МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет комп'ютерних наук, фізики та математики
Кафедра інформатики, програмної інженерії та економічної кібернетики

СТВОРЕННЯ ПРОГРАМНОЇ СИСТЕМИ НАВЧАЛЬНОГО ПРИЗНАЧЕННЯ З ПРЕДМЕТНОГО МОДУЛЯ «ПРОГРАМУВАННЯ МАТЕМАТИЧНИХ ЗАДАЧ»

Кваліфікаційна робота (проєкт)
на здобуття ступеня вищої освіти «магістр»

Виконала: студентка 2 курсу
Спеціальності 121 Інженерія програмного забезпечення
Освітньо-професійної програми «Інженерія програмного забезпечення»
другого (магістерського) рівня вищої освіти
Панова Катерина Олександрівна
Керівники доктор фізико-математичних наук,
професор Львов Михайло Сергійович
кандидат фізико-математичних наук,
доцент Єрмолаєв Вадим Анатолійович,
Рецензент кандидатка педагогічних наук,
старша викладачка Григор'єва Валентина Борисівна

Херсон – 2020

ЗМІСТ

ВСТУП 3

РОЗДІЛ 1. Теоретичні аспекти програмування математичних задач 7

1.1 Методи вивчення навчального модуля «Програмування математичних задач» 7

1.2 Огляд існуючих програмних засобів для вивчення програмування 13

РОЗДІЛ 2. Математичне моделювання програмної системи навчального призначення 23

2.1 Обґрунтування вимог до програмної системи 23

2.2 Огляд інструментів та технологій розробки 29

2.3 Проєктування та реалізація програмної системи навчального призначення 33

ВИСНОВКИ 44

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 46

ДОДАТКИ 51

Додаток А 51

ВСТУП

Актуальність дослідження. Точні та природничі науки посідають особливе місце з-поміж навчальних дисциплін. Вони формують фундаментальні наукові знання, які засновані на точних математичних моделях явищ і процесів як у природі, так і в суспільстві [30]. Учням, які вивчають інформатику на поглибленому рівні, варто оволодіти не лише вмінням написання програмних кодів певною мовою програмування, а й отримати практичні навички алгоритмізації та програмування, використовуючи при цьому сучасні засоби та ІКТ [37].

З таких умов основний аспект учнівської діяльності переноситься на створення алгоритмів і програм. Мета навчання основ алгоритмізації – навчити основних способів організації операцій і даних, а також застосування базових алгоритмічних конструкцій при складанні описів алгоритмів розв’язування різноманітних задач [26].

Проблема застосування педагогічних технологій та педагогічно орієнтованих програмних систем з позиції предметно-орієнтованого підходу вивчалася такими науковцями, як Співаковський О.В., Львов М.С., Кравцов, Г.М., Биков В. Ю., Лапінський В. В. та ін. [17; 31; 32; 44; 45; 51].

Питання вдосконалення змісту і засобів навчання алгоритмізації та програмування розглянуто у численній кількості праць вітчизняних дослідників [18; 19; 24-29; 37; 41; 49; 54].

У нинішній системі освіти перевірка освоєння учнями певної предметної галузі та якості засвоєного матеріалу здійснюється шляхом проведення різноманітних конкурсів та олімпіад [47]. Найчастіше для школярів програмування математичних задач зустрічається на олімпіадах з інформатики та конкурсах з програмування, відповідно, під час підготовки до них діти також вивчають основи алгоритмізації і

програмування. Також алгоритмізацію вивчають студенти на початкових курсах у закладах вищої освіти (інститути, університети, академії), адже розв'язування навчальних задач – це один з найбільш ефективних способів навчитися застосовувати свої знання на практиці. Окремо варто вказати олімпіади та конкурси з програмування для студентів, де завдання також найчастіше орієнтовані саме на програмування математичних задач. Відповідно, програмування математичних задач в умовах олімпіади або змагання матиме певні особливості. Важливим у навчанні програмуванню математичних задач є власне задачі, на яких і можна тренуватися. Отже, мета розв'язування задач полягає не лише в одержанні відповіді, але ще й в опануванні процесу, способів її знаходження. З огляду на потребу в таких ресурсах, почали з'являтися сайти з інтерактивними архівами задач. Тобто, в них можна не просто отримати доступ до тексту задачі, а ще є можливість відправити рішення на перевірку та отримати миттєву реакцію системи. Такі ресурси ведуть підрахунок або ж рейтинг учасників за числом розв'язаних задач.

На сьогодні існує велика кількість доступних ресурсів для вивчення будь-якої дисципліни у мережі Інтернет, у тому числі і для програмування. Однак більшість ресурсів, присвячених основам програмування, приділяють досить небагато уваги дослідженню базових алгоритмів, і відразу орієнтують користувача на вивчення синтаксису та семантики мов чи технологій програмування в умовах конкретних практичних ситуацій.

Сьогодні в рамках шкільної програми з інформатики учні розвивають логіку, розумові здібності тощо. Навчання учнів основам алгоритмізації та програмування зводиться до формування та розвитку алгоритмічного, логічного і математичного мислення.

Використання сучасних програмних засобів навчального призначення дозволяє покращити підготовку учнів у галузі

алгоритмізації та програмування, зробити процес вивчення навчальних модулів більш наочним та динамічним, а , отже, ефективнішим [35].

Отже, для навчання учнів основам алгоритмізації та програмування і, відповідно, формування та розвитку їх алгоритмічного, логічного і математичного мислення є практична потреба у системі педагогічного призначення, яка б дозволяла забезпечити самостійне засвоєння знань та формування практичних навичок програмування навчальних задач практичного змісту із предметної області математики та основ комп'ютерної алгебри як у навчальному процесі в закладі освіти, так і у процесі індивідуальної роботи, що зумовило вибір теми дослідження **«Створення програмної системи навчального призначення з предметного модуля «Програмування математичних задач»».**

Мета дослідження – спроєктувати та розробити програмну систему навчального призначення з предметного модуля «Програмування математичних задач».

Об'єкт дослідження – програмні засоби навчального призначення з програмування.

Предмет дослідження – програмна система навчального призначення з програмування математичних задач.

Відповідно до мети, об'єкту та предмету дослідження було сформульовано такі **завдання дослідження:**

1. Виконати теоретичний аналіз навчально-методичної (у тому числі періодичної) літератури з теми дослідження.
2. Проаналізувати існуючі програмні продукти навчального призначення з програмування, навести їх порівняльну характеристику.
3. Визначити вимоги програмної системи навчального призначення;
4. Описати інструменти та технології розробки програмного засобу та обґрунтувати їх вибір.

5. Створити модель програмної системи навчального призначення для вивчення програмування.
6. Виконати реалізацію програмного продукту.

У ході наукового дослідження були використані наступні загальнонаукові **методи дослідження**: аналіз, синтез, порівняння, узагальнення, моделювання.

Зв'язок роботи з науковими програмами, планами, темами. Кваліфікаційна робота пов'язана з напрямом наукових досліджень кафедри інформатики, програмної інженерії та економічної кібернетики факультету комп'ютерних наук, фізики та математики Херсонського державного університету.

Наукова новизна одержаних результатів. Результати кваліфікаційної роботи є новими і полягають у реалізації програмної системи навчального призначення для вивчення навчального модуля «Програмування математичних задач».

Практична цінність роботи полягає у тому, що представлені результати дозволяють застосовувати навчальний модуль «Програмування математичних задач» при вивченні інформатики у курсі середньої школи, розділу «Основи алгоритмізації і програмування», тем, пов'язаних із курсом математики середньої школи та основами комп'ютерної алгебри, розв'язання навчальних задач під час підготовки до олімпіад та конкурсів з інформатики та програмування.

Апробація результатів роботи. Основні положення та результати роботи були обговорені на засіданні кафедри інформатики, програмної інженерії та економічної кібернетики. Також здійснено їх оприлюднення шляхом публікації у фаховому збірнику наукових праць.

Структура роботи. Робота складається зі вступу, двох розділів, висновків та списку використаних джерел.

РОЗДІЛ 1
ТЕОРЕТИЧНІ АСПЕКТИ ПРОГРАМУВАННЯ МАТЕМАТИЧНИХ ЗАДАЧ

1.1 Методи вивчення навчального модуля «Програмування математичних задач»

Шкільний курс інформатики розпочинає знайомство учнів з сучасними інформаційно-комунікаційними технологіями, продовжується воно у закладі вищої освіти та триває впродовж всього життя. Однією з тем, що вивчаються, є програмування [37].

У профільних класах з інформатики відбувається здобуття учнями практичних навичок роботи з основними складовими сучасного комп'ютерного програмного забезпечення, знайомство з основами технологій рішення задач з використанням комп'ютера, слідуючи від їх постановки й побудови відповідних інформаційних моделей, і до завершення – інтерпретації результатів [38].

Під час подальшого здобуття освіти, у коледжах, інститутах, університетах, академіях, в межах освітнього процесу студенти перших курсів математичних та технічних спеціальностей також вивчають програмування і здобувають фундаментальні знання зі створення програм та алгоритмізації процесів.

Навчальний модуль «Програмування математичних задач» поєднує в собі навчання основам алгоритмізації та програмування на основі розв'язування математичних задач. Тобто, щоб навчитися програмувати математичні задачі, необхідно володіти вмінням їх розв'язувати.

Математичною задачею називають будь-яку вимогу виконати обчислення, перетворення, побудову або доведення методами математики [43].

Математична задача може бути реальною, конкретною проблемою (наприклад, у випадку розв’язання рівняння чи нерівності – наявність числових коефіцієнтів), так і мати більш абстрактний характер.

Розв’язанням задачі називають процес перетворення умови на основі знань з певної галузі (до якої відносять задачу), загальних правил логічного виводу.

Основним завданням педагога під час навчання розв’язування математичних задач є навчання учнів знаходженню зв’язків і добору їх послідовності для визначення невідомого [50].

Для того, щоб школярі вміли розв’язувати задачі, вони повинні знати [50]:

- деякі життєві ситуації,
- залежності між величинами,
- розуміння суті арифметичних дій,
- прийоми обчислень,
- загальні правила причинно-наслідкових зв’язків,
- сутність та структуру задачі тощо.

Отже, мета розв’язування задач полягає не лише в одержанні відповіді, але ще й в опануванні процесу, способів її знаходження.

Начальний модуль «Програмування математичних задач» передбачає розв’язання математичних задач засобами програмування.

