

ПРОЄКТ

Міністерство освіти і науки України

Модельна навчальна програма
навчального предмета
«Інформатика. 10–12 клас. Поглиблений рівень»
для закладів загальної середньої освіти

Автори: Микола Арбузов,
Анастасія Забазнова

*Програма створена в рамках проєкту «Зміст»
благодійного фонду savED
за підтримки фінської міжнародної неурядової організації Finn Church Aid*

Вступна частина

Модельна навчальна програма навчального предмета «Інформатика. 10–12 клас. Поглиблений рівень» (далі – Програма) створена з метою реалізації компетентнісного потенціалу, мети й завдань інформатичної освітньої галузі та призначена для викладання навчального предмета «Інформатика» в 10–12 класах STEM-кластеру з поглибленим вивченням математики, інформатики й технологій у закладах загальної середньої освіти, які забезпечують здобуття профільної середньої освіти за академічним спрямуванням.

Програма створена в рамках проєкту «Зміст» благодійного фонду savED за підтримки фінської міжнародної неурядової організації Finn Church Aid.

Програма *враховує наступність* та логічно продовжує реалізацію завдань інформатичної освітньої галузі, які впроваджувалися через навчальний предмет «Інформатика. 5–6 класи» (будь-яких авторських колективів) та навчальний предмет «Інформатика. 7–9 клас» у циклі базового предметного навчання базової середньої освіти. Програма реалізовує усі обов'язкові результати навчання (орієнтири для оцінювання) поглибленого рівня інформатичної освітньої галузі, визначені Державним стандартом профільної середньої освіти.

Метою викладання навчального предмета «Інформатика» (далі – Навчальний предмет) є формування інформаційно-цифрової компетентності у взаємозв'язку з іншими ключовими компетентностями для успішної освітньої та подальшої професійної діяльності впродовж життя, що передбачає засвоєння системи знань, удосконалення вміння розв'язувати алгоритмічні та практичні задачі.

Такими ключовими компетентностями є: вільне володіння державною мовою (через усвідомлене використання ІТ-термінів, пояснення власних рішень, спілкування у командах під час роботи над проєктами); здатність спілкування рідною (у разі відмінності від державної) та іноземними мовами (використання іноземних ресурсів, інтерфейсів, технічних документацій, комунікації у глобальній ІТ-спільноті); математична компетентність (використання алгоритмів, логічних структур, математичних моделей, виконання математичних розрахунків); компетентності в галузі природничих наук, техніки й технологій (практичне застосування ІТ для розв'язування задач з фізики, біології, математики, проєктна діяльність із використанням сучасних технологій); інноваційність (створення власних проєктів, розроблення нових рішень та реалізація власних ідей, розв'язання технічних, професійних, суспільних проблем, креативне використання ІТ-інструментів для автоматизації процесів та аналізу даних); екологічна компетентність (усвідомлене використання сучасних технологій, розуміння впливу ІТ на навколишнє середовище, екологічне використання технічних засобів); навчання впродовж життя (розвиток навичок самоосвіти, робота з відкритими освітніми ресурсами, цифровими бібліотеками); громадянські та соціальні компетентності (через роботу в командах, участь у спільних проєктах, дотримання правил інформаційної безпеки, цифрової культури та відповідальної поведінки в

інтернеті); культурна компетентність (ефективна співпраця з іншими учнями / ученицями під час роботи над проєктом, врахування особливостей кінцевої аудиторії цифрового продукту, зокрема національних, мовних і культурних, дотримання норм авторського права та академічної доброчесності при використанні чужих і створенні власних розробок); раціональність і фінансова грамотність (розробка стратегій та ідей комерціалізації власних проєктів, оцінювання вартості програмних продуктів та вибір оптимальних). Таким чином, Навчальний предмет сприяє інтегрованому формуванню ключових компетентностей, визначених Державним стандартом профільної середньої освіти.

Основними завданнями Навчального предмета є:

- сформувати критичне й системне мислення, навички використання цифрових технологій у повсякденному житті та професійній діяльності;
- сформувати теоретичну базу знань учнів / учениць щодо процесів обробки різноманітних даних та способів їх організації;
- розвивати уміння розв'язувати змістовні задачі різного рівня складності, користуючись відомими теоретичними положеннями, математичним апаратом, інформаційними моделями, алгоритмічними структурами, літературою та комп'ютерною технікою;
- забезпечити засвоєння інформатики на рівні творчого застосування знань;
- сприяти усвідомленню ролі інформаційних технологій у сучасному суспільстві, розвитку власної ідентичності й відповідального ставлення до цифрового середовища;
- формувати в учнів / учениць розуміння, як набуті знання та навички можуть бути використані в обраній професії;
- розвивати здатність до самонавчання, співпраці, обміну досвідом, командної роботи та проєктної діяльності;
- сприяти інтегруванню інформатики з іншими предметами, що викладаються в навчальних закладах.

Зміст Навчального предмета не суперечить *ціннісним орієнтирам*, визначеним Державним стандартом профільної середньої освіти. Теми Навчального предмета та види навчальної діяльності спрямовані на формування поваги до особистості здобувача / здобувачки освіти, підтримку його / її самостійності, інтересів і власного вибору (через варіативність завдань, індивідуальні та групові проєкти). Під час навчання акцентується увага на дотриманні принципів академічної доброчесності, розвитку логічного і критичного мислення, відповідального ставлення до навчання. Спільне розв'язування проблем, участь у дискусіях і презентаціях результатів навчає взаємоповазі, співпраці та конструктивному діалогу. Таким чином, вивчення інформатики сприяє розвитку вільної, ініціативної особистості, готової до свідомого вибору освітньої й професійної траєкторії та активної участі в житті суспільства.

Основними принципами, на яких ґрунтується Програма, є:

- компетентнісний підхід: основою для програми є досягнення компетентностей, передбачених Державним стандартом у інформатичній освітній галузі;

- цілісність: послідовне викладення тем з урахуванням їхніх взаємозв'язків з іншими предметами, зокрема математичної освітньої галузі, для досягнення всіх очікуваних результатів навчання учнів / учениць, поглибленого рівня інформатичної освітньої галузі, базових знань і компетентнісного потенціалу галузі;

- практична зорієнтованість: у програмі передбачена значна кількість практичної діяльності учнів / учениць та їхнє заохочення до проєктної діяльності, виконання завдань практичного змісту;

- врахування вікових та індивідуальних особливостей розвитку й потреб учнів / учениць: здійснюється через послідовне опанування тем і набуття відповідних компетентностей, створення умов для подолання освітніх розривів чи втрат, формування інтересу до інформатики, а також сприяння подальшому особистісному та професійному розвитку учнів / учениць.

Ключовими пріоритетами викладання Навчального предмета є:

- формування компетентностей учнів / учениць в інформатичній галузі, відповідно до вимог Державного стандарту;

- збалансоване поєднання практичної та теоретичної (лекційної) складових навчання;

- формування й усвідомлення взаємозв'язків між інформаційними технологіями, цифровими інструментами та реальними життєвими й професійними завданнями;

- розвиток критичного й алгоритмічного мислення, цифрової грамотності та навичок використання інформаційних технологій для розв'язання прикладних задач;

- формування основ цифрової компетентності як ресурсу для подальшого професійного розвитку у сфері ІТ;

- дотримання принципів академічної доброчесності серед учительства та учнівства.

Структура Програми охоплює три роки послідовного навчання у 10, 11 та 12 класах і містить дев'ятнадцять тем, серед яких передбачено розробку та захист учнями / ученицями проєктів з дослідження й візуалізації алгоритмів, відповідно до одного з принципів побудови програм.

У 10 класі Програма передбачає оволодіння учнями / ученицями технікою реалізації нескладних консольних додатків мовою програмування C++. Замість C++ можна обрати будь-яку імперативну мову програмування з чіткою типізацією. Здобуття теоретичних знань та практичних навичок відбувається за двома змістовними лініями: базові алгоритмічні конструкції та типи даних.

У 10 класі вивчаються 7 тем:

1. Лінійна алгоритмічна структура.
2. Алгоритмічна структура вибору.
3. Циклічна алгоритмічна структура.

4. Структурне програмування. Процедурне програмування.
5. Масив – фундаментальна структура даних.
6. Обробка текстової інформації. Символи та рядки.
7. Двовимірні масиви. Структури, масиви структур.

В 11 класі програма передбачає знайомство учнів / учениць з фундаментальним напрямком «Алгоритми та структури даних». Одна з основних змістовних ліній – використання математичного апарату для аналізу та доведення правильності алгоритмів. В 11 класі вивчаються 6 тем:

1. Прості структури даних: список, стек, черга, дек.
2. Сортювання та пошук.
3. Алгоритми теорії графів.
4. Алгоритми теорії чисел.
5. Комбінаторні алгоритми.
6. Дослідження алгоритмів. Проєкти.

У 12 класі програма передбачає поглиблене вивчення алгоритмів оптимізації та просунутих структур даних. Наводяться приклади використання та застосування алгоритмів в економіці, генетиці, логістиці тощо. У 12 класі вивчаються 6 тем:

1. Динамічне програмування.
2. Алгоритми на графах.
3. Рядкові величини.
4. Обчислювальна геометрія.
5. Введення в об'єктно-орієнтоване програмування, бібліотеки та фреймворки.
6. Дослідження та візуалізація алгоритмів. Проєкти.

Програма розрахована на вивчення Навчального предмета по 1 годині на тиждень у І семестрі 10 класу та по 3 години на тиждень у всіх інших семестрах. Програма передбачає можливість збільшення годин на вивчення Навчального предмета до 2 годин на тиждень у 10 класі за рахунок навчальних годин для обов'язкових освітніх компонентів за обраним профілем (навчальні предмети / інтегровані курси / міжгалузеві інтегровані курси / проведення індивідуальних консультацій, групових занять та проєктної діяльності). У випадку вказаного збільшення годин варто використати їх на проведення практичних та компетентісно орієнтованих занять, вивчення додаткових понять та тем, організацію проєктної діяльності з інформатики та її застосування.

Модельна навчальна програма навчального предмета «Інформатика. 10–12 класи. Поглиблений рівень» є одним із засобів формування не тільки освітнього, а й розвивального та інтелектуального потенціалу особистості. У процесі профільного вивчення інформатики основні завдання Навчального предмета значно розширюються та доповнюються, що обумовлено необхідністю виявлення та розвитку в учнів / учениць логічних здібностей, підготовки їх до участі в інтелектуальних змаганнях та наукових дискусіях, формування в них стійкого інтересу до інформатики і пов'язаної з нею професійної діяльності, підготовки до навчання у вищих навчальних закладах.

ОСНОВНА ЧАСТИНА

10 клас. I семестр.

Кількість годин на тиждень: 1.

Кількість годин на семестр: 16.

Очікувані результати навчання	Пропонований зміст навчального предмета	Види навчальної діяльності
Тема 1. Лінійна алгоритмічна структура. Кількість годин: 4.		
<i>Учень / учениця:</i> - оцінює можливості, функціонал, обмеження, переваги використання інформаційних технологій для вирішення проблем обраної галузі [12 ІФО 1.1.1-1 П]; - пояснює та наводить аргументи доцільності та ефективності використання цифрових пристроїв / інформаційних технологій в обраній галузі для вирішення життєвих, зокрема професійних, задач [12 ІФО 1.1.1-2 П]; - наводить приклади застосування, пояснює перспективи використання та обмеження сучасних інформаційних технологій, а також оцінює їх вплив на власне життя і суспільство [12 ІФО 1.1.2-1 П]; - аналізує інформаційні системи та їх компоненти, описує та порівнює їх функціонування і принципи організації [12 ІФО 3.2.1-1 П]; - компонує інформаційну систему під задачу та визначає ефективність її функціонування [12 ІФО 3.2.1-2 П];	1.1. Структура програми з інформатики за 10-й клас, критерії оцінювання. Знайомство з системним середовищем: операційна система, облікові записи, тестувальна система. Встановлення та налаштування компіляторів та середовищ розробки для програмування на C/C++. Онлайн-компілятори C++. Інформаційні джерела для вивчення мови програмування C++. Перші програми мовою програмування C++, елементарні обчислення. 1.2. Структура програм мовою програмування C++. Коментарі в програмах. Базове форматування коду. Змінні: ініціалізація і присвоювання. Цілочисельні типи даних: <i>short</i>, <i>int</i> і <i>long</i>. Десяткова система числення, визначення розрядів цілих чисел. Цілочисельне ділення. Ділення із заокругленням. 1.3. Конфлікт імен і простір імен <i>std</i>. Об'єкти <i>cout</i>, <i>cin</i> і <i>endl</i>. Прийоми обробки даних, пов'язаних одиницями виміру часу, відстані, маси, об'єму, грошей тощо. Застосування формула Гауса для обчислення суми послідовних цілих чисел. Кількість та сума чисел на відрізку,	Демонстрація роботи операційної системи та навігації тестувальною системою. Практичне налаштування компілятора та середовища розробки (IDE) для C/C++. Використання онлайн-компіляторів для запуску програм. Пошук та оцінка інформаційних джерел для вивчення C++. Короткі інтерактивні лекції з поясненням синтаксису та типів даних. Демонстрація фрагментів коду та рішень прикладних задач. Практичні вправи на ініціалізацію та присвоювання змінних. Практичні вправи з цілочисельними та числами з плаваючою точкою. Практичні вправи на введення та виведення даних (<i>cin</i>, <i>cout</i>). Обробка даних, пов'язаних з одиницями виміру, грошима, часом тощо. Виконання вправ на арифметичні та геометричні прогресії, пропорції та відсотки.

- надає перевагу програмним засобам і ресурсам з інтерфейсом державною мовою (за наявності) [12 ІФО 3.2.1-3 П]; - будує індивідуальну траєкторію навчання, зокрема для розвитку власної цифрової компетентності та інформаційної культури, використовуючи можливості сучасних цифрових освітніх середовищ [12 ІФО 3.2.1-4 П].	кратних k. 1.4. Типи даних з плаваючою точкою: <i>float</i>, <i>double</i> і <i>long double</i>. Зміна точності та формату виводу значень типу з плаваючою точкою: прапорці групи форматування, маніпулятори, методи (бібліотека <i>iomnip</i>). Арифметичні оператори, інкремент та декремент. Арифметична та геометрична прогресії. Пропорції та відсотки.	Форматування виводу чисел з плаваючою точкою (маніпулятори, <i>iomnip</i>). Обговорення проблемних питань та колективний аналіз рішень. Надання та отримання конструктивного зворотного зв'язку. Виконання діагностичних робіт у формі тестів та листків самоконтролю. Проведення інструктажу з техніки безпеки перед практичними роботами. Рефлексія власних помилок та досягнень, планування навчальної траєкторії. <i>Список орієнтовних практичних завдань.</i> - Встановлення та налаштування IDE для C/C++. - Використання онлайн-компіляторів для запуску програм. - Обчислення суми та середнього значення чисел на відріzkі. - Підрахунок чисел, кратних заданому числу k. - Використання формули Гауса для суми чисел. - Ініціалізація та присвоювання змінних (<i>short</i>, <i>int</i>, <i>long</i>). - Арифметичні операції з числами з плаваючою точкою (<i>float</i>, <i>double</i>, <i>long double</i>). - Використання інкременту та декременту, арифметичних виразів. - Практичні вправи з введення та виведення даних (<i>cin</i>, <i>cout</i>). - Форматування виводу чисел з плаваючою
---	---	--

		точкою (маніпулятори, <i>іотанір</i>). - Обробка даних, пов'язаних із одиницями виміру часу, відстані, маси, об'єму та грошей. - Розв'язання задач із конфліктом імен у програмах.
Тема 2. Алгоритмічна структура вибору. Кількість годин: 4.		
<i>Учень / учениця:</i> - доцільно добирає цифрові пристрої та/або інформаційні технології для забезпечення життєвих потреб, спілкування державною, рідною та іноземною мовами, власного розвитку і навчання [12 ІФО 1.1.1-3 П]; - використовує доступні передові інформаційні технології для розв'язування задач з обраної галузі [12 ІФО 1.1.2-3 П]; - аналізує інформаційні процеси, що здійснюються в інформаційних системах різних галузей [12 ІФО 1.2.1-1 П]; - розуміє та пояснює принципи кодування і декодування цифрових даних, які використовуються в обраній галузі, і враховує їх під час створення інформаційних продуктів [12 ІФО 1.2.2-1 П]; - класифікує, структурує та узагальнює дані з обраної галузі, використовуючи відповідні методи та засоби [12 ІФО 1.2.3-1 П]; - розуміє і враховує особливості (зокрема національні, мовні й культурні) аудиторії, з якою взаємодіє [12 ІФО 4.2.1-1 П]; - розуміє проблему інформаційного перевантаження під час роботи в цифровому	2.1. Основні варіанти базової структури «розгалуження». Умовний оператор. Логічні вирази. Повне та неповне розгалуження. Оператори порівняння. Парність числа. Сортювання декількох чисел. Мінімальне та максимальне число з заданих цифр. 2.2. Логічний тип даних <i>bool</i>. Використання логічного типу даних у розгалуженнях. Логічні оператори: І, АБО, НЕ. