

**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

**ТРЕНАЖЕР З ТЕМИ «МАТЕМАТИЧНІ ОСНОВИ ТЕОРІЇ АЛГОРИТМІВ»
ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО КУРСУ «ТЕОРІЯ ПРОГРАМУВАННЯ»
ТА РОЗРОБКА ЙОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

зі спеціальності 122 «Комп'ютерні науки»

Виконавець роботи Костромін Ігор Ігорович

_____ « ____ » _____ 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.....	8
3. ТЕОРЕТИЧНА ЧАСТИНА	12
3.1. Математичні основи теорії: множини, відношення	12
3.2. Розробка алгоритму роботи тренажеру	29
3.3. Блок-схема, яка підлягає програмуванню	38
4. ПРАКТИЧНА ЧАСТИНА	40
4.1. Обґрунтування вибору програмних засобів.....	40
4.2. Розробка тренажеру	43
ВИСНОВКИ.....	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	47
ДОДАТОК А. КОД ПРОГРАМИ	49

ВСТУП

Теорія алгоритмів (ТА) виникла з внутрішньої потреби математики. Математична логіка, метаматематика, алгебра, геометрія і аналіз залишаються і сьогодні однією з основних областей застосування теорії алгоритмів.

Важливим елементом в освіті є дистанційне навчання, а саме дистанційний курс. Міра і спосіб використання комп'ютерних технологій при підготовці дистанційного курсу значно впливає на ефективність його застосування. Світовий досвід показує, що використання динамічних об'єктів для створення наочних моделей процесів, адаптивне моделювання студента в багатьох випадках значно підвищує навчальний ефект. Для організації дійсно ефективного начального процесу дистанційного навчання необхідна систематична робота з оболонкою студента, так і викладача майже кожного дня на протязі всього терміну навчання. Дистанційне навчання досить актуальне, але основна проблема полягає у тому, що вимоги до фахівців змінюються щороку, виходячи з цього виникає необхідність удосконалення навчального процесу дистанційного направлення.

Метою роботи є програмна реалізація тренажера з теми «Математичні основи теорії алгоритмів» дистанційного начального курсу «Теорія програмування» та закріплення знань.

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

Предметом розробки є програмний продукт, що реалізує тренажер з теми «Математичні основи теорії алгоритмів» на мові програмування Java.

Саме поняття алгоритму тісно пов'язане з лінгвістикою, економікою, фізіологією мозку і психологією, філософією і природознавством тощо. Прикладом може послужити розумова чи практична діяльність людини у будь-якій сфері.

Перелік використаних методів – використання математичних основ теорії алгоритмів.

Робота складається з чотирьох розділів. У першому розділі розглянуто постановку задачі тренажеру. У другому розділі описано інформаційний огляд. У третьому розділі представлено математичні основи теорії: множини і відношення, розробку алгоритму роботи тренажеру та її блок-схему. У четвертому розділі – описано обґрунтування вибору програмних засобів, розробку тренажеру.

Обсяг пояснювальної записки: 53 стор., в т.ч. основна частина - 44 стор., джерела - 17 назв.

1. ПОСТАНОВКА ЗАДАЧІ

Одним з пріоритетних напрямів програми модернізації загальноосвітньої і вищої школи визнане дистанційне навчання. В сучасних умовах існує потреба отримання вищої освіти дистанційно, що викликано необхідністю навчатися без відриву від виробництва, отримання освіти людьми з обмеженими можливостями та тими, що перебувають за кордоном або у місцях позбавлення волі. Таку можливість надає дистанційне навчання, яке здійснюється завдяки інформаційно-освітнім технологіям і системам комунікації.

Дистанційна форма навчання має ряд беззаперечних переваг. Зокрема, здобувач вищої освіти може навчатися у зручний для нього час, звичному оточення та у відносно автономному темпі. Варто врахувати також і нижчу вартість такого навчання, оскільки відпадає потреба в оренді приміщень, оплаті значної кількості персоналу та економію часу.

Проте, система дистанційного навчання має і недоліки. По-перше, для успішної корекції навчання та адекватного оцінювання важливо мати безпосередній контакт із здобувачем. Крім того, неможливо точно перевірити, чи саме та людина працює, виконує завдання чи це робить хтось інший. Тому остаточний контроль якості знань все ж таки проводиться на очній сесії. Крім того, не у всіх населених пунктах є можливість доступу до мережі Інтернет-зв'язку. І найголовніше, при дистанційному навчанні втрачається безпосередній контакт між викладачем та студентом. При тривалому дистанційному навчанні студент перестає правильно формулювати свої думки, висловлюватись та проводити дискусійне обговорення. Разом з тим, така форма навчання потребує свідомого і мотивованого підходу до отримання освіти. Можливість навчатися у зручний час може перетворитися не на систематичне навчання, а на постійну прокрастинацію цього виду

діяльності. Саме тому дистанційна форма потребує особливої самоорганізованості та вміння розрахувати свій час.

За умови дистанційного навчання активна роль викладача не зменшується, оскільки він має визначити рівень знань здобувача, та прийняти рішення щодо коригування програми навчання з тим, щоб домогтися найкращого засвоєння пройденого матеріалу.

За потреби студент може отримати консультативну допомогу викладача, спілкуючись з ним в онлайн режимі, безпосередньо використовуючи інтернет як засіб зв'язку (web-чат, IRC, ICQ, інтерактивне TV, web-телефонію, Telnet).

Досвід використання дистанційної форми навчання виявив іще одну особливість, а саме – велике навантаження на зір у зв'язку з необхідністю тривалий час перебувати за комп'ютером.

Тому, застосовуючи дистанційну форму навчання потрібно урізноманітнювати її види. Найбільш поширеними є наступні види дистанційних технологій:

- чат-заняття, які проводяться синхронно, коли всі учасники мають одночасний доступ до чату;

- веб-заняття, або дистанційні лекції, конференції, семінари, ділові ігри, лабораторні роботи, практикуми та інші форми навчальних занять, що проводяться за допомогою засобів телекомунікацій та інших можливостей інтернету;

- телеконференції, що проводяться, на основі списків розсилки з використанням електронної пошти. Для навчальних телеконференцій характерно досягнення освітніх завдань.

Також існують форми дистанційного навчання, при якому навчальні матеріали висилаються поштою в регіони. Однак не всі знання можна отримати дистанційно. Так, наприклад, навчитися самостійно деяким видам творчої діяльності, при відсутності прямого контакту студента і викладача, практично неможливо. Одна з головних проблем

запровадження інноваційних форм навчання є вибір оптимального співвідношення найкращих традицій наявної освітньої системи, сучасних педагогічних інновацій та інструментарію інформаційно-комунікаційних технологій. Як свідчить практика і деякі дослідження, тенденція навчання чітко розвивається в напрямку змішаного навчання (blended learning) як процесу, котрий створює комфортне інформаційне освітнє середовище, системи комунікацій, що надають всю необхідну навчальну інформацію.

Елементи дистанційного навчання все більше застосовуються в навчальних курсах дисциплін.

Отже, на сьогоднішній день дистанційна освіта розвивається, удосконалюється та охоплює різні групи населення, адже для сучасної економіки – «економіки знань» притаманна парадигма – навчатися протягом всього життя.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

Дистанційне навчання – нова організація освітнього процесу, що ґрунтується на використанні як кращих традиційних методів навчання, так і нових інформаційних та телекомунікаційних технологій, а також на принципах самостійного навчання, призначена для широких верств населення незалежно від матеріального забезпечення, місця проживання, стану здоров'я.

Дистанційне навчання дає змогу впроваджувати інтерактивні технології викладення матеріалу, здобувати повноцінну освіту, підвищувати кваліфікацію співробітників у територіально розподілених місцях. Процес навчання може відбуватися будь-де і будь-коли, єдина умова – доступ до мережі Інтернет.

На сьогоднішній день інформаційні технології є невід'ємною частиною нашого життя. Їх основу становить програмне забезпечення, яке проникло в усі сфери людського життя.

На сучасному етапі у зв'язку з появою нових інформаційних технологій спостерігаються наступні зміни у системі освіти :

1. формальний підхід до навчання замінюється когнитивним;
2. один основний підручник з курсу замінюється можливістю вибрати з великого розмаїття джерел інформації;
3. поява гіпертексту;
4. використання різноманітних технологій і саме тих, що принесуть найбільшу вигоду студентам;
5. навчаючі технології не повинні одна одну замінювати, вони мусять доповнювати одна одну.

Термін вільне програмне забезпечення ввів Річард Столмен, засновник проекту GNU. Вільними називаються програми, автор яких опублікував їх у супроводі так званої «вільної ліцензії», що дозволяє:

1. користуватися програмою для будь-яких цілей і на необмеженій кількості комп'ютерів або місць в мережі;
2. безперешкодно отримувати доступ до її вихідних кодів;
3. виготовляти необмежену кількість додаткових її примірників;
4. модифікувати її як для власного користування, так і для поширення на тих же умовах.

Річард Столмен виділяє такі основні причини використання вільного програмного забезпечення у навчальному процесі:

1. вільне програмне забезпечення дозволить заощаджувати кошти. Так навіть у багатих країнах існують проблеми з фінансуванням освіти. У бідних країнах використання вільного програмного забезпечення, яке можна вільно копіювати і поширювати, дозволить розвивати інформаційні технології;

2. вільне програмне забезпечення дає можливість студентам вивчати, як воно працює — вихідний код є доступним;

3. студенти матимуть змогу копіювати програмне забезпечення у вузах для домашнього використання, що дозволить уникати використання неліцензійного програмного забезпечення.

На сьогоднішній день існує багато рішень для систем дистанційного навчання, відмінних технічними можливостями, наявністю і рівнем складності різних функціональних компонентів, наприклад, Oracle (i-Learning), IBM (Learning Space), WebCT, “Прометей” виробництва НІЦ АСКБ, e-Learning компанії “Гіперметод”, та засоби Open Source: MOODLE, ATutor, Dokeos, Claroline тощо. Проте ми б хотіли розглянути розповсюджену за принципом Open Source на умовах ліцензії GNU/GPL, систему MOODLE.

Автором ідеї і концепції системи MOODLE, а також її засновником є Мартін Даугіамас (Martin Dougiamas – доктор педагогічних наук з Curtin University Technology, Perth, Австралія). Головною його метою було створення системи, відмінної від доступних на ринку, а саме такої, яка б

враховувала педагогічні аспекти, що базуються на основах пізнавальної психології, і, перш за все, однієї з її течій, що називається конструктивізмом.

Інтерфейс, допомога і документація системи MOODLE існують на декількох десятках мовах, які доступні і можуть бути вибрані і використані в разі потреби кожним користувачем.

Система повністю обслуговується з рівня стандартного веб-броузера. Не передбачає ніяких спеціальних вимог до устаткування і операційної системи. Є повністю безкоштовною. В поєднанні з відносно простою і добре описаною інсталяцією платформа MOODLE без спеціальних труднощів може бути швидко встановлена на вузівському, шкільному або навіть власному сервері .