У спрощеному вигляді загальний процес програмування задачі поділяють на такі етапи:

1. укладання алгоритму розв’язання задачі;
2. написання програми обраною мовою програмування;
3. відладка і тестування програми.

Найчастіше для школярів програмування математичних задач зустрічається на олімпіадах з інформатики та конкурсах з програмування, відповідно, під час підготовки до них діти також вивчають основи алгоритмізації і програмування. Також алгоритмізацію

вивчають студенти на початкових курсах у закладах вищої освіти (інститути, університети, академії), адже розв’язування навчальних задач – це один з найбільш ефективних способів навчитися застосовувати свої знання на практиці. Окремо варто вказати олімпіади та конкурси з програмування для студентів, де завдання також найчастіше орієнтовані саме на програмування математичних задач.

Оршанський С. А. [40] у своїй праці, присвяченій розв’язанню олімпіадних задач з інформатики та програмування, вказує такі кроки (етапи) розв’язання олімпіадної задачі, як :

1. Ознайомлення з умовою – необхідно прочитати, спробувати зрозуміти суть задачі, які дані надаються, а які невідомі.

2. Побудова математичної моделі. Математична модель – де досить точний опис задачі за використання математичного апарату (різного роду функцій, рівнянь, системи рівнянь або нерівностей тощо).

3. Побудова алгоритму для розв’язання. Необхідно скласти алгоритм, який можна записати у вигляді блок схеми, псевдоалгоритмічною мовою чи просто звичайною мовою, зрозумілою учневі. Однак при записі кроків алгоритму спеціальною навчальною мовою алгоритмів, це значно полегшить подальший «переклад» розв’язання на мову програмування.

4. Перевірка (узгодження) алгоритму. Інколи, повторно поглянувши на алгоритм, можна знайти більш ефективне рішення. Вдосконалення може бути навіть незначним. Цей крок дозволить побачити сильні та слабкі сторони, з’ясувати області, над якими ще треба попрацювати, щоб наступного разу, при виникненні подібної ситуації чи задачі, її вирішення було швидшим та ефективнішим.

5. Реалізація мовою програмування. Зазвичай під час конкурсів та олімпіад учасник може самостійно обирати мову програмування для реалізації створеного ним алгоритму. Вибір мов програмування залежить від особливостей змагань та їх сфери, однак, якщо говорити

про конкурси та олімпіади, де завдання орієнтовані на програмування математичних задач, то часто на вибір пропонують такі, як C++ , Java, Pascal, Python.

6. Тестування та відладка. Необхідно перевірити, чи коректно працює програма, чи правильно вона опрацьовує вхідні та вихідні дані. Окремим критеріям оцінювання олімпіадних завдань з програмування часто є час виконання програми, на це також варто зосередити свою увагу.

7. Відправка завдання на перевірку.

Найскладнішим з точки зору навчання у цій ситуації постає етап побудови алгоритму. Саме тому розділ «Основи алгоритмізації та програмування» посідає чільне місце в навчальних програмах, пов’язаних з програмуванням.

У працях, присвячених підготовці фахівців з галузі інформаційних технологій, на початковому етапі навчального процесу, коли студенти тільки знайомляться з програмуванням, особливу увагу приділяють вивченню саме такого розділу, як «Основи алгоритмізації» [49; 54].

Зазвичай перед написанням програми розробляють алгоритм, який можуть записувати з використанням навчальної алгоритмічної мови або відповідної реальної мови програмування.

Під час вивчення та застосування алгоритмізації для навчання основ цього розділу можна послуговуватися:

- природною мовою (словесним описом алгоритму);
- мовою графічних схем;
- шкільною навчальною алгоритмічною мовою;
- мовами програмування.

Навчання основам алгоритмізації за змістовним критерієм умовно можна поділити на такі частини:

- вивчення відомих алгоритмів та їх застосування;
- навчання класичним алгоритмам;

– здобуття навичок побудови опису алгоритму із або без використання відомих алгоритмів.

Для вивчення відомих алгоритмів застосовується метод навчання на практиці: учні мають ознайомитися з алгоритмом, записаним в алгоритмічній чи просто словесній формі, а потім записати його псевдомовою алгоритмів (якщо був поданий словесно) чи певною мовою програмування, виконати алгоритм [54].

Виходячи з цього, на початку навчання зазвичай вивчають структурне програмування. Алгоритми за таких умов використовують розгалуження, цикли, підпрограми. Особливу увагу звертають типам даних. Таким чином, відбувається вивчення двох різних, але взаємопов’язаних областей знань – алгоритміки і особливостей вибраної мови програмування.

Також варто вказати, що на цьому етапі навчання програмуванню перевага віддається мовам програмування навчального призначення, тобто таким, які за свою мету мають не застосування їх на практиці, а саме під час навчального процесу.

При цьому студент не має «завчити», як виглядають базові алгоритми на певній мові програмування. Він повинен розуміти саму суть алгоритму, які дії він виконує для досягнення результату та вміти відтворити його у вигляді алгоритмічних приписів, блок-схем, тощо [47].

Для того, щоб навчитися застосовувати теоретичні алгоритми на практиці, якнайкраще підходить саме програмування математичних задач, які змушують студента використовувати:

– логіку,
– життєвий досвід,
– знання з інших навчальних дисциплін (найчастіше це математика та фізика),
– знання алгоритмів у поєднанні зі знанням мови програмування.

Тобто, якщо студент не розуміє логіку, який саме підхід може

застосовуватися під час розв’язання цієї задачі, він не зможе її реалізувати мовою програмування, і, відповідно, виконати завдання.

Зазначимо, що під час вивчення навчального модуля «Програмування математичних задач» користувач може самостійно обрати мову програмування для реалізації завдання, оскільки головним завданням є не вивчення певної мови програмування, а навчання основам написання алгоритмів. Однак на початковому рівні оволодіння знаннями з алгоритмізації та програмування для спрощення процесу навчання можна вибрати структурну методологію алгоритмізації.

Отже, вивчення алгоритмізації та програмування розвиває системно-логічне мислення учнів, а саме:

- стійкі навички і вміння студентів самостійно у новій ситуації визначати завдання, системно розглядати їх,
- визначати вхідні і вихідні дані,
- висувати власні гіпотези, обґрунтовувати їх,
- пропонувати ефективне розв’язання, перевіряти його на практиці
- вміти пояснити отриманий результат.

Все це, відповідно, формує інформатичні компетентності учнів, дає змогу учневі сформувати алгоритмічний стиль мислення, опанувати сучасні ІКТ.

Як вже було зазначено, навчання основам програмування починається з алгоритмізації розв’язання задач та освоєння основ структурного програмування. На початку учні орієнтовані на розв’язання навчальних задач обчислювального характеру, найпростіших задач логіки, графіки та управління [40].

Для навчання учнів програмуванню математичних задач можна використовувати відомі алгоритми і методи програмування, такі як:

- алгоритми комп’ютерної арифметики (елементарної теорії чисел),

- елементарні алгоритми комп’ютерної алгебри,
- багато алгоритмів з інших предметних областей математики.

Отже, навчальний модуль «Програмування математичних задач» може бути використаний при вивченні:

- основи інформатики у курсі середньої школи,
- розділу «Основи алгоритмізації і програмування»;
- тем, пов’язаних із курсом математики середньої школи та основами комп’ютерної алгебри,
- розв’язання навчальних задач під час підготовки до олімпіади з інформатики та програмування.

Головною метою його використання є більш глибоке розуміння процесів, що відбуваються всередині програми, поліпшення логіки написання власних програм.

1.2 Огляд існуючих програмних засобів для вивчення програмування

Сьогодні в рамках шкільної програми з інформатики учні розвивають логіку, розумові здібності тощо. Навчання учнів основам алгоритмізації та програмування зводиться до формування та розвитку алгоритмічного, логічного і математичного мислення.

У Херсонському державному університеті саме для підтримки процесу вивчення програмування було розроблено низку програмних засобів навчального призначення спрямування.

Інтегроване середовище вивчення курсу «Основи алгоритмізації та програмування» було розроблено для використання у закладах вищої освіти. Його призначенням є застосування в освітньому процесі для лекційно-аудиторної, дистанційної та заочної форм навчання, для студентів, які є здобувачами педагогічної, технічної та економічної освіти [32].

Середовище надає як викладачу, так і для студентів усі можливості для забезпечення ефективності під час вивчення курсу, пов’язаного з основами алгоритмізації та програмування [30].

Користувачами програмного засобу можуть бути учні старших класів закладів загальної середньої освіти, студенти закладів вищої освіти, які вивчають основи алгоритмізації і програмування, вчителі з інформатики, викладачі. Робоча мова програмування у курсі з основ алгоритмізації та програмування – Pascal, як навчальна мова програмування [32].

Програмно-методичний комплекс «Відеоінтерпретатор алгоритмів сортування та пошуку» був створений з метою використання в освітньому процесі під час вивчення основ інформатики, розділу «Основи алгоритмізації і програмування»; навчальних модулів та окремих тем, які пов’язані з алгоритмами обробки масивів, задачами на вибір, пошук та впорядкування інформації. Застосування цього програмно-методичного комплексу сприяє більш глибокому розумінню процесів, що відбуваються всередині комп’ютера, поліпшенню логіки написання власних програм [33].

Програмний засіб можуть використовувати учні загальноосвітніх та спеціалізованих шкіл, котрі вивчають основи алгоритмізації і програмування, вчителі інформатики, а також інші особи, які зацікавлені у вивченні основ алгоритмізації та програмування [33].

Аналогічно до вищезгаданого програмного засобу навчального призначення, робочою мовою програмування для навчання основам програмування було обрано Pascal.

Автори програмного засобу так говорять про його призначення: «Розділ «Основи алгоритмізації та програмування» займає особливе місце в навчальних програмах з дисципліни «Основи інформатики та обчислювальної техніки» загальноосвітньої школи. Якщо метою інших розділів є оволодіння учнями вміннями, знаннями і навичками

користувача сучасних інформаційних технологій, то навчальний матеріал цього розділу орієнтований на формування майбутнього фахівця в галузі розробки комп'ютерних технологій» [33].

Використання в освітньому процесі програмно-методичного комплексу «Відеоінтерпретатор алгоритмів сортування та пошуку» дозволяє:

- на високому рівні організувати навчально-дослідницьку діяльність учнів, що приведе до оволодіння ними знань на високому якісному рівні;
- активізувати самостійність учнів в оволодінні знаннями;
- скоротити час, необхідний для налагодження програми;
- прискорити розвиток логічного мислення учнів.

На сьогодні широке коло користувачів, які зацікавлені у вивчені програмування, найчастіше послуговуються ресурсами, доступними у мережі Інтернет.

Основними з-поміж таких вебдодатків є:

1. навчальні курси для вивчення програмування;
2. платформи, що надають можливість практики алгоритмів як необхідного для вивчення програмування етапу;
3. сайти-архіви навчальних завдань;
4. відеолекції;
5. навчальна та професійна література (книги, підручники, посібники, статті тощо);

Часто пошук певних програмних або алгоритмічних рішень користувачі мережі шукають у статтях, блогах, форумах, спільнотах у соцмережах тощо.

Виконаємо огляд найпопулярніших вебресурсів, які надають можливість вивчення основ (і не тільки) програмування, з акцентом на можливість вивчення основних принципів алгоритмізації, як необхідного етапу для подальшої реалізації і запису мовами

програмування.

1. Codecademy [2].

Ресурс призначений для початкового та середнього рівнів користувачів. З використанням цієї платформи користувачі можуть вивчити цілий стек технологій та мов програмування, від HTML і CSS до Python тощо.

Принцип роботи на Codecademy виглядає можна умовно поділити на дві частини:

1. Користувач вивчає теорію.
2. Виконує завдання в інтерактивному редакторі програмного коду.

Користувач має доступ до окремих курсів, присвячених різноманітним технологіям. Доступ до основної частини кожного з курсів є безкоштовним, проте тести для перевірки, а також завдання з розробки проєктів стануть доступними виключно після оформлення й оплати підписки.

В цілому система орієнтована на вивчення вебпрограмування.

Недоліком системи можна вважати те, що вона доступна лише англійською мовою. Це може дещо обмежити користувачів, однак у той же час буде спонукати їх до вивчення англійської.

2. Coursera [14].

Coursera – це ресурс, який надає безліч курсів з різних технологій та мов програмування, у том числі і від провідних університетів світу.

Базовий фунціонал доступний безкоштовно, за додатковий контент передбачена оплата. Наприклад, перегляд теоретичного матеріалу, поданого як текст і відео, є безкоштовним. Однак переважна більшість тестів та практичних завдань, де передбачена перевірка викладача або інших користувачів, будуть доступними тільки з платною підпискою.

Контент системи може бути доступним різними мовами, залежно

від того, якою був створений курс.

На порталі Coursera користувачі знайдуть велику кількість курсів, присвячених різним мовам програмування, від провідних університетів світу. Тут можна вивчати не тільки веброзробку, а ще й розробку мобільних і десктопних програм. Деякі курси є самодостатніми, однак багато з них об'єднані у «спеціалізації», тобто набори пов'язаних між собою курсів.

Однією з таких спеціалізацій є «Алгоритми» [14]– це набір курсів, присвячених алгоритмізації та застосуванню алгоритмів, до складу якого входять курси від таких університетів, як :

- кафедра комп'ютерних наук Стенфордського університету,
- університету Сан-Дієго в Каліфорнії,
- Прінстонського університету тощо.

Для успішного проходження зазначених курсів необхідним є розуміння англійської мови та базових навичок програмування. Практичні задачі з матеріалів курсу розв'язуються з допомогою бібліотеки алгоритмів, яка надається.

3. Stepik [15].

Ще одна некомерційна платформа для вивчення основ програмування. Характерною її рисою є те, що на ній розміщено порівняно невелика кількість матеріалу, присвяченого конкретним мовам програмування. У той же час, є велика кількість курсів, присвячених застосуванню фундаментальних знань з області математики та теорії алгоритмів, які, безсумнівно, будуть корисні фахівцю у будь-якій сфері розробки програмного забезпечення.

Матеріали платформи доступні англійською та російською мовами.

Наприклад, у курсі, присвяченому вивченню основ алгоритмів на цій платформі [15] детально розібрані базові алгоритмічні методи:

- жадібні алгоритми,

- метод «розділяй і володарюй»,
- динамічне програмування.

Для всіх алгоритмів будуть математично строго доведені коректність і оцінка на час роботи. Алгоритми подані таким чином, щоб одночасно була зрозуміла і їх сутність, і шляхи, як можна здогадатися про їх основні ідеї.

Крім теоретичних основ, розказані особливості реалізації алгоритмів на мовах програмування C ++, Java і Python.

Завдання для закріплення матеріалу передбачають, що потрібно буде запрограмувати більшість із вивчених алгоритмів.

Для того, щоб на практиці перевірити своє розуміння алгоритмів та навички програмування, можна скористатися перевіреним способом – розв’язанням задач. Нижче наведено перелік та особливості найпопулярніших сайтів із завданнями з програмування.

1. TopCoder [1].

TopCoder – це одна з оригінальних онлайн-платформ зі спортивного програмування. Вона надає можливість користувачам платформи безпосередньо конкурувати між собою – кілька разів на місяць за певним графіком проводяться «матчі», метою яких є вирішити завдання якомога швидше і при цьому здобути найвищу кількість балів.

Деякі користувачі (зазвичай з найвищими рейтингами на платформі) ведуть на ресурсі власний «щоденник», де викладають свої враження про змагання з коду, обговорюють алгоритми, математику тощо.

Тут можна отримати доступ до списку алгоритмічних задач із різноманітних минулих змагань. Можна спробувати розв’язати їх самостійно, і при цьому використовувати редактор коду (забезпечується самою платформою).

2. Coderbyte [4].

На Coderbyte міститься більше 200 завдань з програмування. Вирішувати їх можна онлайн, користуючись при цьому одним з 10 мов програмування на вибір. Завдання розподілені по групах, від простих (наприклад, знайти найдовше слово в рядку) до складних.

Також на цьому сайті ви знайдете колекцію посібників з алгоритмів, вступні відео і курси з підготовки до співбесід. На відміну від деяких інших сайтів, тут можна безперешкодно переглядати розв'язки інших користувачів, а не лише офіційні рішення від Coderbyte.

3. Project Euler [11].

Сайт Project Euler пропонує користувачеві велику колекцію завдань з інформатики та математики. Завдання зазвичай пов'язані з написанням програми невеликого обсягу для вирішення певних математичних завдань (наприклад, необхідно написати програму для знаходження суми всіх чисел у послідовності).

На сайті немає редактора коду, відповідно вирішення завдань онлайн на цій же платформі неможливе, тому потрібно буде скористатися іншим онлайн редактором і компілятором (або ж встановленими на власному комп'ютері), і тільки потім вставити своє рішення в форму на сайті.

4. HackerRank [8].

На HackerRank ви знайдете безліч алгоритмів, різнорівневі за складністю завдання з множини областей, пов'язаних з програмуванням та структурами даних, наприклад:

- алгоритми,
- математичні задачі,
- SQL,
- функціональне програмування тощо.

Для виконання завдання можна скористатися онлайн-редактором коду безпосередньо на платформі. Після виконання завдання можна бачити лише свої результати, варіанти рішень інших людей не можна.

До кожного завдання ведеться турнірна таблиця і гілка обговорень (щоб користувачі мали можливість побачити, як інші розв’язали це завдання). Більшість задач супроводжуються поясненнями самого завдання і підходами до її вирішення.

5. CodeChef [3].

CodeChef – це індійський сайт зі спортивного програмування. Тут ви знайдете сотні завдань, відсортованих за рівнем навичок. Код можна писати в онлайн-редакторі.

CodeChef може похвалитися великою спільнотою: програмісти спілкуються на форумах, пишуть мануали і беруть участь в змаганнях.

6. Exercism.io [7].

Сайт Exercism пропонує більше трьох тисяч завдань на 52 мовах програмування. Користувач може обрати мову, навички роботи з якою хоче поліпшити, а потім розв’язувати відповідні завдання прямо на своїй машині. Тобто, Exercism має власний інтерфейс командного рядка, який можна завантажити з GitHub.

Цей сайт має одну істотну відмінність від інших у списку: тут після розв’язання кожної чергової задачі користувач працює з наставником (тьютором). Він переглядає ваші відповіді онлайн і, за необхідності, допомагає із покращенням рішення. Доступ до наступного стеку завдань буде відкрито лише після успішного вирішення попереднього завдання.

7. Codewars

Сайт Codewars пропонує велику колекцію завдань з програмування, представлених членами спільноти. Вирішувати ці завдання можна онлайн в редакторі, вбудованому безпосередньо в сайт.

Користувачеві надається можливість обрати для розв’язання одну з декількох запропонованих мов.

Кожне завдання надає доступ як до обговорення, так і до розв’язків інших користувачів.

8. LeetCode [9].

LeetCode – це популярний сайт зі спортивного програмування. Він надає більш ніж 190 завдань, розв'язувати які можна на 9 різних мовах програмування.

На LeetCode перегляд рішень інших учасників недоступний, однак натомість платформа веде і дозволяє відслідковувати статистику власних рішень користувача. Наприклад, можна побачити, наскільки швидкий та продуктивний ваш код в порівнянні з кодом інших користувачів. З огляду на це, користувач може зробити висновок, чи можна оптимізувати його код.

Завдання на сайті мають 3 основні рівні складності:

- легкий,
- середній,
- важкий.

Крім того, на LeetCode організовуються змагання між учасниками.

Також на сайті є розділ зі статтями для кращого розуміння проблематики окремих завдань та галузей програмування.

9. CodinGame [5].

CodinGame дещо відрізняється від інших сайтів в нашому списку. Тут ви не просто вирішуєте окремі завдання в онлайн-редакторі. Тут ви, власне, берете участь в написанні коду для ігор, в які можна грати на сайті.

До гри додаються:

- опис проблеми,
- критерії тестування,
- редактор, де можна писати власне рішення на 20 мовах програмування.

Незважаючи на відмінності цього сайту від більш стандартних сайтів зі спортивного програмування, він досить популярний серед програмістів, які люблять вирішувати завдання і брати участь в

змаганнях.

11. Geekforgeeks [10].

Досить суперечлива платформа, класифікація вправ на ній не є досконалою. Однак на ній є багато рішень для загальних вправ з алгоритмізації. Кожне рішення включає безліч різних методів, а також складність / пам’ять кожного методу, що є дуже цікавим і корисним для розуміння принципів роботи.

Отже, на сьогодні наявна велика кількість ресурсів, які надають доступ до матеріалів, з допомогою яких можна навчитися програмуванню.

