$)

При помилці вказується відповідне повідомлення: «Відповідь неправильна. За потреби зверніться до довідки». Якщо все вказано правильно, то відображається наступний крок.

Крок 8. Виводиться завдання (в дужках вказано пусті комірки, що потрібно заповнити):

Вираз (1) позначає *ітерацію* мови L – мову (2), тобто $\{\varepsilon\} \cup L \cup L^2 \cup \dots$

Доступні варіанти для (1):

- L^+
- L^* (+)

Доступні варіанти для (2):

- $\{w^i \mid w \in L \text{ за } i > 0\}$
- $\{w^i \mid w \in L \text{ за } i \geq 0\}$ (+)

При помилці вказується відповідне повідомлення: «Відповідь неправильна. За потреби зверніться до довідки». Якщо все вказано правильно, то відображається наступний крок.

Крок 9. Вказується завдання (в дужках вказано пусті комірки, що потрібно заповнити):

Зазначимо, що ітерація не подається жодним скінченим виразом з операціями катенації та \cup і тому (1).

Доступні варіанти для (1):

- не є похідною від них (+)
- є похідною від них

При помилці вказується відповідне повідомлення: «Відповідь неправильна. За потреби зверніться до довідки». Якщо все вказано правильно, то відображається наступний крок.

Крок 10. Вказується завдання (в дужках вказано пусті комірки, що потрібно заповнити):

Якщо в мові L є непорожнє слово, то мова L^* (1).

Доступні варіанти для (1):

- нескінченна (+)
- скінченна

При помилці вказується відповідне повідомлення: «Відповідь неправильна. За потреби зверніться до довідки». Якщо все вказано правильно, то відображається наступний крок.

Крок 11. Вказується завдання (в дужках вказано пусті комірки, що потрібно заповнити):

Наприклад, вираз $\{ab\}^*$ задає мову (1), $\{a,b\}\{a,b,1\}^*$ – (2).

Доступні варіанти:

- множину ідентифікаторів у алфавіті $\{a,b,1\}$ (для 2)
- $\{\varepsilon, ab, abab, ababab, \dots\}$ (для 1)

При помилці вказується відповідне повідомлення: «Відповідь неправильна. За потреби зверніться до довідки». Якщо все вказано правильно, то відображається наступний крок.

Крок 12. Виводиться завдання (в дужках вказано пусті комірки, що потрібно заповнити):

Вираз (1) позначає позитивне замикання мови L – мову (2), тобто $L \cup L^2 \cup \dots$

Доступні варіанти для (1):

- L^+ (+)
- L^*

Доступні варіанти для (2):

- $\{w^i \mid w \in L \text{ за } i > 0\}$ (+)
- $\{w^i \mid w \in L \text{ за } i \geq 0\}$

При помилці вказується відповідне повідомлення: «Відповідь неправильна. За потреби зверніться до довідки». Якщо все вказано правильно, то відображається наступний крок.

Крок 13. Виводиться завдання (в дужках вказано пусті комірки, що потрібно заповнити):

Регулярні операції застосовні (1). За означенням, застосування їх до регулярних мов породжує регулярні мови.

Доступні варіанти для (1):

- не до будь-яких мов, а тільки до регулярних
- до будь-яких мов, а не тільки до регулярних (+)

При помилці вказується відповідне повідомлення: «Відповідь неправильна. За потреби зверніться до довідки». Якщо все вказано правильно, то відображається наступний крок.

Крок 14. Виводиться завдання (в дужках вказано пусті комірки, що потрібно заповнити):

Нехай L – множина букв $\{A, \dots, z\}$, а D – множина $\{0, \dots, 9\}$.

Прикладами мов можуть бути: (1) – множина букв і цифр.

Доступні варіанти для (1):

- LD
- $L \cup D$ (+)
- $L(L \cup D)^*$

При помилці вказується відповідне повідомлення: «Відповідь неправильна. За потреби зверніться до довідки». Якщо все вказано правильно, то відображається наступний крок.

Крок 15. Виводиться завдання (в дужках вказано пусті комірки, що потрібно заповнити):

Нехай L – множина букв $\{A, \dots, z\}$, а D – множина $\{0, \dots, 9\}$.

Прикладами мов можуть бути: (1) – множина рядків з букви і цифри.

Доступні варіанти для (1):

- LD (+)
- $L \cup D$
- $L(L \cup D)^*$

При помилці вказується відповідне повідомлення: «Відповідь неправильна. За потреби зверніться до довідки». Якщо все вказано правильно, то відображається наступний крок.

Крок 16. Виводиться завдання (в дужках вказано пусті комірки, що потрібно заповнити):

Нехай L – множина букв $\{A, \dots, z\}$, а D – множина $\{0, \dots, 9\}$. Прикладами мов можуть бути: (1) – множина усіх рядків з букв і цифр, що починаються з букви.

Доступні варіанти для (1):

- LD
- $L \cup D$
- $L(L \cup D)^*$ (+)

При помилці вказується відповідне повідомлення: «Відповідь неправильна. За потреби зверніться до довідки». Якщо все вказано правильно, то відображається наступний крок.

Крок 17. Виводиться завдання (в дужках вказано пусті комірки, що потрібно заповнити):

Нехай L – множина букв $\{A, \dots, z\}$, а D – множина $\{0, \dots, 9\}$. Прикладами мов можуть бути: (1) – множина чотирибуквених рядків.

Доступні варіанти для (1):

- D^+
- L^*
- L^4 (+)

При помилці вказується відповідне повідомлення: «Відповідь неправильна. За потреби зверніться до довідки». Якщо все вказано правильно, то відображається наступний крок.

Крок 18. Виводиться завдання (в дужках вказано пусті комірки, що потрібно заповнити):

Нехай L – множина букв $\{A, \dots, z\}$, а D – множина $\{0, \dots, 9\}$.
 Прикладами мов можуть бути: (1) – множина усіх рядків з букв, включаючи ε .

Доступні варіанти для (1):

- D^+
- L^* (+)
- L^4

При помилці вказується відповідне повідомлення: «Відповідь неправильна. За потреби зверніться до довідки». Якщо все вказано правильно, то відображається наступний крок.

Крок 19. Виводиться завдання (в дужках вказано пусті комірки, що потрібно заповнити):

Нехай L – множина букв $\{A, \dots, z\}$, а D – множина $\{0, \dots, 9\}$.
 Прикладами мов можуть бути: (1) – множина усіх рядків з однієї або декількох цифр.

Доступні варіанти для (1):

- D^+ (+)
- L^*
- L^4

При помилці вказується відповідне повідомлення: «Відповідь неправильна. За потреби зверніться до довідки». Якщо все вказано правильно, то відображається наступний крок.

Крок 20. Виводиться результат. Пропонується завершити роботу.

2.3. Розробка блок-схеми, яка підлягає програмуванню

На рисунку 2.1 зображено блок-схему алгоритму роботи тренажера.

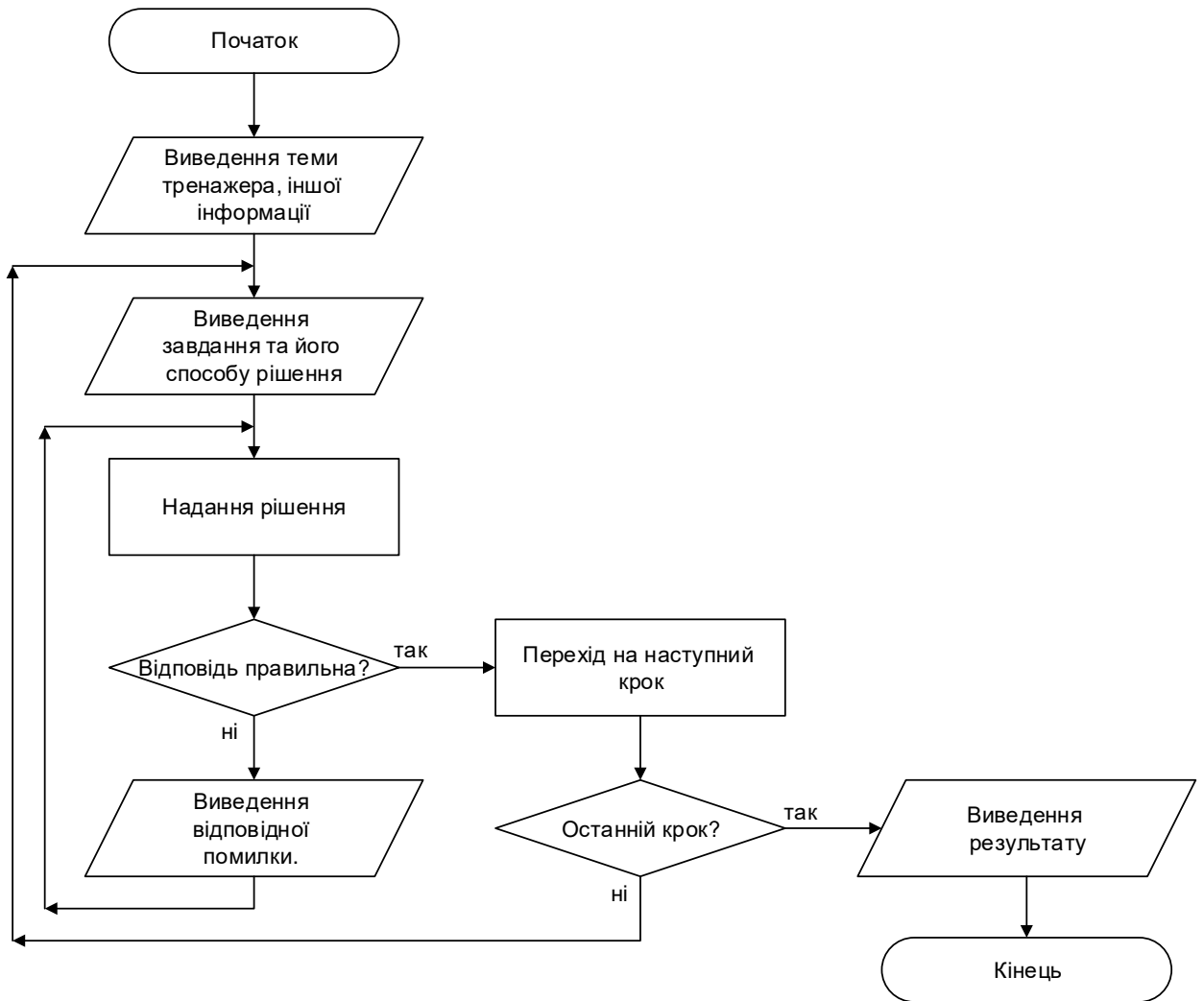


Рисунок 2.1 – Блок-схема алгоритму роботи тренажера

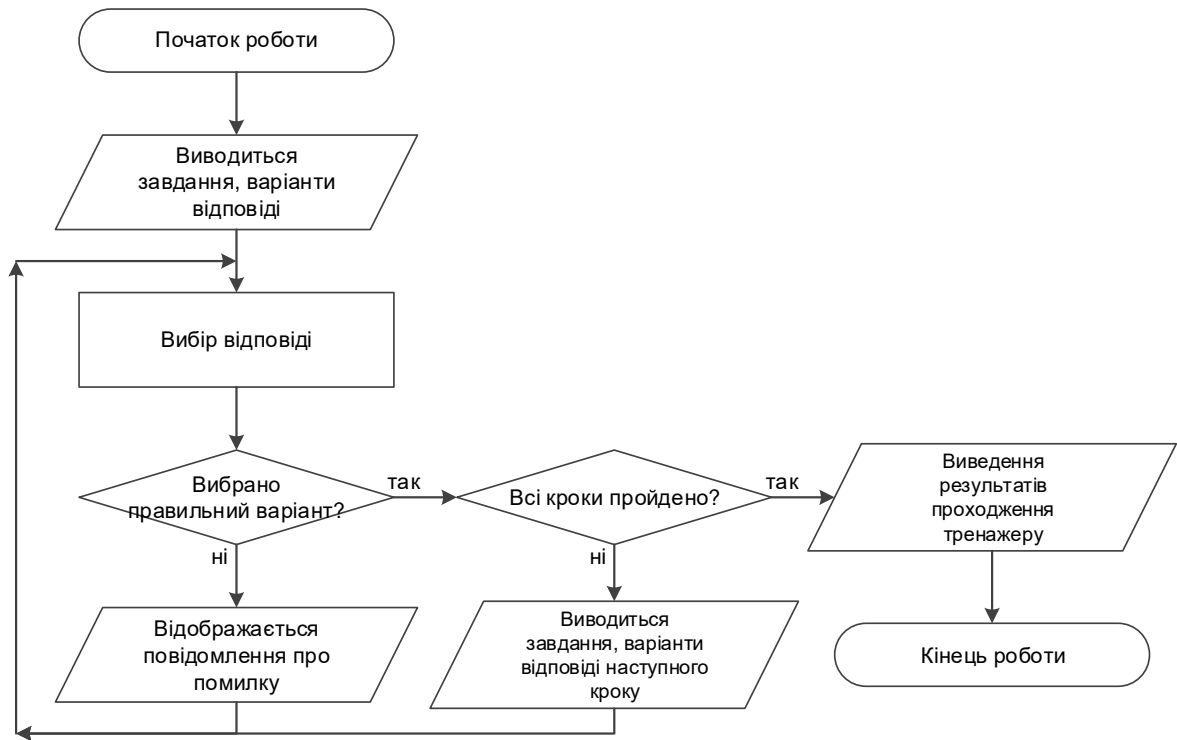


Рисунок 3.2 – Блок-схема алгоритму тренажеру

3. ПРАКТИЧНА ЧАСТИНА

3.1. Обґрунтування вибору програмних засобів для реалізації завдання роботи

JAVA — це об'єктно-орієнтована мова програмування, розроблена Джеймсом Ослінгом і запущена в 1995 році в якості одного із ключових компонентів платформи JAVA Sun Microsystems.

Синтаксис мови взятий в більшості з мови програмування C++ і C, проте JAVA має більш просту об'єктивну модель і більш низький рівень об'єктів. JAVA – додатки як правило трансліровані в байт-код (довжина кожного коду операції — один байт). Який може бути оброблений будь-якою віртуальною машиною JAVA (Java Virtual Machine. JVM) незалежно від комп'ютерної архітектури.

Першочергово реалізація посилань на компілятори JAVA, віртуальних машин і бібліотек класів, була представлена компанією Sun в 1995 році. В травні 1997 року разом з JAVA Community Process Sun розробила технологію JAVA згідно з GNU General Public License. Інші компанії також шукали альтернативи технології запропонованої Sun як GNU Compiler, для JAVA і GNU Classpath.

Розробляючи мову спеціалісти керувались п'ятьма завданнями, які були перетворені в наступні принципи.

- ⇒ простота в використанні, об'єктна орієнтованість і легкість вивчення
- ⇒ надійність і безпечність
- ⇒ незалежність від архітектури
- ⇒ можливість інтерпретації
- ⇒ інтерактивність і динамічність

Для того щоб зрозуміти як працювати з мовою JAVA, необхідно в першу чергу познайомитись з платформою і характеристикою цієї мови.

Можливість JAVA виконувати свій код налюбій із підтримуваних платформ, досягається за рахунок того що її програми транслюються в проміжне уявлення, яке називається байт кодом. Байт код може інтерпретуватись влюбій системі, в якій присутнє середовище виконання JAVA.