Серед основних можливостей використання системи, можна виділити наступні:

1. Надання через мережу освітніх матеріалів всім бажаним студентам або вибраній групі користувачів.

2. Забезпечення і підтримка можливості взаємного спілкування зарівно як між учнями/студентами, які беруть участь в курсі, так і між учнями/студентами і вчителем/ведучим.

3. Документування і збереження робіт, результатів дискусії, документування заданих питань і одержаних відповідей.

4. Надання інструментів, які забезпечують можливість здійснення поточного контролю і оцінки досягнень окремих учасників і доставку зворотних даних щодо кожної теми, у тому числі оцінок і відгуків (рецензій) на їх роботи.

5. Надання можливості аналізу участі і активності окремих учасників курсу, аналіз часу, присвяченого на роботу з матеріалами.

Отже, ми можемо побачити, що система MOODLE швидко розвивається і відіграє велику роль для дистанційного навчання у вищих начальних закладах. Вона відкриває студентам доступ до нетрадиційних

джерел інформації, підвищує ефективність самостійної роботи, знаходження і закріплення різних професійних навичок, а викладачам дозволяє реалізовувати принципово нові форми і методи навчання із застосуванням концептуального і математичного моделювання явищ і процесів.

Система дистанційної освіти може і повинна зайняти своє місце в системі освіти, оскільки при грамотній її організації вона може забезпечити якісну освіту, що відповідає вимогам сучасного суспільства сьогодні.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Математичні основи теорії: множини, відношення

1. Означення алгоритму

Поняття алгоритму є одним з основних понять сучасної математики. Ще на початкових етапах її розвитку (Стародавній Єгипет, Вавилон, Греція) у математиці стали формуватися різноманітні обчислювальні процеси. За їх допомогою при розв'язуванні задач шукані величини обчислювалися послідовно з висхідних величин за певними правилами та інструкціями. З часом такі правила (інструкції) одержали назву алгоритмів.

Термін алгоритм походить з імені середньовічного вченого-математика Аль-Хорезмі, котрий ще в IX ст. (825 р.) дав правила виконання чотирьох операцій арифметичних дій в десятковій системі числення. Перелік виконуваних арифметичних дій був названий ал-горизм.

З 1747 р. замість слова алгоризм стали вживати алгорисмус, смисл якого полягав в комбінуванні чотирьох операцій арифметичного обчислення — додавання, віднімання, множення та ділення. Через століття для створеного Бебіджем комп'ютера, Адою Лавлейс було написано першу програму, що є першим технічним втіленням алгоритму.

До 1950 р. алгорисмус став алгорифмом. Смысл алгорифму найчастіше пов'язувався з алгорифмами Евкліда — процесами знаходження найменшої спільної міри (двох відрізків, найменшого спільного дільника двох многочленів тощо).

Аж до 30-х років XX століття поняття алгоритму мало методологічне значення. Його означували евристично.

Означення 1. Скінченна сукупність коректно сформульованих правил розв'язування задачі з класу (множини) задач називається алгоритмом.

Таке означення алгоритму не є строгим, математичним, оскільки у ньому не означено понять клас задач, правило (їх розв'язування). Протягом тривалого часу математики задовольнялися цим означенням, теорії алгоритмів не існувало. Не було серйозних випадків, коли математики розійшлися б в тлумаченнях стосовно того, чи є алгоритмом певний конкретно заданий процес.

Становище суттєво змінилося, коли технічні засоби математичних обчислень стали настільки потужними та зручними у практичному користуванні, що розв'язок задачі в аналітичному вигляді (у вигляді формули) виявився не потрібним. Уможливилось розв'язування задач, з невідомим розв'язком, отримання якого в аналітичному вигляді було важко доступним. Почали бурхливо розвиватися обчислювальні (чисельні) методи розв'язування задач, і на перший план висунулася проблема доведення існування відповідних алгоритмів. Виявилось, що довести наявність чи відсутність алгоритму не все одно. Перше можна зробити шляхом фактичного опису хоч одного методу розв'язування. В цьому випадку достатньо означення 1 поняття алгоритму, щоб переконатися, що описане є алгоритмом. Довести відсутність існування алгоритму таким шляхом неможливо. Для цього потрібно коректніше означити, що таке алгоритм.

В 20-х роках ХХ-го століття проблема означення алгоритму стала однією з основних, фундаментальних математичних проблем. Її вирішення було одержано у двох формах всередині 30-х років 20-го століття у роботах таких відомих математиків як Гільберт, Гедель, Черч, Кліні, Пост, Т'юрінг на базі означень понять класів:

- а) арифметичних функцій (які назвали рекурсивними);
- б) процесу.

Пізніше в роботах Маркова і Калужніна з'явилося інше трактування алгоритму — як особливої відповідності між словами в абстрактному алфавіті (з означенням понять алфавіт, слово, відповідність).

Отже, для коректного математичного означення алгоритму потрібно в свою чергу ввести, означити першорядні, математично строгі поняття клас функцій, рекурсивна функція, процес, клас процесів, абстрактний алфавіт тощо. З технічної, інженерної, практичної сторони цього не має потреби робити.

Означення 2. Скінченна сукупність операцій, якій притаманні властивості

a) дискретності — виконання чергової операції починається у певний момент часу, моменти часу не обов'язково еквідистантні;

b) детермінованості — перед виконанням першої операції відомі початкові дані, у будь-який момент часу точно відомі наступні операції (прогнозованість) та результат попередньої операції;

c) впорядкованості — встановлена черговість операції (інакше — відношення порядку на їх множині);

d) масовості — алгоритм розв'язує задачу, що належить до класу задач;

e) елементарності — завжди можна вказати елементарну (виконувану апаратно, не програмовану) операцію — називатимемо алгоритмом.

Таким чином, алгоритм виникає при вирішенні проблеми (інформування, управління). Для цього будується математична модель проблеми і на її базі ставиться задача(і) та будуються методи її розв'язування. Вважатимемо, що алгоритм задано, математичні проблеми його існування, коректності постановки задачі, яку за його допомогою розв'язують, вирішено. А під ТА розумітимемо систему фактів про властивості відповідних математичних об'єктів, якими подають алгоритм, у тому числі про взаємозв'язки між ними (їх алгебру, структури; для детальнішого знайомства з цими поняттями потрібно звернутися до відповідних посібників та підручників, зокрема, конспекту лекцій з курсу дискретної математики, основ алгебри тощо). Особливо про такі

взаємозв'язки, які підпадають означенню представлення. При цьому не забуватимемо про математичну коректність, зокрема, вербального, графічного, аналітичного та інших способів подання (визначення) алгоритму.

2. Подання алгоритмів

В інформаційних та керуючих технологіях для інформування та вироблення керівного впливу використовуються вимірювання — спеціальні процедури з вимірюваним об'єктом та мірою (однорідною з об'єктом зразковою величиною). Здавна відомий алгоритм Евкліда знаходження спільної міри двох взаємно несумірних відрізків (найбільшого спільного дільника двох чисел). Розглянемо технічні та інформаційні ресурси (характеристики процесора та математичну модель розв'язуваної задачі) для алгоритму Евкліда, його алгебричну структуру. Її знання уможливорює тотожні перетвори алгоритму.

2.1. Засоби подання алгоритму

Для розуміння властивостей вербального подання потрібно ознайомитися з основами лінгвістики. Наприклад: Соссюр, Фердінан де. Курс загальної лінгвістики. — Київ: Основи, 1998. — 324 с.

2.1.1. Математична модель.

Задача знаходження найбільшого спільного дільника постала при вирішенні проблеми вимірювань — знаходження масштабу міри. Відомо, що вимірювання є певною процедурою, алгоритмом над вимірюваною величиною та мірою. При порівнянні об'єктів за значеннями їм властивої величини потрібно спочатку вирішити проблему вибору міри та знайти її значення. Для числових величин цю проблему вирішено постановкою та розв'язуванням задачі найбільшого спільного дільника.

Розглянемо цілі числа (натуральний ряд чисел). Всяке ціле, що ділить без остачі a, b, \dots, l , називається їх спільним дільником. З теорії подільності відома теорема про ділення з остачею:

Теорема 1. Всяке ціле a представляється єдиним чином за допомогою цілого b рівністю $a = bq_0 + r_0$; $0 \leq r_0 < b$.

Якщо взяти числа b та r_0 , то отримаємо представлення $b = r_0q_1 + r_1$. Далі, розглядаючи послідовно подібним чином числа $r_0, r_1; \dots, r_{n-1}, r_n; r_n, r_{n+1}$, легко встановити, що коли $r_{n+1} = 0$, то r_n буде найбільшим спільним дільником чисел a, b [див. Виноградов И.М. Основы теории чисел.— М.: Наука, 1981.— 176 с.].

2.1.2. Ресурси та система команд процесора.

Потрібні ресурси процесора та система команд визначаються моделлю задачі та методом її розв'язання. Це диктує розроблення спеціалізованого процесора, "під задачу". Проте часто вибирається процесор "близький" до потрібного — "під клас задач" у найкращому випадку. Отже, нехай маємо процесор з (а) — ресурсами:

i. арифметичний пристрій з регістрами для результату q_0 та x_0 і регістрами x_1, x_2 для операндів;

ii. пристрій вводу;

iii. пристрій виводу;

та (б) — системою команд (множиною елементарних операцій арифметичного пристрою процесора):

i. перемножити операнди;

ii. ділити з відкиданням дробової частини результату (функція ціла частина від числа) та присвоїти знак мінус;

iii. присвоїти регістрові вміст іншого регістру;

iv. перейти на (відповідний) крок (конкретний зміст дивись в наведеному алгоритмі);

v. ввести число у регістр з пристрою вводу;

vi. вивести число на пристрій виводу;

vii. зупинити.

2.1.3. Лінгвістичні засоби подання алгоритму.

Розглянемо лінгвістичні засоби на прикладі української мови (алфавіту, скінченної множини спеціально вибраних слів та граматики). Назвемо тому його вербальним (на словах) поданням. Отже, для вербального подання алгоритму використовують звичайну мову, її граматику — алфавіт, лексику, синтаксис тощо. При цьому використовується й семантика мови, виконуючи функцію коментаріїв. Але поліморфізм мови порушує однозначність слів — інформативних денотатів понять, якими означають операції та ін. складники алгоритму. Це знижує ефективність використання мови, зокрема, семантики при поданні алгоритму. Тотожні перетвори (алгебра) алгоритму забезпечуються граматику мови, вибраної для його вербального подання, при врахуванні властивостей моделі задачі, методу її розв'язування, процесора тощо.

Засобами мови (вербальні) подаються усі операції та оператори (арифметичні, логічні) алгоритму.

2.1.4. Алгоритм (Евкліда).