Ресурси відрізняються своїм спрямуванням – деякі орієнтовані лише на ознайомлення з теоретичним матеріалом та завданнями, інші надають можливість написання та виконання програмного коду онлайн на їх платформі у вбудованих редакторах коду, інші надають можливість спілкування з іншими учасниками курсу чи платформи для спільного пошуку чи обговорення тих чи інших алгоритмічних і програмних рішень стосовно певного завдання.

Однак, не зважаючи на значну відмінність між всіма перерахованими ресурсами, можна виділити у них спільну рису: вони популяризують програмування поміж учнів та студентів, зацікавлюють їх, спрощують їм доступ до навчальних ресурсів та дозволяють підлаштовувати процес навчання програмуванню під індивідуальні особливості кожного.

РОЗДІЛ 2
МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОГРАМНОЇ СИСТЕМИ НАВЧАЛЬНОГО ПРИЗНАЧЕННЯ

2.1 Обґрунтування вимог до програмної системи

Учням, які вивчають інформатику на поглибленому рівні, варто оволодіти не лише вмінням написання програмних кодів певною мовою програмування, а й отримати практичні навички алгоритмізації та програмування, використовуючи при цьому сучасні засоби та ІКТ [38].

За таких умов основний аспект учнівською діяльності переноситься на створення алгоритмів і програм. Мета навчання основ алгоритмізації – навчити основних способів організації операцій і даних, а також застосування базових алгоритмічних конструкцій при складанні описів алгоритмів розв’язування різноманітних задач [38].

На сьогодні актуальним є перехід до такої форми навчального процесу, коли безпосередня взаємодія (особиста присутність) учасників начального процесу замінюється формами дистанційної взаємодії. З огляду на це, програмні засоби навчального призначення (або педагогічні програмні засоби), як програми, які призначені для забезпечення навчання, також мають відповідати вимогам сучасності та адаптуватися до них.

Тобто, програмні засоби не мають бути прив’язані до певного конкретного комп’ютера. Доступ до них здійснюється з допомогою мережі Інтернет з будь-якого пристрою, який має таку технічну можливість. Це значно полегшує доступ до навчального матеріалу учасникам навчального процесу.

Сучасний педагогічний програмний засіб – це вебдодаток, що містить певний набір програмних модулів для можливості організації повноцінного навчального процесу, а також забезпечує всіх учасників

навчального процесу можливістю брати участь у створенні та модифікації навчального контенту [51].

Однією з переваг педагогічних програмних засобів є швидкість отримання зворотнього зв'язку між користувачем і програмним середовищем навчання, тобто їх інтерактивність. Окрім того, використання програмних систем навчального призначення дозволяє організувати навчання може здійснюватись у зручному для учнів темпі.

За цільовим призначенням педагогічні програмні засоби є:

- проблемноорієнтованими (вирішують певну навчальну проблему, що має бути вивченою / вирішеною);
- об'єктно-орієнтованими (здійснюють деяку діяльність з об'єктним середовищем, наприклад, системою підготовки текстів, базою даних тощо);
- предметно-орієнтованими ПЗ (здійснюють діяльність у певному предметному середовищі, у найкращому випадку − з інтегрованими елементами технологій навчання).

Педагогічні програмні засоби повинні мати у своєму складі ретельно структурований навчальний матеріал як послідовність текстових та мультимедійних інтерактивних елементів.

Гіпертекстова структура програмного додатку не тільки сприяє формуванню оптимальної траєкторії вивчення матеріалів, а ще й зручності під час роботи: при обранні власного оптимального темпу навчання та способу викладу матеріалу. Це відповідає психофізіологічним особливостям сприйняття учня [17].

Комп'ютерні моделі, конструктори й тренажери дозволяють закріпити знання та сформувати навички їхнього практичного застосування в ситуаціях, що моделюють реальні [44].

Програмна система навчального призначення для навчання програмування математичних задач повинна мати за основу набір завдань. Кожне завдання повинно мати:

- свій індивідуальний номер,
- умову (постановку) задачі,
- вказівки до розв’язання / побудови алгоритму,
- приклад авторського коду для реалізації розв’язання задачі,
- записану у системі відповідь на нього, яку можна певним чином порівняти з відповіддю користувача.

У збірниках задач з алгоритмізації та програмування, на відміну від математичних, фізичних, хімічних тощо, не наводяться відповіді, оскільки результатом розв’язання задачі постає сам алгоритм чи програма. Відповідь на те, чи вірно складений алгоритм, дасть комп’ютер на етапі тестування та виконання програми.

Основою програмної системи навчального призначення є посібник М.С. Львова «Програмування математичних задач. Перші кроки».

У посібнику використовуються алгоритми вирішення конкретних завдань для викладу багатьох відомих алгоритмів і методів програмування. Це алгоритми комп’ютерної арифметики (елементарної теорії чисел), елементарні алгоритми комп’ютерної алгебри, багато алгоритмів з інших предметних областей математики.

Зазвичай, правила оформлення вхідних і вихідних даних олімпіадних завдань з програмування вимагають використання стандартного текстового файлу для зберігання даних. Автор зазначає, що, оскільки цей навчальний посібник не прив’язаний до жодної з олімпіад, формат входу-виходу користувачі можуть вибирати на власний розсуд.

У посібнику автор наводить тільки вказівки до розв’язання кожного завдання. Повні алгоритми або ж їх фрагменти, що демонструють ключові обчислення, включаються до тексту вказівки тільки тоді, коли, на думку автора, це є корисним для розуміння суті. При цьому використовується нотація, близька до мови Паскаль.

Для прикладу наведемо задачу, яка міститься у посібнику.

Задача 3. Квадрати.

Дано прямокутник розміру $A \times B$, де A, B – натуральні числа. Розробіть програму розбиття цього прямокутника на найменшу можливу кількість квадратів. Розміри квадратів можуть бути різними.

Вказівки до розв'язання задачі 3.

Алгоритм розв'язання задачі – це, в принципі, алгоритм Евкліда. Насправді, для того, щоб кількість квадратів під час розбиття була мінімальною, треба відрізати від прямокутника квадрати до тих пір, доки це можливо (рис. 2.1).

Рисунок 2.1 – Ілюстрація алгоритму розв'язання задачі 3

У результаті таких дій ми або отримаємо остачу, в якій орієнтації меншої та більшої сторін поміняються місцями, або процес закінчиться без остачі. У першому випадку необхідно повторити дії з новою парою сторін. У другому випадку – задача розв'язана.

Оскільки сторони вихідного прямокутника – натуральні числа, процес розбиття закінчиться через скінченну кількість кроків.

Якщо A_j, B_j – розміри прямокутника, що залишився на j–му кроці, то умова завершення – довжина меншої сторони рівна нулю: $B_j = 0$.

Зразок авторського коду:

```
Procedure Square(A, B: Integer);
Var N: Integer;
begin       {A > 0, B > 0}
   While (A<>0)and(B<>0) do
```

```
    if A > B
      then begin
         N := A div B; Writeln(N, A); A := A mod B end
      else begin
         N := B div A; Writeln(N, B); B := B mod A end;
end;
```

Отже, процес ознайомлення з наочними матеріалами у цій системі повинен бути побудований наступним чином (рис. 2.2):

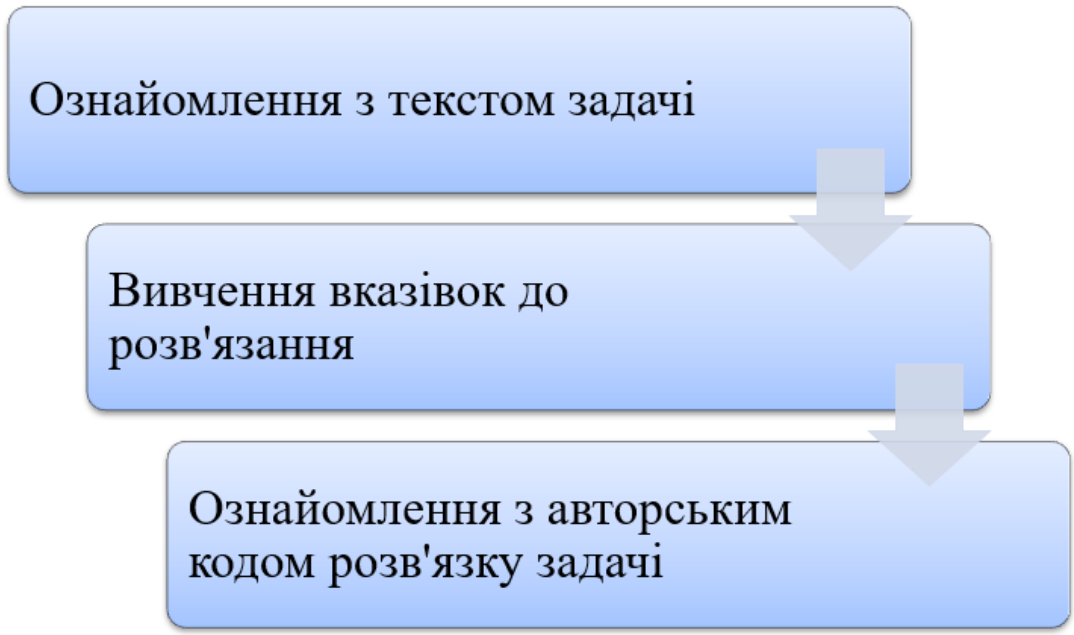

Рисунок 2.2 – Етапи ознайомлення з наочними матеріалами ресурсу

Для програмування математичних задач в якості користувацького інструментарію для написання коду програми для її подальшого виконання рекомендовано використовувати мову програмування навчального призначення, наприклад, Pascal.

Мова програмування Pascal була створена наприкінці 1960-х років з метою навчання студентів програмуванню. І на сьогодні вона теж використовується в багатьох закладах освіти, незважаючи на виключно освітню спрямованість цієї мови програмування, наприклад, на початковому етапі вивчення програмування. Наразі існує низка діалектів

мови та середовищ розробки, які дещо різняться між собою [53].

Мова Pascal має ряд переваг, з-поміж яких:

- досить простий для розуміння синтаксис.
- дуже невеликий обсяг базових понять.
- написані нею програми легко читаються та розуміються людиною.
- низькі системні та апаратні вимоги як для компілятора мови, так і для написаних нею програм.
- мова Pascal – універсальна та може використовуватися для розв’язання великої кількості завдань з програмування.
- Можливість реалізації програмування «згори-донизу», об’єктно-орієнтованого та структурного програмування [12].

Ще однією перевагою Паскаля є те, що, вивчивши з його допомогою основні поняття програмування та принципи, буде набагато легше перейти до більш затребуваних на сьогодні мов програмування практичної направленості.

Таким чином, перехід від напрацьованої бази програмування до більш складних мов програмування буде спрощеним, і вимагатиме від користувача вивчення синтаксису та семантики операторів нової мови [20].

Отже, з огляду на специфіку посібника, коло користувачів програмного додатку буде досить широким:

- учні загальноосвітніх та спеціалізованих шкіл, а також студенти, які вивчають основи алгоритмізації і програмування;
- вчителі інформатики та викладачі курсів з основ програмування;
- інші особи, які вивчають основи алгоритмізації і програмування.

У процесі розробки програмного засобу навчального призначення неможливо уникнути роботи з вимогами до програмного забезпечення. У загальному випадку, сучасні програмні засоби навчального призначення мають задовольняти наступним системним вимогам [51]:

- підтримувати роботу на будь-якій платформі,
- зручний та адаптивний інтерфейс,
- зберігати дані у стандартизованому форматі збереження учбової інформації.

Тобто, система повинна підтримувати наступні процеси, які є необхідними для взаємодії навчальних систем з іншими системами:

1. Зберігати персональні дані.
2. Управляти групами навчання.
3. Управляти реєстрацією.
4. Обробляти і зберігати кінцеві результати (рейтинг користувачів системи, результати виконання завдань тощо).

Більш детально про етапи реалізації програмної системи описано у розділі 2.3.

2.2 Огляд інструментів та технологій розробки

Інструментом реалізації серверної частини вебдодатку «Програмування математичних задач» було обрано мову програмування Python, а саме її фреймворк Django. Для розробки користувацької частини додатку були використані мова програмування JavaScript (а саме – фреймворк Angular), мова розмітки HTML з використанням CSS використовувалися для створення каркасу. Сервером бази даних було обрано PostgreSQL.

Django — високорівневий відкритий Python-фреймворк для вебдодатків, що використовує шаблон проєктування MVC (Model-View-Controller). Однак на відміну від «контролеру» з класичної моделі MVC, Django називає аналогічну частину функціоналу «видом» (view), а «вид» MVC замінюється на «шаблон» (template). Таким чином, розробниками Django шаблон MVC називається як MTV (Model-Template-View) [6].

У загальному випадку високорівнева архітектура додатку,

реалізованого на Django, матиме вигляд, як показано на рис. 2.3.

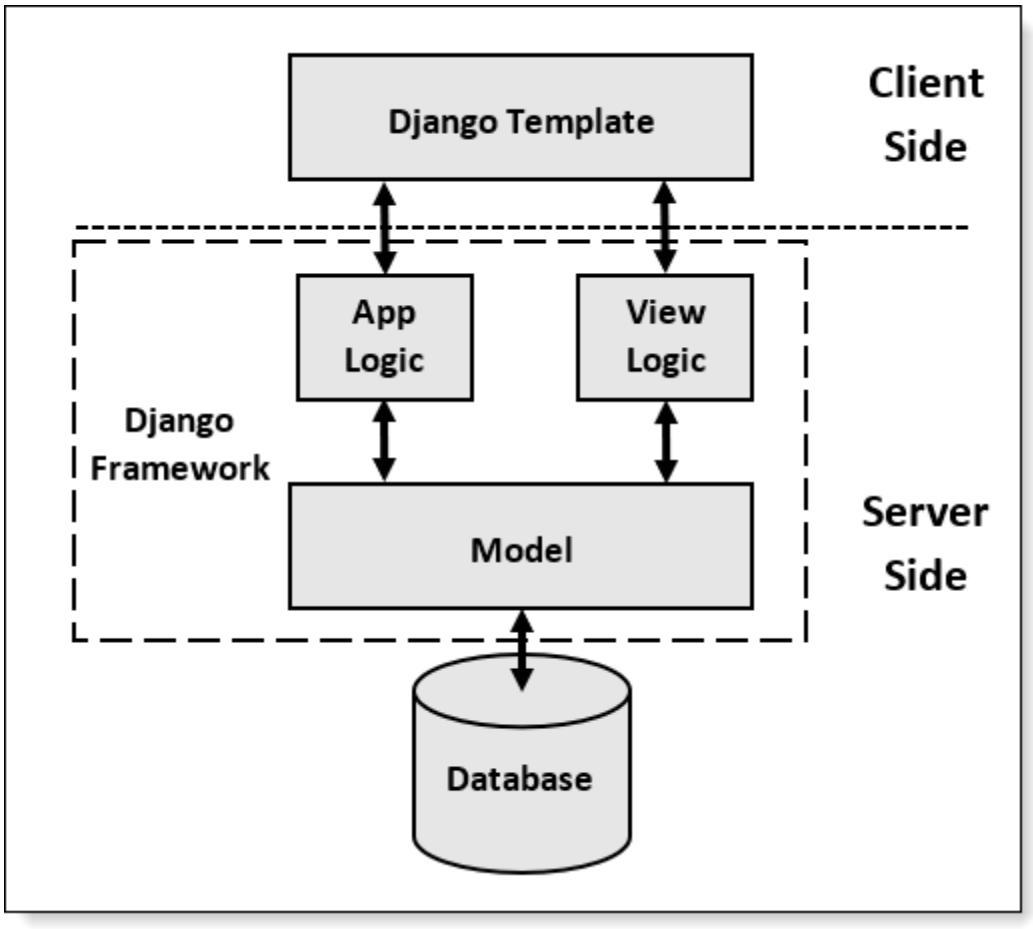

Рисунок 2.3 – Архітектура системи [6]

Django підтримується парадигма об’єктно-орієнтованого програмування. Об’єкти бази даних в рамках термінології Django мають назву «моделі». Також фреймворком надається у розпорядження розвинутий прикладний програмний інтерфейс для доступу до даних на високому рівні.

Для роботи з базою даних Django використовує власний ORM, в якому модель даних описується з допомогою класів Python, и вже на її основі генерується схема бази даних. ORM (англ. Object-relational mapping, Об’єктно-реляційна проекція) — технологія програмування, яка зв’язує бази даних з концепціями об’єктно-орієнтованих мов програмування, створюючи «віртуальну об’єктну базу даних».

В Django є окрема мова для опису шаблонів, що виконує функцію відображення даних. Змінити дані в БД її операторами неможливо. В ній присутні оператори циклу, умови, форматування даних.

У складі Django присутній власний вебсервер для розробки і налагоджування.

Однією із сильних сторін Django є автоматичний інтерфейс адміністратора. Він використовує метадані моделі для надання багатофункціонального та вже відразу готового до використання інтерфейсу для роботи із вмістом сайту.

PostgreSQL – уніфікований сервер баз даних, що має єдиний двигун – storage engine. PostgreSQL використовує клієнт-серверну модель.

Для кожного клієнта на сервері створюється новий процес (не потік). Для роботи з такими клієнтськими процесами сервер використовує семафори.

Клієнтський запит проходить наступні стадії.

1. З’єднання.
2. Парсинг: перевіряється коректність запиту і створюється дерево запиту (query tree). В основу парсера покладені базові UNIX утиліти yacc і lex.
3. Rewrite: береться дерево запитів і перевіряється наявність в ньому правил (rules), які лежать в системних каталогах. Кожного разу користувацький запит переписується на запит, який отримує доступ до таблиць бази даних.
4. Оптимізатор: на кожен запит створюється план запиту - query plan, який передається виконавцю – executor. Сенс плану в тому, що в ньому перебираються всі можливі варіанти отримання результату, і вибирається найшвидший варіант.
5. Виконання запиту: виконавець рекурсивно проходить по дереву і отримує результат (при цьому використовується сортування,

тощо), і повертає рядки. PostgreSQL є об’єктно-реляційною базою даних, кожна таблиця в ній представляє клас, між таблицями реалізовано наслідування.

Найважливіші технічні характеристики PostgreSQL:

- *Надійність і стійкість*. Надійність PostgreSQL є відомим фактом, доведеним на прикладі багатьох проєктів, в яких PostgreSQL працює без збоїв під високими навантаженнями протягом декількох років.
- Чудова підтримка. Спільнота PostgreSQL надає кваліфіковану та швидку допомогу. Комерційні компанії пропонують свої послуги по всьому світу.
- Конкурентна робота при великому навантаженні. PostgreSQL використовує багатоверсійність (MVCC) для забезпечення надійної і швидкої роботи в конкурентних умовах під високим навантаженням.
- Масштабованість. PostgreSQL відмінно використовує сучасну архітектуру багатоядерних процесорів - його продуктивність зростає лінійно аж до 64-х ядер. Кластерні рішення на основі Postgres-XC, Postgres-XL допомагають з горизонтальною масштабністю.
- Кросплатформенність. PostgreSQL працює під усіма видами UNIX-подібних систем, включаючи Linux, FreeBSD, Solaris, HP / UX, Mac OS X, а також під Windows.
- Можливість розширення. Доступні вихідні коди PostgreSQL, що робить можливим додавання нових функцій для вашого проєкту без додаткових проблем. Можливість розширення PostgreSQL дозволяє створювати нові типи даних і методи доступу. Доступність. PostgreSQL поширюється під ліцензією, що не накладає ніяких обмежень на комерційне використання і не вимагає ліцензійних виплат.

У системі передбачена можливість отримання даних про користувачів, виконання ними завдань, а також про бібліотеку завдань в цілому у форматі XML.

Тип XML призначений для зберігання XML-даних. Його перевага в порівнянні зі звичайним типом text полягає в тому, що він перевіряє значення, які вводяться, на допустимість за правилами XML. Також роботу з цією технологією доповнюють безпечні для типів даних функції.

Для отримання XML-контенту з даних SQL існує цілий набір функцій і функціональних виразів, особливо корисних для видачі клієнтським додаткам результатів запиту у вигляді XML-документів.

Веб-сайти будуються з допомогою мови розмітки HTML та каскадних таблиць стилів CSS. Мова HTML описує структуру та вміст вебсторінки, а з допомогою мови CSS можна надати сторінці потрібного вигляду. HTML/CSS запускається на стороні комп'ютера-користувача.

JavaScript (JS) — динамічна, об'єктно-орієнтована прототипна мова програмування. Частіше за все її використовують, щоб створити сценарії вебсторінок, для отримання можливості на стороні клієнта (тобто на пристрої кінцевого користувача) взаємодії з користувачем, керування браузером, асинхронного обміну даних із сервером, зміни структури та зовнішнього вигляду вебсторінки.

Тобто, за допомогою JavaScript створюють веб-сайти, які без перевантаження і навігації постійно надають користувачу оновлену інформацію.

Отже, розробка програмної системи навчального призначення проводилася з використанням стандартного набору технологій для розробки вебдодатків.

2.3 Проєктування та реалізація програмної системи навчального призначення

У процесі розробки програмного забезпечення значна увага приділяється роботі з вимогами. Здебільшого це здійснюється на етапах аналізу вимог та подальшої специфікації програмного забезпечення.

Вимогами до програмного забезпечення називається набір вимог щодо властивостей, якості та функцій програмного забезпечення, яке буде розроблено. Вимоги визначаються в процесі аналізу вимог та фіксуються в специфікації вимог, діаграмах прецедентів та інших артефактах процесу аналізу та розробки вимог.

Зазвичай у процесі розробки вимог до системи може виділяють декілька етапів:

- Збір вимог.
- Аналіз вимог (перевірка цілісності та закінченості).
- Специфікація (документування вимог).
- Тестування вимог.

Визначають три рівні вимог до програмного забезпечення [21].

1. *Бізнес-вимоги* визначають призначення програмного засобу.
2. *Користувацькі вимоги* визначають набір завдань користувача, які повинна вирішувати програма, а також сценарії їхнього вирішення в системі. Можуть мати вигляд тверджень, варіантів використання, історій користувача, сценаріїв взаємодії.
3. *Функціональні вимоги* визначають, що саме повинен робити програмний продукт. Зазвичай ці вимоги описуються у специфікації програмного забезпечення.

Специфікація вимог до програмного забезпечення (Software Requirements Specification, SRS) – це повний опис поведінки розроблюваної системи. Складається з набору функціональних та нефункціональних вимог.

Методами знаходження вимог зазвичай є спілкування з майбутнім користувачем (інтерв’ю або анкетування), аналіз нормативної документації та законодавства, а також аналіз бізнес-процесів.

Проєктування та реалізація математичних систем навчального

призначення здійснюється:

- на основі функціональних вимог;
- з дотриманням державних стандартів якості;
- з урахуванням специфіки розробки програмних систем навчального призначення;
- за технологіями розроблення та підтримки протягом життєвого циклу середніх та великих програмних систем.

Окремими завданнями реалізації технологій створення програмних математичних систем є завдання реалізації редакторів.

Як було зазначено у попередніх розділах, наповнення системи має за основу авторський посібник та навчальний курс М.С. Львова «Програмування математичних задач. Перші кроки».

Задля збереження авторського структурування матеріалу, всі завдання, рішення та методичні матеріали у додатку змістовно поділяються на такі розділи:

1. Задачі та алгоритми комп’ютерної арифметики.
2. Задачі та алгоритми комп’ютерної алгебри.
3. Різні задачі.
4. Задачі для самостійного розв’язання.
5. Тренувальні задачі.

На основі аналізу вимог до систем навчального призначення з урахуванням специфіки навчального модуля «Програмування математичних задач» сформульовано короткий опис майбутнього вебдодатку.

Система має бути простою і зрозумілою, мати інтуїтивно зрозумілий інтерфейс та функціонал. Майбутній вебдодаток має містити в собі певний структурований набір завдань. Завдання мають бути спрямовані на практичну реалізацію математичних алгоритмів засобами

програмування, набуті навички будуть використані студентами у подальшій професійній діяльності. З огляду на вищезазначене, система має надавати можливість ознайомитися з короткими методичними вказівками щодо кроків алгоритму, поданим у доступній для сприйняття формі.

Після завершення розв’язання будь-якої задачі користувач повинен буде отримати інформацію про свій результат.

Реєстрація користувача в системі надаватиме йому можливість бачити особисту історію виконання задач, тобто відображення динаміки результатів.

На основі аналізу методів вивчення програмуванню в загальному випадку, та, зокрема, методів навчання програмуванню математичних задач, нами було сформульовано основні функціональні та нефункціональні вимоги до програмної системи навчального призначення, на основі яких і буде проводитися проєктування веб-сайту.

Функціональні вимоги:

- Як користувач вебресурсу, я хочу мати можливість зареєструватися в системі.
- Як користувач веб-сайту, я хочу мати можливість переглядати основні сторінки сайту.
- Як зареєстрований користувач веб-сайту, я хочу бачити власну історію розв’язання завдань.
- Як вчитель, я хочу мати можливість додавати та змінювати наповнення сторінок сайту.
- Як адміністратор веб-сайту, я повинен мати можливість керувати даними для легкого редагування, оновлення та додавання інформації.
- Як адміністратор веб-сайту, я повинен мати можливість видаляти користувачів.

Нефункціональні вимоги:

1. Інформація всіх завдань та користувачів повинна зберігатися в базі даних, доступній вебдодатку.
2. Сервер PostgreSQL буде використовуватися як SQL-рушій і база даних.
3. Вебдодаток працює 24 години на добу.
4. Користувач може отримати доступ з будь-якого комп'ютера, який має можливості перегляду Інтернету та підключення до Інтернету.
5. Вебдодаток повинен обробляти очікувані та непередбачувані помилки таким чином, щоб запобігти втраті інформації.
6. Звичайні користувачі можуть просто читати інформацію, але вони не можуть нічого редагувати або змінювати.

Згідно з вимогами до системи, нами було виокремлено основні структурні частини сайту (рис. 2.4).

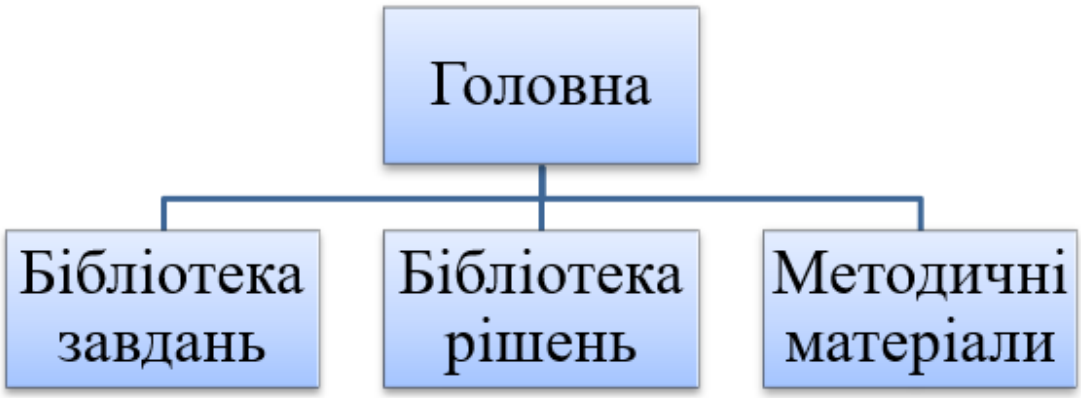

Рисунок 2.4 – Основні структурні частини сайту

Окремо слід виділити:

- персональну сторінку користувача,
- форму входу / реєстрації.

Високорівнева архітектура системи відповідає стандартній архітектурі додатків, розроблених на основі Django. Взаємодія між клієнтом і сервером реалізовується стандартно, із використанням протоколу HTTP, тобто вебдодаток працює за стандартним принципом.

Користувач за допомогою браузера відправляє HTTP-запит за певною URL-адресою, яка зіставляється з ресурсами на web-сервері (Django). Якщо вказана URL-адреса вказує на динамічний ресурс, то сервер запускає деяку зовнішню програму (web-додаток), яка формує HTML-сторінку, що може бути відображена браузером, і повертає її клієнту. Основною частиною web-додатку є його логіка на стороні web-сервера.

На основі аналізу та узагальнення наведених вимог, нами було виокремлено наступні ролі користувачів:

- Адміністратор системи,
- вчитель,
- зареєстрований користувач,
- гість.

Адміністратор системи може:

- входити у систему зі своїми правами,
- переглядати основні сторінки сайту,
- видаляти користувачів,
- додавати/видаляти сторінки,
- змінювати структуру та наповнення сторінок сайту.

Вчитель може:

- зареєструватися в системі,
- здійснити вхід /вихід з системи,
- переглядати основні сторінки сайту,
- змінювати структуру та наповнення сторінок сайту.

Учень може:

- зареєструватися в системі,
- здійснити вхід /вихід з системи,
- переглядати основні сторінки сайту,

– отримати доступ до власної історії виконаних завдань.

Гість зможе переглядати основні сторінки системи.

Функціональні можливості кожної ролі представлені за допомогою діаграми активності (прецедентів) (рис. 2.5).

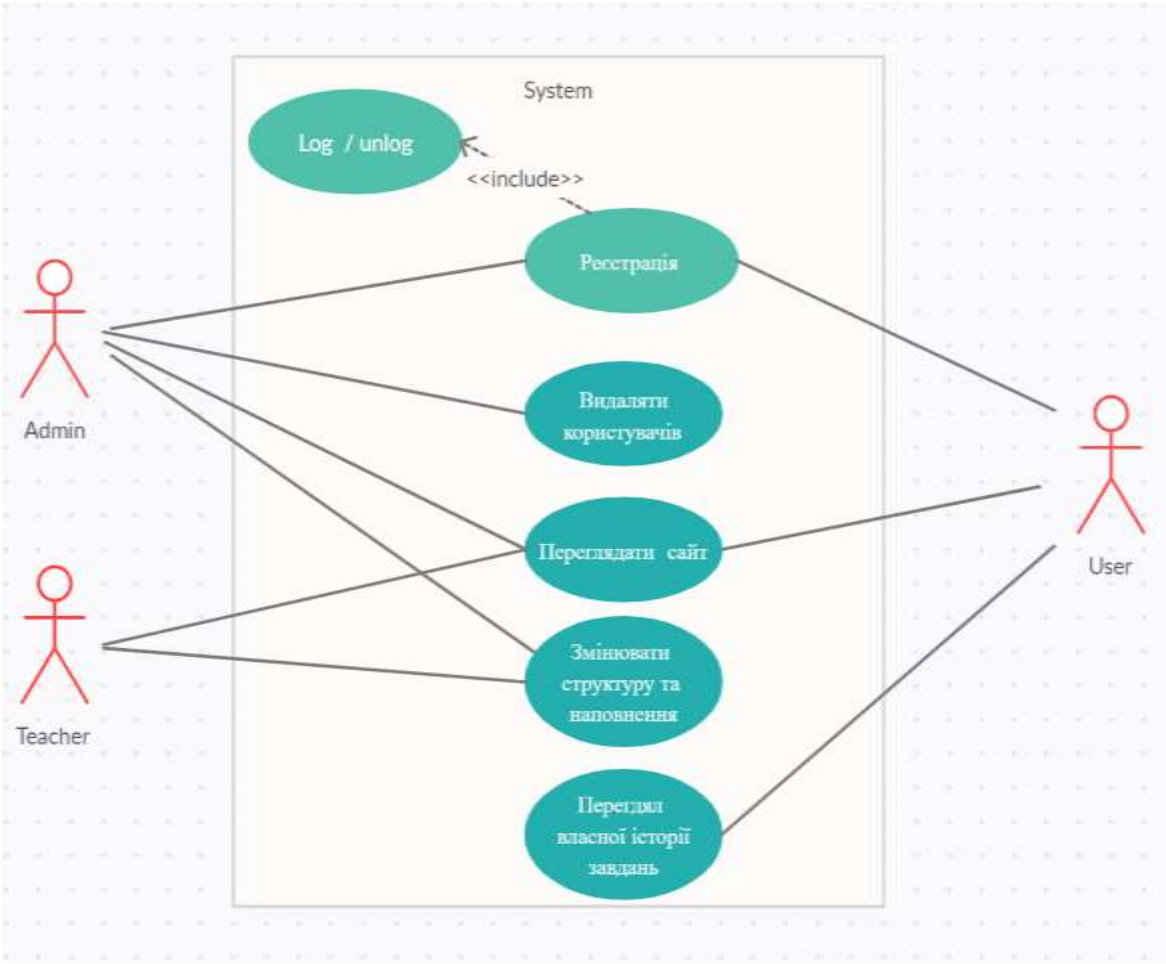

Рисунок 2.5 – Діаграма активності користувачів

Зауважимо, що гостьовий доступ буде сприяти популяризації вебдодатку, оскільки більшість користувачів бажають спочатку «спробувати» функціонал, дізнатися, наскільки він їм сподобається, перш ніж зареєструватися в системі.

Зауважимо, що можливість доступу до основних сторінок додатку жодною мірою не буде зменшена для гостей ресурсу порівняно із

зареєстрованими користувачами.

Проаналізувавши вимоги до системи, виокремлені користувацькі ролі та основні структурні елементи системи, було спроєктовано базу даних наступної структури (рис. 2.6).

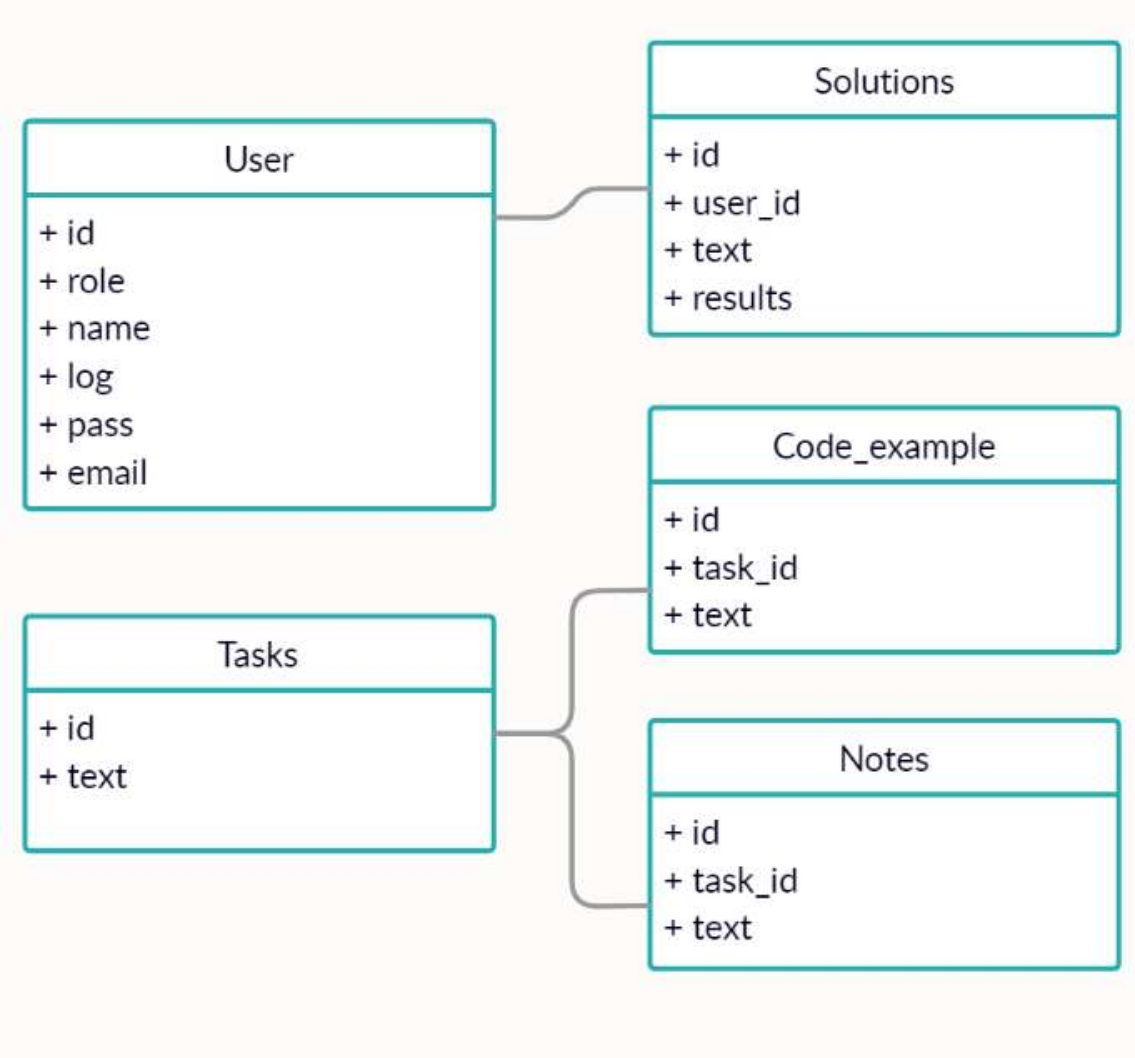

Рисунок 2.6 – Структура бази даних системи

Система надає доступ до завдань навчального курсу з програмування математичних задач. Для їх розв’язання будуть потрібні знання алгоритмів з елементарної теорії чисел та з основ комп’ютерної алгебри.

Спочатку учень має можливість ознайомитись із постановкою задачі, тобто її текстовим представленням. Далі він має вибір: перейти

відразу до розв’язання задачі, тобто до написання програмного коду певною мовою, чи спочатку ознайомитись із вказівками до розв’язку.

Вказівки складаються із таких структурних частин:

- теоретичний матеріал із предметної області задачі,
- методи і формули,
- пояснення ходу мислення,
- приклад фрагменту коду / алгоритму.

Зауважимо, що, на відміну від вебдодатку, в авторській версії посібника програмний код додавався до вказівок лише в тому випадку, якщо автор вважав це доцільним для кращого сприйняття сутності завдання та алгоритму розв’язання задачі в цілому.

Після виконання завдання (тобто після написання алгоритму та реалізації його обраною мовою програмування) користувач має можливість звірити свою отриману відповідь з правильною, вказаною на сайті. Якщо задача передбачає відповідь у формі числа, то можна буде переглянути її окремо. У всіх інших випадках користувачу буде доступний перегляд авторського програмного коду для розв’язання задачі, щоб він міг порівняти алгоритми та проаналізувати написану програму.

На особистій сторінці (у персональному кабінеті) кожного користувача розміщена статистична інформація про попередні виконання завдань.

Інтерфейс системи представлений трьома мовами: українською, англійською та російською. Наявність декількох мов дозволяє користувачам обирати для роботи з системою ту, яка найзручніша.

Зазначимо, що для реалізації цієї можливості було виконано переклад всіх задач та вказівок до них, а також коментарів в зразках авторського коду. Такий мультимовний інтерфейс сприяє популяризації вебдодатку навчального призначення і, відповідно, навчального модуля «Програмування математичних задач».

Отже, створений вебдодаток можна використовувати для навчання програмуванню як під час навчальних занять, так і під час самостійного вивчення.

Таким чином, навчання учнів основам алгоритмізації та програмування зводиться до формування та розвитку алгоритмічного, логічного і математичного мислення. з використанням нашої системи дозволяє реалізувати такі завдання:

1. Зробити процес навчання інформатики більш наочним та динамічним, а , отже, ефективнішим:
2. Покращити підготовку учнів у галузі алгоритмізації та програмування;
3. Розвинути системне мислення учнів, їх творчі та дослідницькі здібності
4. Підвищити професійну зорієнтованість учнів у природничому та технічному напрямках, розвинути професійні компетентності та практичні вміння застосування ІКТ як на уроках, так і у позаурочній діяльності;
5. Здійснити інтеграцію освітніх предметів (математики, інформатики, можливо, фізики) через виконання проєктних робіт.

У процесі індивідуальної роботи забезпечується самостійне засвоєння знань, формується пізнавальність, самостійність та адекватність самооцінки.

До курсу «Програмування математичних задач» включені доцільно дібрані навчальні задачі практичного змісту із предметної області математики та основ комп’ютерної алгебри.

Виконання користувачами додатку задач з алгоритмізації та програмування допоможе у розвитку логічного мислення та сприятиме формуванню базових практичних професійних навичок, що неодмінно матиме позитивний вплив на подальшу фахову діяльність студента.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було досягнуто поставленої мети, яка полягала у проєктуванні та розробці програмної системи навчального призначення з навчального модуля «Програмування математичних задач».

У ході дослідження були виконані наступні завдання:

1. На основі виконаного теоретичного аналізу навчально-методичної літератури, пов’язаної з програмуванням математичних задач, було виділено методи вивчення навчального модуля «Програмування математичних задач». Навчання програмуванню починається з алгоритмізації розв’язання задач та освоєння основ структурного програмування. Для навчання учнів програмуванню математичних задач можна використовувати відомі алгоритми і методи програмування, такі як алгоритми комп’ютерної арифметики (елементарної теорії чисел), елементарні алгоритми комп’ютерної алгебри, багато алгоритмів з інших предметних областей математики.

2. Проаналізовано існуючі програмні продукти навчального призначення з програмування, наведено їх ключові характеристики, які мають значення при вивченні основ алгоритмізації і програмування. На сьогодні наявна велика кількість ресурсів, які надають доступ до матеріалів, з допомогою яких можна навчитися програмуванню.

Ресурси відрізняються своєю спрямованістю – деякі орієнтовані лише на ознайомлення з теоретичним матеріалом та завданнями, інші надають можливість написання та виконання програмного коду онлайн на їх платформі у вбудованих редакторах коду, інші надають можливість спілкування з іншими учасниками курсу чи платформи для спільного пошуку чи обговорення тих чи інших алгоритмічних і програмних рішень стосовно певного завдання. Однак вони популяризують

програмування поміж учнів та студентів, зацікавлюють їх, спрощують їм доступ до навчальних ресурсів та дозволяють підлаштовувати процес навчання програмуванню під індивідуальні особливості кожного.

3. Визначено вимоги до програмної системи навчального призначення «Програмування математичних задач» як до сучасного педагогічного програмного засобу, тобто вебдодатку, що містить певний набір програмних модулів для повноцінної організації навчального процесу. Найголовніше, система має бути простою і зрозумілою, мати інтуїтивно зрозумілий інтерфейс та функціонал. Майбутній вебдодаток має містити в собі певний структурований набір завдань. Завдання мають бути спрямовані на практичну реалізацію математичних алгоритмів засобами програмування, набуті навички будуть використані студентами у подальшій професійній діяльності.

4. Описано інструменти та технології розробки програмного засобу та обґрунтовано їх вибір. Розробка програмної системи навчального призначення проводилася з використанням стандартного набору технологій для розробки вебдодатків.

5. Спроєктовано та реалізовано програмну систему навчального призначення для вивчення предметного модуля «Програмування математичних задач». Розроблена система може бути використана при вивченні основ інформатики у курсі середньої школи, розділу «Основи алгоритмізації і програмування», тем, пов’язаних із курсом математики середньої школи та основами комп’ютерної алгебри, розв’язання навчальних задач під час підготовки до олімпіад та конкурсів з інформатики та програмування.

Головною метою його використання є більш глибоке розуміння процесів, що відбуваються всередині програми, поліпшення логіки написання власних програм.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Challenges. URL:https://www.topcoder.com/challenges/
2. Codecademy. URL: https://www.codecademy.com
3. CodeChef. URL: https://www.codechef.com
4. Coderbyte: Code Screening, Challenges, & Interview Prep. URL: https://coderbyte.com
5. CodinGame. URL: https://www.codingame.com.
6. Django's Structure. URL: https://djangobook.com/mdj2-django-structure/
7. Exercism. Code practice and mentorship for everyone. URL: https://exercism.io/
8. HackerRank. URL: https://www.hackerrank.com
9. Leetcode: A New Way to Learn. URL: https://leetcode.com/
10. Practice. URL: https://practice.geeksforgeeks.org/
11. Project Euler. URL: https://projecteuler.net
12. Richter T.V. Theory and practice of the solution of mathematical problems of the programming language Turbo Pascal in the conditions of distance learning. Solikamsk, 2013. 112 p.
13. Акулич И. Л. Математическое программирование в примерах и задачах. *Высшая школа*, 1993. С. 336-336.
14. Алгоритмы. URL: https://www.coursera.org/browse/computer-science/algorithms.
15. Алгоритмы: теория и практика. Методы. URL: https://stepik.org/course/217
16. Балл Г.А. Теория учебных задач. Психолого-педагогический аспект. Москва: Педагогика, 1990. 184 с.
17. Биков В. Ю., Лапінський В. В. Методологічні та методичні основи створення і використання електронних засобів навчального призначення. *Комп'ютер у школі та сім'ї*. 2012. Т. 3. С. 3-6.

18. Богач І.В., Довгалець С.М., Дубовой В.М. Алгоритми розв’язання задач з програмування. Вінниця: ВНТУ, 2017. 119 с.
19. Вакалюк Т. А. Математичні основи розв’язування олімпіадних задач з інформатики на сайті e-olimp. *Інформаційні технології в освіті*. 2010. №. 7. С. 139-144.
20. Вакалюк Т. А., Жуковський С. С. Структурне програмування мовою Pascal (лабораторний практикум). Навчальний посібник для студентів фізико-математичного факультету. Житомир: Вид-во ЖДУ, 2011, С. 1-15.
21. Вигерс К. Разработка требований к программному обеспечению. М.: Издательско-торговый дом «Русская Редакция», 2004. 576 с.
22. Грешилов А. А. Математические методы принятия решений. Издательство МГТУ им. Баумана, 2014. 647 с.
23. Дикин И.И., Зоркальцев В.И. Итеративное решение задач математического программирования. Новосибирск, 1990. 144 с.
24. Загуменов Ю., Шелкович Л., Шварц Г. Проектно-проблемний підхід до формування творчого мислення. 2005. URL: http://osvita.ua/school/method/technol/1414/.
25. Капітонова Ю.В., Кривий С.Л., Летичевський О.А., Луцький Г.М., Печурін М.К. Основи дискретної математики. К., 2002.
26. Караванова Т.П. Методика розв’язування алгоритмічних задач. Основи алгоритмізації та програмування. Кам’янець-Подільський: Аксіома, 2013. 460 с.
27. Караванова Т.П. Основи алгоритмізації та програмування: 777 задач з рекомендаціями та прикладами. Київ: Генеза, 2006. 288 с..
28. Карманов В.Г. Математическое программирование: Учеб. пособие. Москва: Физматлит, 2011. 264 с.
29. Кушнір В. А., Кушнір Г. А. Комп’ютерне моделювання у розв’язуванні задач з параметрами. *Комп’ютер у школі та сім’ї*. 2010. №. 5. С. 29-34.

30. Львов М.С. Интеллектуальные свойства систем компьютерной математики учебного назначения и методы их реализации. *Искуственный интеллект*. 2011. № 2. С. 45-52.

31. Львов М.С. Математические модели предметных областей в системах компьютерной математики учебного назначения. *Вестник Харьковского национального университета (Серия «Математическое моделирование. Информационные технологии. Автоматизированные системы управления»)*. 2011. № 987. С. 46-60

32. Львов М.С. Основные принципы построения педагогических программных средств поддержки практических занятий. *Управляющие системы и машины*. 2006. № 6. С. 70 - 75.

33. Львов М.С., Співаковський О.В. ПМК «Відеоінтерпретатор алгоритмів пошуку та сортування». *Інформатизація освіти України: стан, проблеми, перспективи*. 2003. С. 100 - 102.

34. Метельский Н.В. Дидактика математики: Общая методика и ее проблемы: Учебн. пособие для вузов. Минск: Изд-во БГУ, 1982. 256с.

35. Минкіна Т. В., Шендриков Н. В. Застосування сучасних інформаційних технологій для розв'язання задач математичного програмування. *Культура і суспільство: історія та сучасність*. 2014. С. 83-87.

36. Михалевич В. М., Тютюнник О. І. Проектування навчальних задач з лінійного програмування з використанням систем комп'ютерної математики. *Інформаційні технології і засоби навчання*. 2013. Т. 38, Вип. 6. С. 123-137.

37. Морзе Н. В. Концепція навчання учнів інформатики у 5-9-х класах загальноосвітніх навчальних закладів. *Інформатика та інформаційні технології в навчальних закладах.* 2012. № 3. С. 8-23

38. Навчальна програма з інформатики (профільний рівень) для 10-11 класів загальноосвітніх шкіл. 2017. 14 с. URL: https://mon.gov.ua/storage/app/media/zagalna%20serednya/programy-10-

11-klas/2018-2019/01/10-11-profilniy-riven.docx.

39. Нікітченко М. С. Математична логіка та теорія алгоритмів. Київ: ВПЦ «Київський університет», 2008. 207 с.
40. Оршанский С. А. О решении олимпиадных задач по программированию. *Мир ПК.* 2005. №9. С. 1-27.
41. Присяжнюк Т. А. Оптимізація розв'язання задач з програмування засобами математики. *Комп'ютер у школі та сім'ї.* 2010. №. 3 (83). С. 16-17.
42. Про освіту: Закон України від 05.09.2017 № 2145-VIII про освіту. Голос України. 2017. 27 вересня. С. 178-179.
43. Слєпкань З. І. Методика навчання математики: підручник. Київ: Вища школа, 2006. 582 с.
44. Словінська Ю. А. Огляд педагогічних програмних засобів, що використовуються в навчальному процесі. *Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку*. 2013. С. 212-214.
45. Співаковський О.В., Львов М.С., Кравцов Г.М. Педагогічні технології та педагогічно орієнтовані програмні системи: предметно-орієнтований підхід. *Комп'ютер у школі й сім'ї.* 2002. № 4 (22). С. 24-28.
46. Сулейманов Р. Р. Компьютерное моделирование математических задач. Учебное пособие. Издательство «Лаборатория знаний». 2014. 155 с.
47. Харченко В. М. Щодо форм самостійної підготовки учнів до участі в олімпіадах з математики та інформатики. *Редакційна колегія*. 2014. С. 120.
48. Чадин М. С., Бакаева О. А. Алгоритм как основа решения олимпиадных задач по программированию. *Современные инновации в науке и технике.* Курск: Юго-Западный государственный университет, 2019. С. 416-418.

49. Черкаська Л. П. Алгоритми в навчанні математики: перспективи використання. *Сучасні інформаційні технології та інноваційні методики навчання у підготовці фахівців: методологія, теорія, досвід, проблеми*. 2013. № 34. С. 134-139.

50. Шаповал І., Королюк О. М. Текстові задачі на сумісну роботу і планування в шкільному курсі математики. Науковий пошук молодих дослідників: збірник наукових праць студентів, магістрантів та викладачів. 2014. №. 7. С. 59-62.

51. Шевчук С.В. Критерії якості та загальні вимоги до педагогічних програмних засобів. URL: http://enpuir.npu.edu.ua/bitstream/123456789/5842/1/Shevchuk.pdf.

52. Шишкіна М. П. Класифікація програмних засобів навчального призначення. Наукові записки. 2009. Т. 2. №. 82. С. 286-292.

53. Язык Pascal (Паскаль). Программирование для начинающих. URL: https://pas1.ru.

54. Ящик О.Б. Методика навчання алгоритмізації та програмування старшокласників на рівні поглибленого вивчення інформатики: автореф.дис…канд.пед.наук:13.00.02. Київ, 2016. 23 с.

ДОДАТКИ

Додаток А