Значна кількість раних систем, в якій вводилась можливість незалежності від платформи, зіштовхувалась з недостатком заключеному в зниженні продуктивності.

Байт код легко переводиться безпосередньо в рідні машинні коди, незалежно на те що в JAVA використовується інтерпретатор при якому досягається дуже висока продуктивність. Інструкції java bytecode походжі з інструкціями машинного коду, але призначені для віртуальної машини і створені спеціально для апаратного власника.

Кінцеві користувачі приміняють як правило Java Runtime Environment (JRE конфігурація віртуальної машини, необхідна для виконання Java додатків) встановлену як правило на власному компютері для додатків Java або в веб боаузері для Java аплетів.

Віртуальна машина Java (Java Virtual Machine) це ключовий компонент платформи Java, представляючий собою віртуальне середовище, віртуальний компютер з характеристиками нопмального компютера (реєстром, кодами, набором інформації) який керує додатками Java.

Віртуальна машина Java (Java Virtual Machine) інтерпретує і виконує байт код Java, передчасно створений із початкового тексту Java програми компілятором Java.

Бібліотеки в терміні мов програмування це ресурси програмного забезпечення, які потрібні мові для функціонування. Java використовує такі бібліотеки, які зберігають спеціальні директорії, в папці додатку Java. Звідтам вони завантажуються кожен раз, коли ініціалізується додаток Java. Стандартні бібліотеки Java забезпечують загальний доступ до конкретних функцій Java, які реалізують основні можливості мови [5].

NetBeans IDE - це вільна IDE для мов Java, Python, PHP, JavaScript, C, C++. Проект NetBeans IDE підтримується і спонсорується компанією Oracle, однак розробка NetBeans ведеться незалежною спільнотою розробників-ентузіастів (NetBeans Community) і компанією NetBeans Org.

За якістю і можливостями останні версії NetBeans IDE наближаються до кращих комерційних (платних) IDE, таких як IntelliJ IDEA, підтримуючи рефакторинг, профілювання, виділення синтаксичних конструкцій кольором, автодоповнення набраних конструкцій на льоту, безліч вбудованих шаблонів коду. Перевагою цієї IDE також є те, що вже базова версія має повну підтримку технологій Java EE (тобто для розробки веб-додатків не потрібно нічого встановлювати додатково).

Однак в плані зручності розробки NetBeans IDE поступається IntelliJ Idea, оскільки не має настільки розвинених інтелектуальних засобів аналізу коду, а у багатьох задачах (особливо пов'язаних з J2EE-розробкою) вимагає набагато більше ручної роботи зі створення інтерфейсів, дескрипторів та ін.

[6]

Можливості програмування в NetBeans:

- Розробка Java десктоп додатків з професійними графічними інтерфейсами користувача. Використовується візуальний редактор - Swing GUI Builder. Робота здійснюється шляхом перетягування елементів графічного інтерфейсу з палітри на полотно. Попереднє позиціонування елементів можна здійснювати за допомогою покажчика миші. Панель властивостей і інспектор компонентів надають можливість тонкої настройки кожного компонента інтерфейсу.
- Створення веб-додатків і корпоративних додатків відповідно до стандартів. Середовище NetBeans надає повну підтримку Java EE 6. Дозволяє розробляти веб-сторінки, сервлети, веб-сервіси, Enterprise Java Beans (EJB), проекти Java EE з використанням JavaServer Faces 2.0 (Facelets), Spring, Struts і Hibernate.

- Програмування на PHP, підтримка всіх супутніх мов програмування, технологій і веб-стандартів. Можливість створювати проекти PHP на основі платформи Zend або Symfony. Редактор PHP динамічно інтегрований з функціями редагування HTML, JavaScript і CSS. Проекти PHP можуть бути розгорнуті з середовища NetBeans на локальному або віддаленому сервері при взаємодії через FTP або SFTP.

- Можливість створення, тестування, налагодження і впровадження програм, що функціонують на мобільних телефонах, кишенькових комп'ютерах, телеприставки і вбудованих системах. Visual Mobile Designer (VMD) створює всю необхідну модульну інфраструктуру проекту та забезпечує швидку розробку графічних інтерфейсів шляхом перетягування в робочу область компонентів - екран очікування, екран входу в систему, оглядач файлів, засіб складання повідомлень SMS і екран заставки. Можливість створення призначеного для користувача інтерфейсу на основі SVG.

- Використання JavaFX Composer для візуального структурування додатки JavaFX з графічним інтерфейсом, аналогічно конструктору GUI Swing для Java десктоп додатків.

- Можливість розробки професійних програм на мовах C, C++ для різних платформ - Windows, Linux, Mac і Solaris. Підтримуються всі широко використовувані компілятори - GNU, Cygwin і MinGW. Існує можливість установки необхідного компілятора, визначень препроцесора, параметрів часу компіляції і т.д.

- Розширені можливості по роботі з базами даних - вбудований клієнт до баз даних - MySQL, Postgres, Oracle та ін., Редактор запитів SQL, можливість редагувати таблиці баз даних безпосередньо через редактор таблиць.

- Інтеграція з серверами додатків і контейнерами сервлетів - автоматичне розгортання додатків, управління сервером - запуск, зупинка, перезапуск.

- Багатомовний користувальницький інтерфейс.
- Розширення функціональності за допомогою модулів, гнучка система управління компонентами, модулями, оновленням та завантаженням модулів через інтернет.

NetBeans - єдина IDE, яка влаштує і початківця розробника і професіонала. Наявність докладної вбудованої довідкової системи забезпечить швидкий старт для початківців користувачів [7].

3.2. Опис процесу програмної реалізації

Тренажер реалізовано мовою програмування Java в середовищі програмування NetBeans.

Для переключення між вікнами тренажера було створено змінну сі типу CardLayout. Також було створено наступні масиви типу String:

- String[] step – перелік кроків;
- int[] k – кількість зроблених помилок по кожному завданню.

Для функціонування тренажеру було створено наступні функції:

- nextStep() ;
- previousStep();
- nextPanel(String t);
- nextQuestion();
- correctAnswer();
- ShowMessage();
- helpText().

Функція nextStep() переходить до наступного кроку алгоритму, а previousStep() – до попереднього.

Функція nextPanel(String t) переключає поточне вікно на наступне, вказане параметром t.

Для зміни завдання використовується функція nextQuestion().

Функція `correctAnswer()` була створена для перевірки, чи правильна вибрана відповідь на питання тестів.

При невірній відповіді викликається функція `ShowMessage()`.

```
private void ShowMessage() {
    String text = " Відповідь неправильна. За потреби зверніться до довідки.";
    JLabel label = new JLabel(text);
    label.setFont(new Font("Tahoma", 0, 14));
    JOptionPane.showMessageDialog(Main, label, "",
JOptionPane.WARNING_MESSAGE);
}
```

`helpText()` – показує довідку.

При натисненні кнопки «Вперед» відбувається подія:

```
private void ForwardActionPerformed(java.awt.event.ActionEvent evt) {
    nextStep();
}
```

При натисненні кнопки «Назад» відбувається подія:

```
private void BackActionPerformed(java.awt.event.ActionEvent evt) {
    previousStep();
}
```

При натисненні кнопки «Завершити роботу» відбувається подія:

```
private void CompleteActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}
```

Якщо відповіді на питання тестів представлені у вигляді формул, то при натисненні на формулу відбувається подія, яка властивості Selected відповідного `JRadioButton` встановлює значення `true`.

Для ініціалізації аплету використовується:

```
try {
    java.awt.EventQueue.invokeLaterAndWait(new Runnable() {
        public void run() {
            initComponents();
        }
    });
} catch (Exception ex) {
    ex.printStackTrace();
}
```

Для запуску тренажера, встановлення його розмірів і виведення використовується стандартний вивід:

```
public static void main(String[] args) {
    JFrame frame = new JFrame("Тренажер");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    JApplet applet = new JApplet();
    frame.getContentPane().add(applet);
    applet.init();
    applet.start();
    frame.pack();
    frame.setSize(600, 520);
    frame.setResizable(false);
    frame.setLocationRelativeTo(null);
    frame.setVisible(true);
}
```

ВИСНОВКИ

Метою виконаного проекту є створення програми-тренажера з теми «Регулярні операції над мовами» для дистанційного курсу «Теорія програмування». Проект виконаний з наведеними алгоритмом роботи програми, блок-схемою алгоритму, описом програмної реалізації.

Під час виконання роботи було створено алгоритм роботи тренажера. Для програмної реалізації тренажера була використана мова програмування Java, середовище програмування NetBeans.

Для полегшення розміщення програми-тренажера на сайт дистанційної освіти тренажер розроблений у вигляді Java-аплету.

У результаті виконання роботи був створений тренажер з теми «Регулярні операції над мовами» із зрозумілим для користувача інтерфейсом.

Для переключення між вікнами тренажера було створено змінну сі типу CardLayout. Також було створено наступні масиви типу String:

- String[] step – перелік кроків;
- int[] k – кількість зроблених помилок по кожному завданню.

Для функціонування тренажеру було створено наступні функції:

- nextStep() ;
- previousStep();
- nextPanel(String t);
- nextQuestion();
- correctAnswer();
- ShowMessage();
- helpText();

Також при проходженні тестів реалізована можливість завершення тестування.

При завершенні тестування виводиться повідомлення про проходження тестування, а також кількість зроблених помилок.

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. Кадемія М. Ю., Шахіна І. Ю. Інформаційно-комунікаційні технології в навчальному процесі : Навчальний посібник [Текст] / Кадемія М.Ю., Шахіна І.Ю. / – Вінниця, ТОВ «Планер». – 2011. – 220 с.
3. Черненко О.О. Електронний навчально-методичний посібник для самостійного вивчення навчальної дисципліни «Теорія програмування» для студентів напряму 6.040302 «Інформатика»
4. Нікітченко М.С. Теоретичні основи програмування: Навчальний посібник [Електронний ресурс] / М.С. Нікітченко. – Київ: КНУ ім. Т.Г. Шевченка, 2009. – 200 с. – Режим доступу: <http://ttp.unicyb.kiev.ua/doc/TOP.pdf>.
5. Подробнее о технологии Java [Електронний ресурс] /. – Режим доступу: <https://www.java.com/ru/about/>.
6. Кадомський К.К. Java. Теорія і практика: навчальний посібник для студентів природничих спеціальностей університетів [Текст] / К.К. Кадомський, П.К. Ніколюк / – Вінниця, Донну, 2019. – 197 с.
7. Java [Електронний ресурс] / Матеріал з Вікіпедії — вільної енциклопедії. – Режим доступу: <http://uk.wikipedia.org/wiki/Java>.
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