Користуючись обмеженою кількістю (множиною) слів (понять), які повно і несуперечливо означають поняття моделі задачі, систему команд процесора та означенням алгоритму побудуємо алгоритм Евкліда:

- i. **ВВЕСТИ** більше число у реєстр x_2 ;
- ii. **ВВЕСТИ** менше число у реєстр x_1 ;
- iii. **функція** — ціла частина q_0 частки від ділення вмісту реєстрів x_2 на x_1 зі знаком мінус;
- iv. **добуток** x_0 вмісту реєстрів x_1 та q_0 ;
- v. **сума** x_0 вмісту реєстру x_0 з вмістом реєстру x_2 ;
- vi. **переслати** вміст реєстру x_1 у реєстр x_2 ;
- vii. **переслати** вміст реєстру x_0 у реєстр x_1 ;
- viii. **переслати** нуль в реєстр x_0 ;
- ix. **якщо** x_1 не рівне нулеві — перейти на крок iii;
- x. **ВИБЕСТИ** вміст реєстру x_2 .

xi. ЗУПИНИТИ.

Формальне подання алгоритмів

1. Подання алгоритму блок-схемами

На практиці алгоритми подають за стандартом — у вигляді блок-схеми, рис. 3.1:

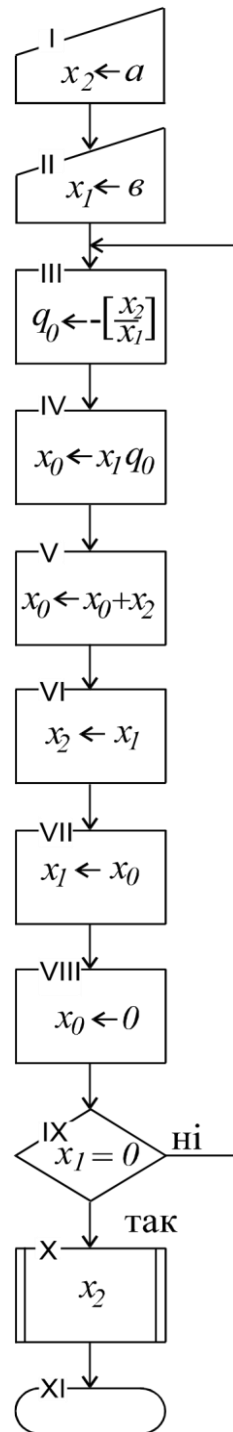


Рисунок 3.1 – Блочна впорядкована схема алгоритму (номери блоків відповідають номерам кроків алгоритму)

Стандартна блок-схема алгоритму в умовних позначеннях, виконана за певними правилами відображає порядок обчислень та їх зміст, логічні операції, операції вводу-виводу тощо. За умови збереження порядку обчислень, при фіксованих моделі задачі, методі її розв'язування та процесорі (системі команд) виконувати тотожні перетворення такої блок-схеми неможливо. Знімаючи ці обмеження, наприклад, при надмірних множинах команд процесора, моделей задачі на базі їх алгебр можлива побудова алгебри блок схем алгоритмів. Це уможливило їх тотожні перетвори.

З теорії цифрових кіл (див., наприклад, Рабинер Л., Голд Б. Теорія и применение цифровой обработки сигналов.—М.: Мир, 1978) відомі функціональні блок-схеми. У їх позначеннях подаються алгоритми обчислень (рис. 3.2):

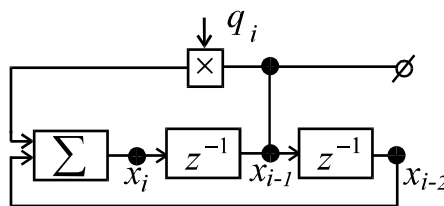


Рисунок 3.2 – Блочна функціональна схема алгоритму

($x_i, x_{i-1}, x_{i-2}, q_i$ — вміст регістрів x_0, x_1, x_2, q_0 на i -му кроці обчислень; позначення всередині блоків: Σ — суматор, \times — помножувач, z^{-1} — обмін даними між регістрами на i -му кроці обчислень ("затримка" даних на час обчислень); стрілками показано функціональні "входи").

На функціональній блок-схемі алгоритму впорядкування обчислень позначають нумерацією її вузлів, яку виконують спеціально, використовуючи теорему про вигляд матриці впорядкованого цифрового кола. Логічних операцій ("галуження" обчислень) така блок-схема не відображає. Тому рис. 3.2 відображає лише частину алгоритму з рис. 3.1. На функціональній блок схемі (цифровому колі) існує алгебра лінійних тотожних перетворень. Крім того, існує ізоморфне його відображення та його алгебра, що надає великі можливості для розробки алгоритмів.

2. Аналітичне подання алгоритму

В аналітичному поданні алгоритму операції подають у вигляді формул, зі степенем дискретності, який відповідає елементарності операцій процесора. Ці формули повинні належати певній математичній структурі, щоби таким чином поданий алгоритм належав певній алгебрі (задля виконання тотожних його перетворів). Крім того, формули подають впорядковано за часом їх виконання. Циклічність та галуження обчислень у цій послідовності подають за допомогою умовних, безумовних переходів, логічних операцій.

Проаналізуймо властивості послідовності операцій алгоритму Евкліда:

- i. ВВЕСТИ більше число у реєстр x_2 ;**
- ii. ВВЕСТИ менше число у реєстр x_1 ;**
- iii. Функція — ціла частина q_0 частки від ділення вмісту реєстрів x_2 на x_1 зі знаком мінус;**
- iv. Добуток x_0 вмісту реєстрів x_1 та q_0 ;**
- v. Сума x_0 вмісту реєстру x_0 з вмістом реєстру x_2 ;**
- vi. Переслати вміст реєстру x_1 у реєстр x_2 ;**
- vii. Переслати вміст реєстру x_0 у реєстр x_1 ;**
- viii. Переслати нуль в реєстр x_0 ;**
- ix. Якщо x_1 не рівне нулеві — перейти на крок iii;**
- x. ВІВЕСТИ вміст реєстру x_2 .**
- xi. ЗУПИНИТИ.**

Видно, що операції **i-iv** та **vi-viii** є операціями алгебри чисел. Інші операції (наприклад, **v** та **ix**) — ні. Це, а також впорядкування операцій за часом, не дозволяє означити алгебричну структуру, а, отже, й аналітичну модель алгоритму при заданій математичній моделі задачі, множині елементарних команд процесора.

Проблема існування аналітичної моделі алгоритму розв'язання задачі знаходження найбільшого дільника (двох цілих) чисел (необхідних та достатніх умов для цього) вимагає спеціального розгляду.

3. Подання алгоритму графом

Безпосередньо за функціональною блок-схемою будується граф цифрового кола, рис. 3.3:

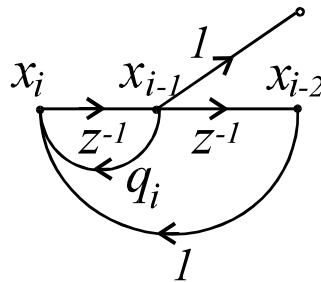


Рисунок 3.3 – Подання алгоритму у вигляді графу (цифрового кола)

Графи є об'єктом вивчення теорії графів (див., наприклад, Оре О. Теория графов.— М.: Наука, 1980, або Белов В.В., Воробьев Е.М., Шаталов В.Е. Теория графов.— М.: Высшая школа, 1976, чи конспект лекцій з дискретної математики). Формально граф алгоритму подається множинами вузлів v , направлених гілок h з "вагою" (деколи їх називають вершинами та ребрами) і предикатними відношеннями між ними. Для подання цих відношень використовуються матриці. Матрицею інциденцій $I = [V \times H]$ описують відношення між вузлами та гілками. Її елементами є 0 чи 1, залежно до відношення між відповідним вузлом та гілкою. Матрицею суміжності $R = [V \times V]$ описують відношення між вузлами. Її елементами r_{ij} є числа, якими подають число гілок між відповідними вузлами. Між цими матрицями існує зв'язок.

Використовуючи матриці інциденцій і суміжності, інші властивості графу обчислюють характеристики графу чи його частин — шляхів, дерев тощо. Ці характеристики мають свої інтерпретації та аналогії на характеристиках алгоритму (складність, швидкодія тощо). Наприклад, знаючи час виконання операцій, які відображені вузлами та гілками графу

досліджуваного алгоритму, та використавши поняття шляху можна побудувати алгоритм обчислення часу роботи досліджуваного алгоритму; автоматичним підрахунком числа вузлів, гілок можна встановити складність алгоритму (надаючи певним операціям відповідну "вагу"). Матриці R та I використовують при цьому для подання алгоритму відповідною машинною мовою програмування.

Використовуючи функціональну інтерпретацію вузлів, гілок та їх характеристик за допомогою графів описують точність, робастність алгоритму (стійкість та чутливість до "збоїв" процесора та неточних даних чи параметрів алгоритму). Тут використовуються результати лінійної алгебри, теорії дискретних (цифрових) кіл. Зокрема, матриці I та R використовуються при розв'язуванні відповідних систем лінійних алгебричних рівнянь, що виникають. Складати ці рівняння немає потреби, бо вони автоматично задаються цими матрицями. Алгоритми їх розв'язування та програми також задаються відповідними програмними середовищами. Ці програмні середовища мають інтерфейс користувача (наприклад, "віконний"), за допомогою якого вказується потрібний вигляд аналізу (робастність, точність тощо).

На базі графів алгоритму проводяться також тотожні перетвори його структури. При цьому модифікуються матриці I та R . Ця модифікація є комбінаторною з використанням тотожних перетворів матриць та відношення порядку на множині вузлів. За таких перетворів отримують конвеєризацію операцій, їх розпаралелювання, виявлення систолік алгоритму тощо.

Алгоритми інформаційних та керуючих систем

1. Алгоритми вимірювання

Для обчислень значень інформативних величин, для керування потрібні дані. Дані отримуються за допомогою спеціальних процедур (алгоритмів) над відібраними від явища значеннями властивих йому

величин та їх мірами. Об'єктом виміру є властивість. Маса, колір, електричний опір, попит на товар, купівельна спроможність, розумова спроможність тощо. Властивості відносять до емпіричних явищ — людей, речовин, фізичних полів. Один об'єкт має кілька властивостей. Вимірюючи одну властивість нехтуємо іншими. Властивості перебувають у взаємозв'язках. Тому можливі непрямі вимірювання. З точки зору результату виміру різні об'єкти можна вважати еквівалентними, якщо він однаковий. Властивості об'єктів моделюють математичними об'єктами. У цьому сенсі вони наділяються математичною структурою.

Означення 11. Гомоморфне відображення властивостей виміру на математичний простір називають вимірювальною шкалою.

Теорія шкал базується на теоретико множинному апараті відношень. Множина величин S з емпірично встановленими відношеннями між ними R_i складає систему $E : \{S, R_i\}$. На практиці досить зручно користуватись числовою системою $N : \{M, P_i\}$, де M — множина чисел, P_i — відношення між числами. Тобто, потрібно емпірично встановлені пари величин пов'язати з парами чисел. У теорії шкалу подають трійкою $\langle E, N, \Psi \rangle$, де Ψ — гомоморфне відображення (в один бік) E на N .

Відомі у застосуваннях (розпізнавання образів, побудова емпіричних залежностей тощо) такі шкали — абсолютна, відношень, інтервалів, порядку, назв.

Абсолютна шкала є метричною, без перетворень. Наприклад, вимірювання характеристик мікросвіту (енергії кванту) за її макро-проявами (частотою випромінювань ν) здійснюють за співвідношенням $E=h\nu$, відомим у квантовій механіці (тут h — константа).

Шкала відношень (пропорційна) є метричною з перетвореннями (подібності — зсуву, масштабу тощо). Наприклад, знаючи коефіцієнт k для пружини можна міряти силу F за її розтягом x використавши співвідношення $F=kx$ (закон Гука).

Шкала інтервалів метрична частково, допускає перетворення вигляду $f(x) = kx + b$. Наприклад, відомі перетворення температурних шкал (Фаренгейта — Цельсія, Реомюра — Цельсія тощо).

Шкалу порядку називають ординальною. У цій шкалі деякі об'єкти стають еквівалентними. Допустимими перетворами такої шкали є перетворення, які не змінюють порядку.

Шкала назв (номінальна) найбільш слабка. Тут числа служать умовними назвами об'єктів вимірювань. Допустимими перетвореннями такої шкали є пермутаційні (перестановки).

Між метричними шкалами існують взаємні перетворення та відношення. Найбільш сильною шкалою є абсолютна шкала (вона включає в себе всі інші шкали).

Означення 12. Шкала — алгоритм гомоморфного відображення емпіричної системи з відношенням у числову систему.

На теоретичному формальному рівні вимірювання трактується лінійним перетворенням, оператором. Це трактування формально означається як еквівалентність (процедури вимірювання, фільтрації). Фільтр математично означається оператором. За виглядом розрізняють диференціальні, інтегральні, інтегрально-диференціальні оператори. Диференціальним оператором є, наприклад, звичайне диференціальне (та його дискретний аналог — різницеве) рівняння. На рис. 3.4 подано алгоритм вимірювання.

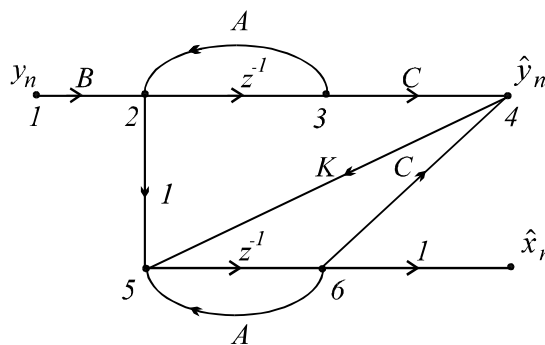


Рисунок 3.4 – Алгоритм Калмана-Бюсі оптимального вимірювання значення випадкового процесу

(A , B , C — задані матриці, елементи яких є параметрами вимірюваного об'єкту; y_n — вимірюваний стан об'єкту, вектор; x_n — оптимальна оцінка вектора стану; K — коефіцієнти Калмана, шукають в процесі обчислень за відповідним алгоритмом)

Для зашумлених даних (суміші даних та шуму) будують оптимальні алгоритми оцінювання значень вимірюваної величини. До таких алгоритмів належить, наприклад, оптимальний фільтр Колмогорова-Вінера. Його реалізують у вигляді трансверсального алгоритму (рис. 5), параметри a_i , $i=1, 2, 3, \dots, n$ якого шукають розв'язуючи відповідну задачу, постановка якої передбачає відомий вигляд суміші (адитивна) даних і "шуму" та відомі їх статистики (кореляційні функції чи спектри). Іншим є оптимальний фільтр Калмана-Бюсі. Алгоритм його роботи має характер ітерацій при пошуку кожної з оцінок значень даних.

Задавши об'єкт формальним описом (математичною моделлю) у просторі його станів вектором y , (компоненти вектора — величини, властиві об'єкту), результати вимірювання їх значень позначивши вектором x , а вплив різних факторів на стан хворого і виміри їх значень — u та w відповідно, отримаємо

$$dy = Ay + Bu dt, \quad x = Cy + Dw$$

де матриці відповідно визначають спосіб дії: B , D — вплив, C , A — вимірів та станів. Процес встановлення конкретного вигляду всіх цих позначень називають математичним моделюванням. Зокрема, їх певний вигляд відповідає постановці задачі фільтрації Калмана-Бюсі — оцінювання стану (об'єкту). Калманівська фільтрація є моделлю оцінювання (вимірювання) стану. Якщо процесор, виконує обчислення, необхідні для фільтру (розрахунок коефіцієнтів K розв'язуванням рівняння Рікати тощо), то отримаємо автоматичного оптимального спостерігача (технічного засобу, що виконує одну з функцій людини — оцінювання

стану керованого об'єкту). Для дискретних значень компонент векторів дістанемо модель у просторі станів

$$x_{n+1} = -1Cy = Ay_n + nDw + Bu_n,$$

де y_{n-1} — вектор з компонентами різниць першого порядку (дискретний аналог похідних). Для цієї моделі алгоритм пошуку компонент вектора подано на рис. 7 (де вплив шумів, похибок на вимір компонент вектора станів x_n відсутній).

2. Алгоритми керування

Виміряні дані використовують для керування. Поняття керування виникло давно. Наприклад, вже у стародавньому Китаї, Єгипті, древній Греції та старому Римі розуміли роль зворотного зв'язку та поінформованості при забезпеченні стабільності та потрібного розвитку різних процесів. Проте теорія керування розвинулася лише протягом XIX та XX століття, а практично широке застосування його розвинулося у другій половині XX століття.

Загалом керування розрізняють:

- (а) програмне (за наперед відомим алгоритмом з постійними параметрами);
- (б) автоматичне, коли враховуються вхідні дані;
- (в) адаптивне (коли параметри алгоритму змінюються залежно від зовнішніх умов, даних).

Адаптивне та автоматичне керування здійснюють використовуючи зворотній зв'язок. Коли, враховуючи різні обмеження, характеристика алгоритму автоматичного чи програмного керування за встановленим критерієм найкраща, то керування називають оптимальним.

Поняття зворотного зв'язку є одним з фундаментальних у кібернетиці — науці про ефективне керування (розрізняють технічну кібернетику та інші: медичну, криміналістичну, спортивну й т.п. — див., наприклад, книгу С. Бир Мозг фірми.—М.: Радио и связь, 1993. 416 с.). Іншим фундаментальним поняттям цієї науки є інформація. Воно

безпосередньо розвивалося в теорії комунікації (зв'язку). Ці поняття й розглядатимемо у контексті, прийнятому в цих технічних науках.

В широкому розумінні інформація — це якісь відомості, дані про конкретне явище. У техніці її, як правило, пов'язують зі змінною фізичною величиною, сигналом (та його математичною моделлю — функцією). Сигнал є фізичним носієм інформації. Таким чином, вимірюючи параметри та характеристики сигналу отримуємо інформацію якщо вони міняються, а їх інтерпретація відома.

Розглянемо елементарну систему керування (спеціальним чином з'єднані пристрої, рис. 3.5). Феноменологічно такі системи були зауважені у природі ще в древності. Задля забав їх виконували (імітували, повторювали) різними технічними засобами. Пізніше вони використовувалися у годинниках, парових машинах в регуляторах (стабілізаторах) швидкості обертання валу механізму.

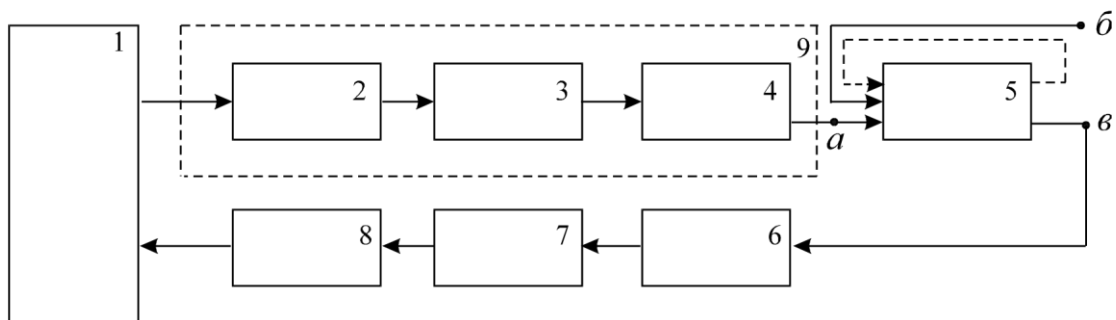


Рисунок 3.5 – Блок-схема системи керування.

Елементарна система керування складається з ланок: пристрою порівняння (5), регулятора (6, 7, 8), зворотного зв'язку (2, 3, 4). Сигнал $x(t)$ від керованого об'єкту (1) та відповідні характеристики ланок системи у лінійній теорії сигналів та систем подають як функції комплексної змінної $p=j\omega$, де $\omega=2\pi f$ (інакше – як лапласівські образи функцій часу, який є дійсною змінною), f — частота, гц. Нагадаймо, що для ланок системи цими характеристиками є тоді лапласівські образи їх вагових функцій (відгуків на дельта-збурення) — функції передачі регулятора $K_p(p) = K_6(p) K_7(p) K_8(p)$ та зворотнього зв'язку $K_{зв}(p) = K_2(p) K_3(p) K_4(p)$. Це приводить до

зручнішого подання залежності між вхідним сигналом $X(p)$ (на вході блоку 2) та вихідним — на виході блоку 8, $Y(p)$:

$$Y(p) = (X(p) + Y(p)K_{зв}(p))K_p(p),$$

або у наочнішому для користування вигляді

$$Y(p) = X(p) \frac{K^{р(р)}}{1 - K_{зв}(p)K_p(p)}.$$

Вирази $K_{р,зв}(p)$ є алгебричними поліномами вигляду $\sum k_i p^i$ (вони називаються функціями передачі). Від їх вигляду (порядку N , значень коефіцієнтів k_i) залежить стійкість системи.

Для дискретних систем (і в основному й для цифрових, коли дискретні значення сигналів кодуються двійковим чи іншим кодом) матимемо подібний вираз, але як функцію від $z = \exp(j2\pi fT_d)$, де T_d — період дискретизації неперервного сигналу (фіксовані інтервали часу для виконання операцій в алгоритмі)

$$Y(z) = X(z) \frac{K^{р(z)}}{1 - K_{зв}(z)K_p(z)}.$$

Цей вираз є розв'язком системи лінійних рівнянь, складених за рис. 8 з використанням законів дискретних кіл (мереж) чи властивостей відповідного йому графу. Таким чином, чисельник цього виразу є доповненням до елемента з індексами рівними номерам вузлів входу-виходу матриці системи рівнянь, а знаменник — детермінантом цієї системи.

Функції $K_{р,зв}(z)$ у загальному випадку дробово-раціональні, їх обчислення виконується за алгоритмами, подібними на алгоритм ланки управління. Вони за структурою подібні до алгоритмів вимірювання.

На множині номерів вузлів дискретного (цифрового) кола моментом часу обчислення задається відношенням порядку. Множина вузлів розбивається на підмножини (за відношенням еквівалентності). В свою чергу, на цих підмножинах також задається часове відношення порядку. Виникають тим самим умови для паралелення та конвеєризації обчислень.

В алгоритмах векторно-матричних обчислень спостерігається залежність кількості зайнятих в обчисленнях ресурсів від часу, яку за аналогією з хвилями в кровоносній системі організмів називають систолічністю. Структури, в яких з метою підвищення їх ефективності враховуються ці особливості, називають систолічними. Ці властивості використовують під час побудови оптимальних за швидкістю чи кількістю обладнання алгоритмів.

3.2. Розробка алгоритму роботи тренажеру

Крок 0.

Виводиться стартове вікно з темою роботи, надається вибір відкрити теорію або розпочати проходження тренажеру.

Якщо вибрано теорію, то відкриється файл з теоретичним матеріалом.

Якщо вибрано розпочати тренажер, то перехід до кроку 1.

Крок 1.

Виводиться запитання і варіанти відповіді:

Що таке алгоритм?

А) Скінченна сукупність коректно сформульованих правил розв'язування задачі з класу (множини) задач.

Б) Нескінченна сукупність коректно сформульованих правил розв'язування задачі з класу (множини) задач.

Якщо вибрано правильну відповідь, тобто варіант А, то перехід на наступний крок.

Якщо вибрано неправильну відповідь – повідомлення з правильною: Правильна відповідь – «А».

Крок 2.

Виводиться запитання і варіанти відповіді:

Для обчислень значень інформативних величин, та для керування потрібні?

- А) Дані.
- Б) Системи.
- В) Алгоритми.
- Г) Множини.

Якщо вибрано правильну відповідь, тобто варіант А, то перехід на наступний крок.

Якщо вибрано неправильну відповідь – повідомлення з правильною: Правильна відповідь – «А».

Крок 3.

Виводиться запитання і варіанти відповіді:

Що таке рекурсія?

А) Процедура обчислень коли існує її алгоритм називається рекурсією.

Б) Сукупність числових функцій співпадаюча з сукупністю ефективно вичислимих функцій називається рекурсією.

В) Рекурсією називається спосіб завдання функції, при якому біжуче значення функції визначається через попередні для меншого аргументу.

Г) Числова функція, значення якої можна обчислити називається рекурсією.

Якщо вибрано правильну відповідь, тобто варіант В, то перехід на наступний крок.

Якщо вибрано неправильну відповідь – повідомлення з правильною: Правильна відповідь – «В».

Крок 4.

Виводиться запитання і варіанти відповіді:

Що таке шкала?

А) Числова функція, значення якої можна обчислити називається шкалою.

Б) Алгоритм гомоморфного відображення емпіричної системи з відношенням у числову систему.

В) Процедура обчислень коли існує її алгоритм називається шкалою.

Г) Шкалою називається спосіб завдання функції, при якому біжуче значення функції визначається через попередні для меншого аргументу.

Якщо вибрано правильну відповідь, тобто варіант Б, то перехід на наступний крок.

Якщо вибрано неправильну відповідь – повідомлення з правильною: Правильна відповідь – «Б».

Крок 5.

Виводиться запитання і варіанти відповіді:

Яка процедура обчислення називається ефективною?

А) Коли існує її алгоритм.

Б) Значення якої можна обчислити.

В) Яка співпадає з сукупністю ефективно вичислимих функцій.

Г) Всі варіанти відповіді вірні.

Якщо вибрано правильну відповідь, тобто варіант А, то перехід на наступний крок.

Якщо вибрано неправильну відповідь – повідомлення з правильною: Правильна відповідь – «А».

Крок 6.

Виводиться запитання і варіанти відповіді:

Всі команди розпадаються на?

А) Логічні та нелогічні.

Б) Алгоритми та системи.

В) Алгоритми та множини.

Г) Логічні та арифметичні.

Якщо вибрано правильну відповідь, тобто варіант Г, то перехід на наступний крок.

Якщо вибрано неправильну відповідь – повідомлення з правильною:
Правильна відповідь – «Г».

Крок 7.

Виводиться запитання і варіанти відповіді:

Що таке математична модель?

А) Це кількісна характеристика, що відображує споживані алгоритмом ресурси під час свого виконання.

Б) Система математичних співвідношень, які описують досліджуваний процес або явище.

В) Числова функція.

Г) Алгоритм гомоморфного відображення емпіричної системи з відношенням у числову систему.

Якщо вибрано правильну відповідь, тобто варіант Б, то перехід на наступний крок.

Якщо вибрано неправильну відповідь – повідомлення з правильною:
Правильна відповідь – «Б».

Крок 8.

Виводиться запитання і варіанти відповіді:

Областю застосовності алгоритму називається?

А) Сукупність тих об'єктів, до яких його можна застосувати.

Б) Значення яке можна обчислити.

В) Процедура обчислень.

Г) Числова функція.

Якщо вибрано правильну відповідь, тобто варіант А, то перехід на наступний крок.

Якщо вибрано неправильну відповідь – повідомлення з правильною:
Правильна відповідь – «А».

Крок 9.

Виводиться запитання і варіанти відповіді:

Клас рекурсивних функцій тотожний класові скрізь означених вичислимих функцій?

А) Довести можливо.

Б) Довести неможливо.

Якщо вибрано правильну відповідь, тобто варіант Б, то перехід на наступний крок.

Якщо вибрано неправильну відповідь – повідомлення з правильною: Правильна відповідь – «Б».

Крок 10.

Виводиться запитання і варіанти відповіді:

Шкала інтервалів метрична частково, допускає перетворення вигляду?

А) $\phi(x) = kx - b$.

Б) $\phi(y) = kx + x$.

В) $\phi(x) = kx - v$.

Г) $\phi(x) = kx + b$.

Якщо вибрано правильну відповідь, тобто варіант Г, то перехід на наступний крок.

Якщо вибрано неправильну відповідь – повідомлення з правильною: Правильна відповідь – «Г».

Крок 11.

Виводиться запитання і варіанти відповіді:

Які властивості притаманні дискретності?

А) Алгоритм розв'язує задачу, що належить до класу задач.

Б) Перед виконанням першої операції відомі початкові дані, у будь-який момент часу точно відомі наступні операції (прогнозованість) та результат попередньої операції.

В) Виконання чергової операції починається у певний момент часу, моменти часу не обов'язково еквідистантні.

Г) Встановлена черговість операції (інакше — відношення порядку на їх множині).

Якщо вибрано правильну відповідь, тобто варіант В, то перехід на наступний крок.

Якщо вибрано неправильну відповідь – повідомлення з правильною: Правильна відповідь – «В».

Крок 12.

Виводиться запитання і варіанти відповіді:

Які властивості притаманні детермінованості?

А) Алгоритм розв'язує задачу, що належить до класу задач.

Б) Перед виконанням першої операції відомі початкові дані, у будь-який момент часу точно відомі наступні операції (прогнозованість) та результат попередньої операції.

В) Завжди можна вказати елементарну (виконувану апаратно, не програмовану) операцію — називатимемо алгоритмом.

Г) Встановлена черговість операції (інакше — відношення порядку на їх множині).

Якщо вибрано правильну відповідь, тобто варіант Б, то перехід на наступний крок.

Якщо вибрано неправильну відповідь – повідомлення з правильною: Правильна відповідь – «Б».

Крок 13.

Виводиться запитання і варіанти відповіді:

Які властивості притаманні впорядкованості?

А) Алгоритм розв'язує задачу, що належить до класу задач.

Б) Перед виконанням першої операції відомі початкові дані, у будь-який момент часу точно відомі наступні операції (прогнозованість) та результат попередньої операції.

В) Завжди можна вказати елементарну (виконувану апаратно, не програмовану) операцію — називатимемо алгоритмом.

Г) Встановлена черговість операції (інакше — відношення порядку на їх множині).

Якщо вибрано правильну відповідь, тобто варіант Г, то перехід на наступний крок.

Якщо вибрано неправильну відповідь – повідомлення з правильною: Правильна відповідь – «Г».

Крок 14.

Виводиться запитання і варіанти відповіді:

Які властивості притаманні масовості?

А) Алгоритм розв’язує задачу, що належить до класу задач.

Б) Перед виконанням першої операції відомі початкові дані, у будь-який момент часу точно відомі наступні операції (прогнозованість) та результат попередньої операції.

В) Завжди можна вказати елементарну (виконувану апаратно, не програмовану) операцію — називатимемо алгоритмом.

Г) Встановлена черговість операції (інакше — відношення порядку на їх множині).

Якщо вибрано правильну відповідь, тобто варіант А, то перехід на наступний крок.

Якщо вибрано неправильну відповідь – повідомлення з правильною: Правильна відповідь – «А».

Крок 15.

Виводиться запитання і варіанти відповіді:

Які властивості притаманні масовості?

А) Алгоритм розв’язує задачу, що належить до класу задач.

Б) Перед виконанням першої операції відомі початкові дані, у будь-який момент часу точно відомі наступні операції (прогнозованість) та результат попередньої операції.

В) Завжди можна вказати елементарну (виконувану апаратно, не програмовану) операцію — називатимемо алгоритмом.

Г) Встановлена черговість операції (інакше — відношення порядку на їх множині).

Якщо вибрано правильну відповідь, тобто варіант В, то перехід на наступний крок.

Якщо вибрано неправильну відповідь – повідомлення з правильною: Правильна відповідь – «В».

Крок 16.

Виводиться запитання і варіанти відповіді:

Проаналізуймо властивості послідовності операцій алгоритму Евкліда:

- хii. ВВЕСТИ більше число у реєстр x_2 ;**
- хiii. ВВЕСТИ менше число у реєстр x_1 ;**
- хiv. Функція — ціла частина q_0 частки від ділення вмісту реєстрів x_2 на x_1 зі знаком мінус;**
- хv. Добуток x_0 вмісту реєстрів x_1 та q_0 ;**
- хvi. Сума x_0 вмісту реєстру x_0 з вмістом реєстру x_2 ;**
- хvii. Переслати вміст реєстру x_1 у реєстр x_2 ;**
- хviii. Переслати вміст реєстру x_0 у реєстр x_1 ;**
- хix. Переслати нуль в реєстр x_0 ;**
- хx. Якщо x_1 не рівне нулеві — перейти на крок iii;**
- хxi. ВИВЕСТИ вміст реєстру x_2 .**
- хxii. ЗУПИНИТИ.**

А) Видно, що операції **i-iv** та **vi-viii** є операціями алгебри чисел. Інші операції (наприклад, **v** та **ix**) — ні. Це, а також впорядкування операцій за часом, не дозволяє означити алгебричну структуру і аналітичну модель алгоритму при заданій математичній моделі задачі, множині елементарних команд процесора.

Б) Видно, що операції v та ix є операціями алгебри чисел. Інші операції (наприклад, $i-iv$) — ні. Це, а також впорядкування операцій за часом, не дозволяє означити алгебричну структуру і аналітичну модель алгоритму при заданій математичній моделі задачі, множині елементарних команд процесора.

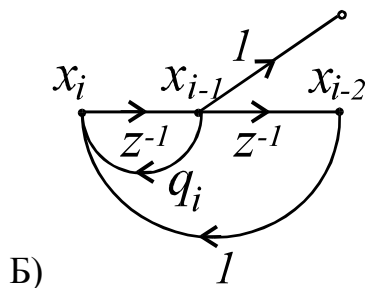
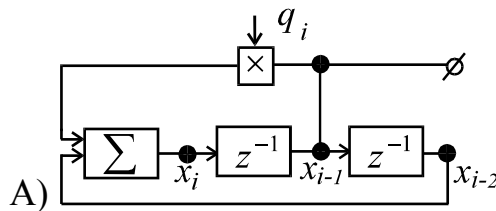
Якщо вибрано правильну відповідь, тобто варіант А, то перехід на наступний крок.

Якщо вибрано неправильну відповідь – повідомлення з правильною: Правильна відповідь – «А».

Крок 17.

Виводиться запитання і варіанти відповіді:

З теорії цифрових кіл відомі функціональні блок-схеми. У їх позначеннях подаються алгоритми обчислень:



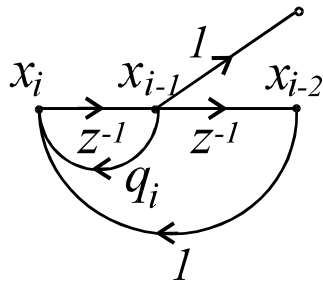
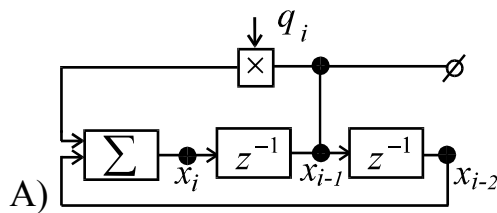
Якщо вибрано правильну відповідь, тобто варіант А, то перехід на наступний крок.

Якщо вибрано неправильну відповідь – повідомлення з правильною: Правильна відповідь – «А».

Крок 18.

Виводиться запитання і варіанти відповіді:

Безпосередньо за функціональною блок-схемою будується граф цифрового кола:



Якщо вибрано правильну відповідь, тобто варіант Б, то перехід на наступний крок.

Якщо вибрано неправильну відповідь – повідомлення з правильною: Правильна відповідь – «Б».

Крок 19.

Виводиться повідомлення про кінець:

Вітаємо! Ви завершили виконання тренажеру з теми «Математичні основи теорії алгоритмів» дистанційного начального курсу «Теорія програмування».

Надається вибір перейти на стартове вікно і розпочати знову або закрити тренажер.

3.3. Блок-схема, яка підлягає програмуванню

На рисунку 3.6 зображено блок-схему алгоритму роботи тренажеру.

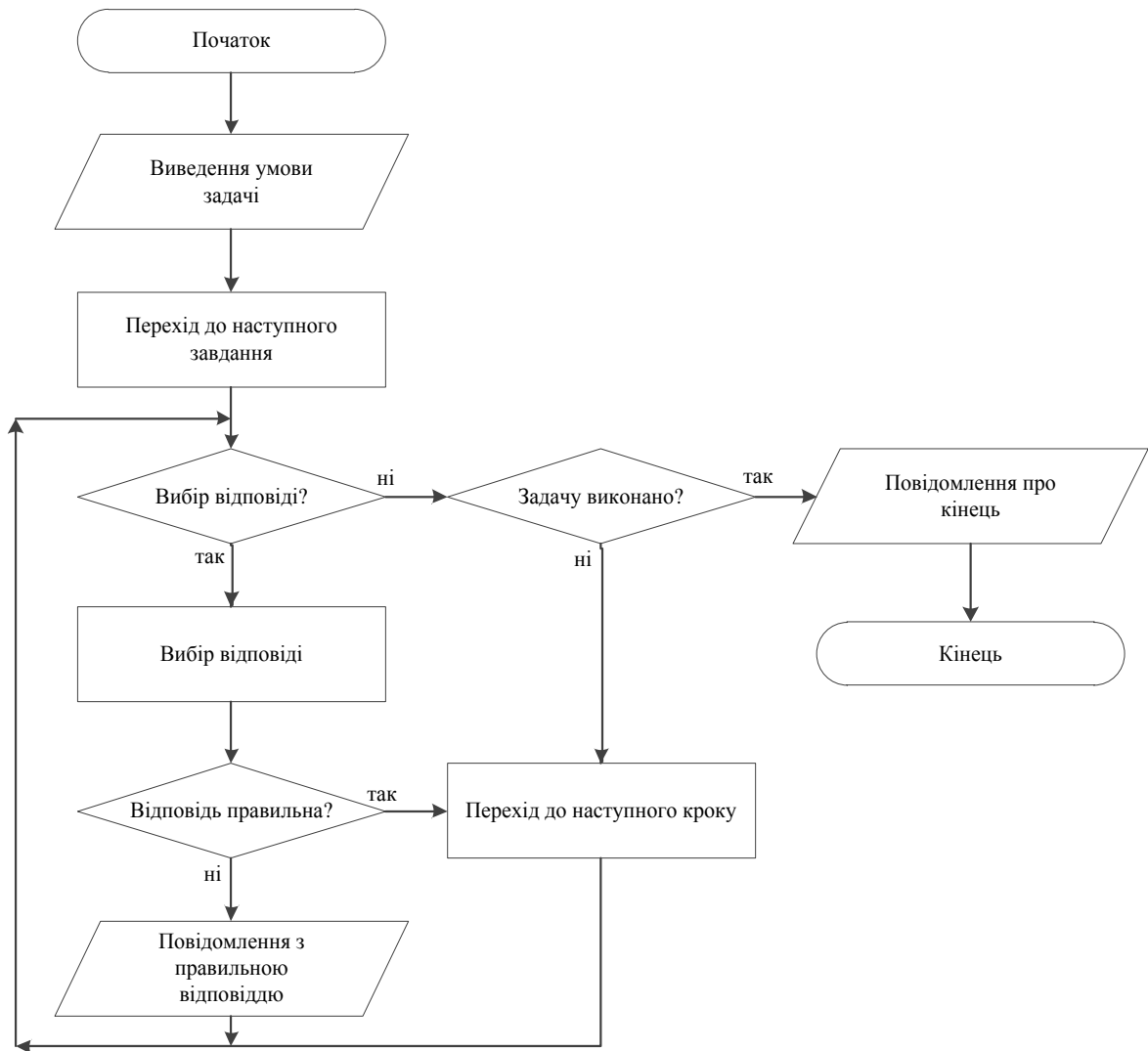


Рисунок 3.6 – Блок-схема алгоритму роботи тренажеру

4. ПРАКТИЧНА ЧАСТИНА

4.1. Обґрунтування вибору програмних засобів

Java є мовою програмування, за допомогою якої розробники програмного забезпечення (програмісти) створюють різні прикладні додатки для комп'ютерів, смартфонів, планшетів та інших інтелектуальних пристроїв. Особливістю програм на Java є те, що вони можуть запускатись на будь-яких комп'ютеризованих пристроях, які працюють під різними операційними системами, причому без повторної компіляції коду.

Для їх виконання необхідно лише встановити середовище для виконання – JRE (Java Runtime Environment), завантаживши його з сайту <http://www.oracle.com>. JRE розроблені для багатьох операційних систем – Linux(x86,x64), Mac OS X64, Solaris, Windows (x86,x64), завдяки чому код Java працює майже на всіх різновидах комп'ютерів та операційних систем.

JRE забезпечує безпечну та зручну роботу додатків на Java, тому користувачі можуть не турбуватись про несанкціоноване втручання до ресурсів свого персонального комп'ютера з боку стороннього Java коду. Необхідно лише періодично оновлювати JRE, на даний момент остання версія JRE 8 Update 171. Вбудована технологія забезпечення безпеки Java включає в себе значний набір API (Application Programming Interface) механізмів та додаткових інструментів, включаючи широковідомі та надійні алгоритми та протоколи безпеки.

Це передбачає використання криптографічних механізмів захисту інформації, інфраструктуру відкритих ключів, захищений зв'язок, автентифікацію та контроль доступу.

Однієї із найбільш цікавих для користувачів Java є платформа JavaFX. JavaFX являє собою набір графічних та медіа пакетів (бібліотек Java), які дозволяють програмістам проектувати, створювати, тестувати та

розгорнути RIA (Rich Client Applications) додатки, які фактично являють собою програми із графічним інтерфейсом, які виглядають та працюють однаково на всіх платформах та пристроях. Архітектура платформи JavaFX представлена на малюнку нижче.

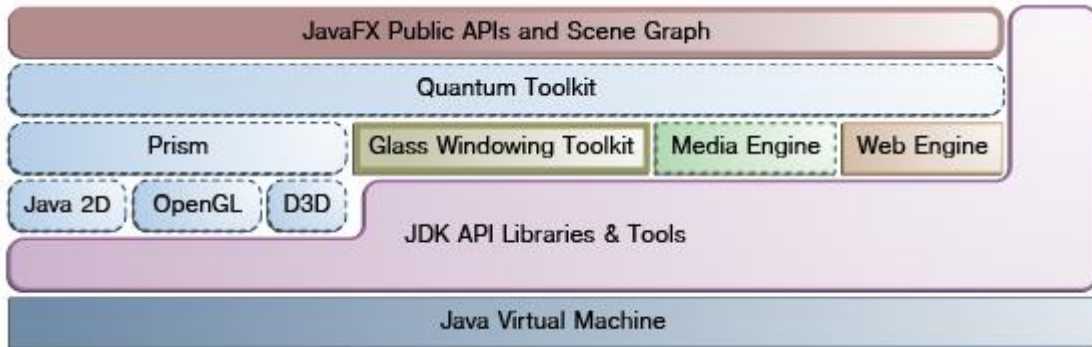


Рисунок 4.1 – Архітектура платформи JavaFX

Цікавою особливістю є те, що вигляд та поведінку програм, написаних на JavaFX, можна налаштовувати за допомогою каскадних таблиць стилів (CSS – Cascading Style Sheets). Цей підхід, який історично був першим застосований для веб-сайтів, дозволяє відокремити зовнішній вигляд програм (інтерфейс користувача) від реалізації, що дозволяє програмістам концентруватись на кодуванні. Турбота про графічний інтерфейс тепер лежить на плечах графічних дизайнерів, які налаштовують зовнішній вигляд за допомогою скриптової мови FXML та технології CSS, а програмісти зосереджені на розробці бізнес-логіки додатка. Нижче наведено вигляд додатку із різними налаштуваннями CSS.

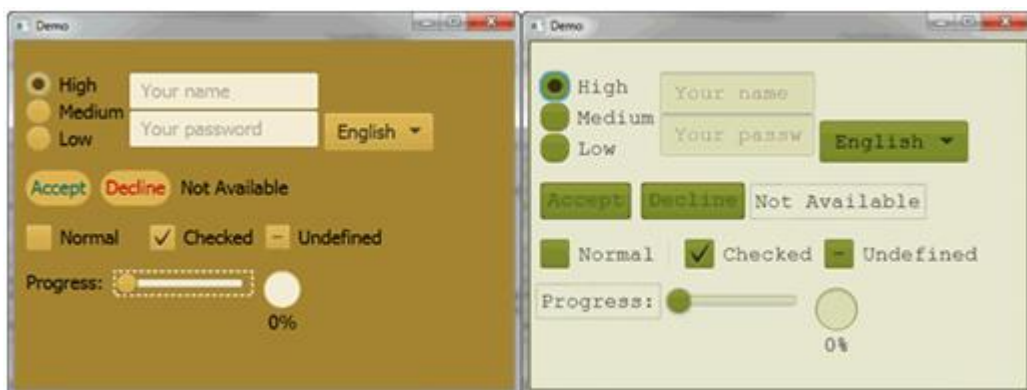


Рисунок 4.2 – Архітектура платформи JavaFX

Безумовно, наведений приклад є досить простим, і може виникнути питання – невже за допомогою мови Java та JavaFX можна розробляти тільки такі прості програми?

Звичайно ж не тільки такі! Типова Java програма є клієнт-серверним додатком, що працює як на сервері (де зосереджена бізнес-логіка додатку) та на клієнтських пристроях, які підключаються до сервера за допомогою мережі Інтернет. Платформа JavaFX дозволяє проектувати високопродуктивні інтерфейси користувача на клієнтських додатках, які окрім базових графічних компонентів, включають відео, звук, анімацію.

- Так де ж використовується Java? А й справді – де?
- Microsoft Office (Excel, Word і т.д.) – не на Java написаний
- Adobe Acrobat – не на Java написаний
- YouTube – не на Java написаний
- Google Chrome – не на Java написаний
- операційні системи Windows, Linux, MacOS – не на Java написані
- комп'ютерні ігри – теж не на Java (за винятком Minecraft)

І де ж тоді Java застосовується? Адже, згідно з даними Oracle, понад 3 мільярди приладів працюють на Java. А ось і відповідь:

- Java, насамперед, – це мова, яка використовується для серверних додатків у великих корпораціях. Особливо часто Java використовується в банках, страхових компаніях, роздрібних мережах і т.д. Наприклад, такі банки, як Deutsche Bank, Citigroup, Barclays, Goldman Sachs і багато-багато інших, використовують Java для написання бек-енд і фронт-енд офісних електронних систем тощо. Ще один приклад: серверна частина Google+ написана на Java.

- Веб-додатки. Широке застосування Java знайшла в електронній комерції і веб-додатках. Наприклад, якщо говорити про ринок Європи і США, багато веб-додатків держустанов, страхових, освітніх й оборонних установ написані на Java.

- Трейдингові додатки. Наприклад, LMAX, Murex.
- Android додатки. Якщо у Вас телефон на Android, то всі програми в ньому написані на Java з використанням Google і Android API.
- Desktopні додатки, програмні засоби і засоби розробки. Наприклад, Eclipse, Netbeans IDE, jEdit (Programmer's Text Editor), jDownloader (open-source download management tool), Azureus/
- Вбудовані системи. Java використовується на смарт-картах і сенсорах. Наприклад, у банківських картках використовується Java.

4.2. Розробка тренажера

Основний код тренажера з теми «Математичні основи теорії алгоритмів» написаний в java за допомогою Buffered Reader. Для того щоб зайти в тест, потрібно відкрити java.

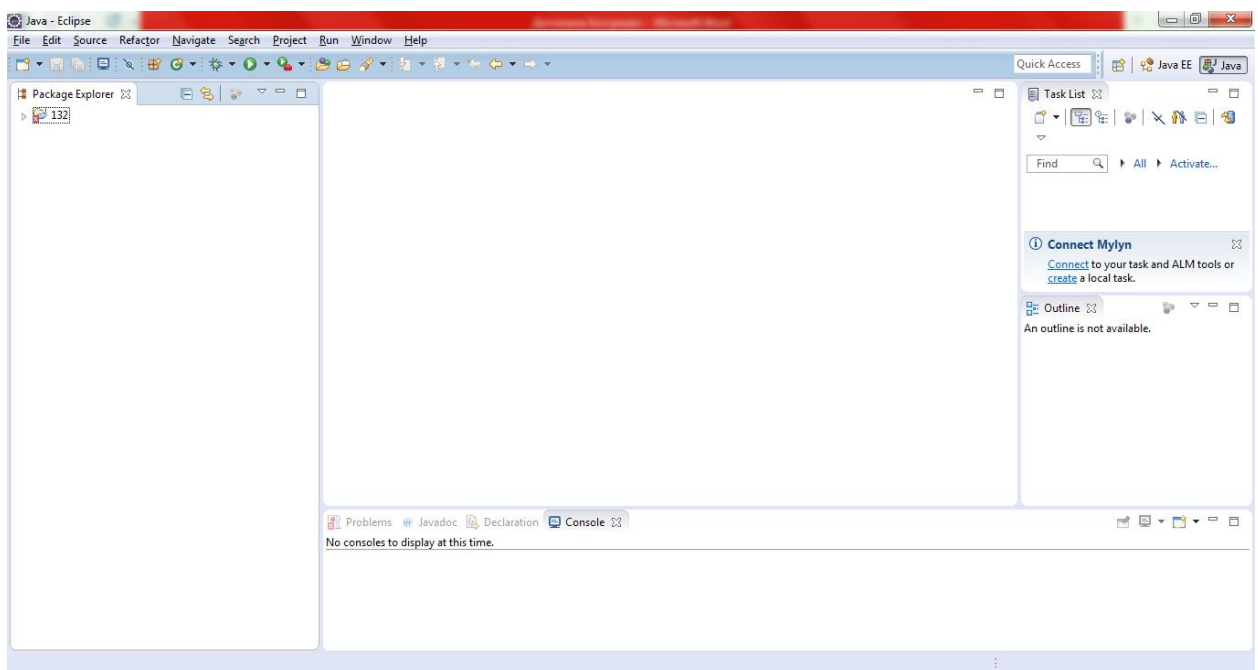


Рисунок 4.3 – Buffered Reader

Далі потрібно створити клас та скопіювати код.

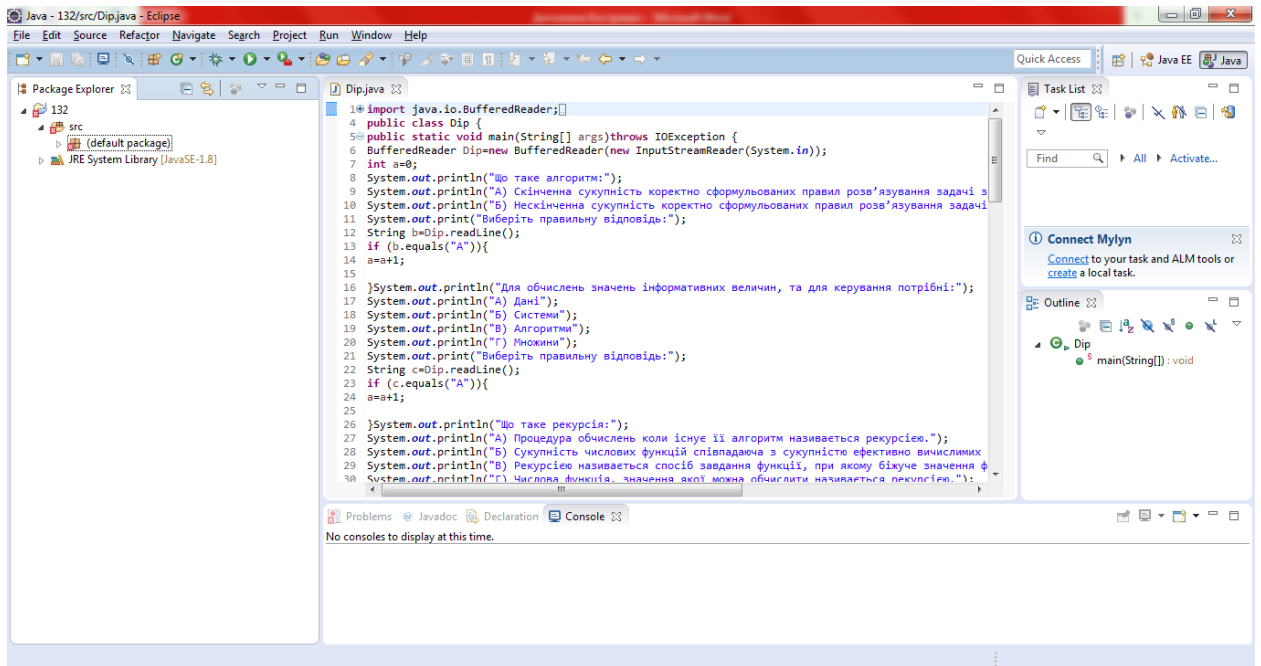


Рисунок 4.4 – Створити клас та скопіювати код

Далі потрібно натиснути клавішу «Run»

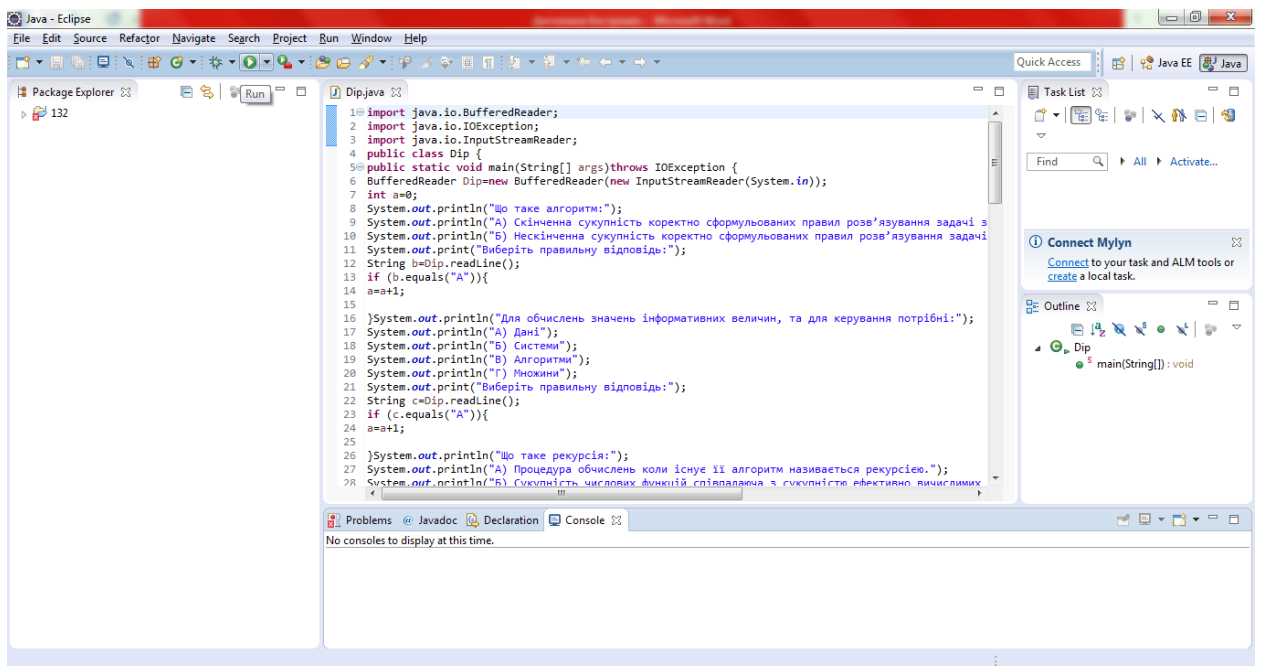


Рисунок 4.5 – Натиснути клавішу «Run»

В компіляторі тепер потрібно вибрати варіанти відповіді, потрібно вибрати одну з літер «А», «Б», «В» або «Г».

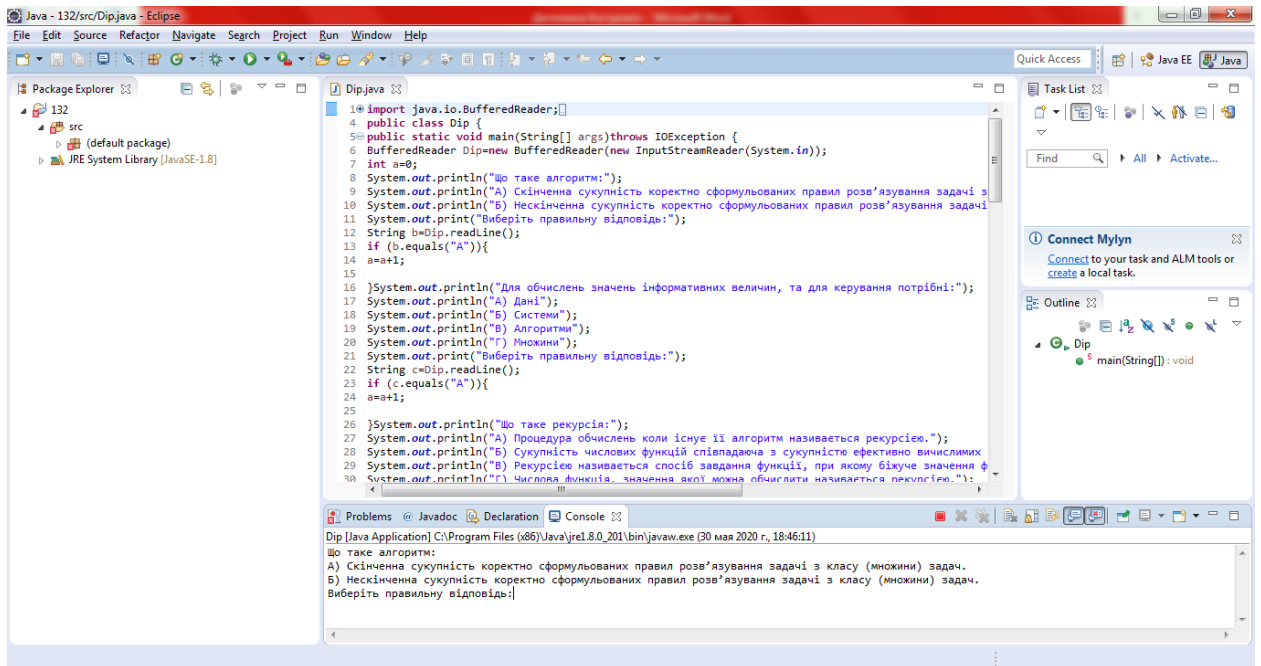


Рисунок 4.6 – Потрібно вибрати варіанти відповіді

Якщо відповідь вірна, буде зачислена +1 бал

В кінці після закінчення усіх питань буде виведено кількість правильних відповідей з 0 по 11.

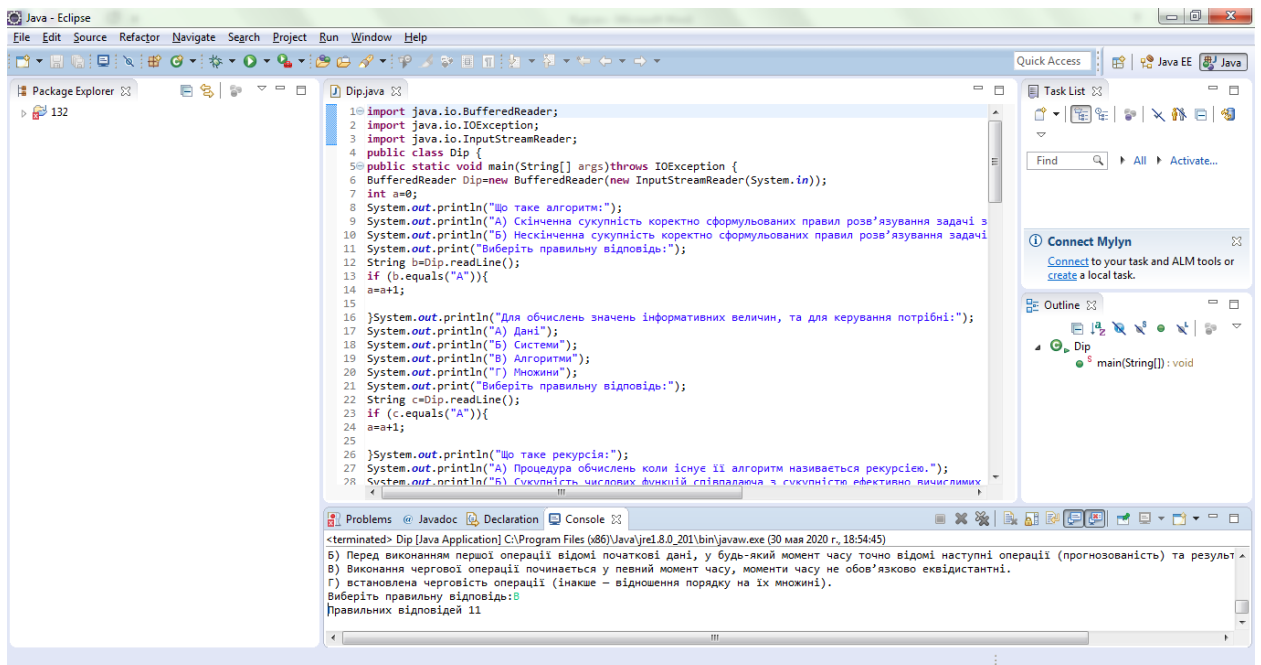


Рисунок 4.7 – Кількість правильних відповідей

ВИСНОВКИ

За результатами виконання дипломної роботи можна зробити такі висновки:

1. Закріплено лекційний матеріал по темі дистанційного навчання. Виконані поставлені задачі щодо оформлення та виконання курсової роботи. Виконана поставлена задача по створенню тренажера з теми «Математичні основи теорії алгоритмів».

2. Розглянуто теоретичні відомості по темі «Математичні основи теорії алгоритмів». Поставлено практичні завдання та розроблений алгоритм програми.

3. Виконано розбір розроблення програмного. Повністю проведено тестування програми по всім запитанням. Написано інструкцію по користуванню програмою.

В алгоритмі описано роботу тренажера:

Виводиться стартове вікно з темою роботи, надається вибір відкрити теорію або розпочати проходження тренажера.

Якщо вибрано теорію, то відкриється файл з теоретичним матеріалом.

Якщо вибрано розпочати тренажер, то перехід до кроку 1.

На останньому кроці виводиться повідомлення про кінець:

Вітаємо! Ви завершили виконання тренажера з теми «Математичні основи теорії алгоритмів» дистанційного начального курсу «Теорія програмування».

Надається вибір перейти на стартове вікно і розпочати знову або закрити тренажер.

В практичній частині наведено процес програмної реалізації.

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. Грин Д., Кнут Д. Математические методы анализа алгоритмов.- М.: Мир, 1987.- 120 с.
3. Алферова. З.В. Теория алгоритмов.- М.: Статистика, 1973.- 164 с.
4. Кириллов А.А. Элементы теории представлений. - М.: Наука,1972.336с.
5. Лавров И.А., Максимова Л.Л. Задачи по теории множеств, математической логике и теории алгоритмов. – М.: Наука, 1984. – 224с.
6. Драган Я.П., Сікора Л.С., Яворський Б.І. Основи сучасної теорії стохастичних сигналів: енергетична концепція, математичний апарат, фізичне тлумачення. - Львів: ЕБТЕС, 1999.- 132 с.
7. Кузьмин И.В., Березюк И.Т., Фурманов К.К., Шаронов В.Б. Синтез вычислительных алгоритмов управления и контроля. – Л.: Техніка, 1975. – 248 с.
8. Яворський Б. І. Математичні основи радіоелектроніки/ В 3-х частинах. – Тернопіль: ТПШ, 1996. – 336 с.
9. Вирт Н. Алгоритмы + структуры данных = программы.— М.: Мир, 1985.-406 с.
10. Громов Г.Р. Очерки информационной технологии.— М.: ИнфАрт, 1992. – 336 с.
11. Сигорский В.П. Математический аппарат инженера.— К.: Техника, 1975.— 768 с.
12. Томович Р., Вукобратович М. Общая теория чувствительности.— М.: Сов. радио, 1972.—240 с.

13. Александров А.Г. Оптимальные и адаптивные системы.— М.: Высш. школа, 1989.— 263 с.
14. Сергиенко И.В. Математические модели и методы решения задач дискретной оптимизации.— Киев: Наук. думка, 1988.— 472 с.
15. Капітонова Ю.В., Кривий С.Л., Летичевський О.А. та ін. Основи дискретної математики.— Київ: Наук. думка, 2002.— 560 с.
16. Брукшир Дж. Гленн Введение в компьютерные науки.— М.: Издательский дом "Вильямс", 2001.— 668 с.
17. 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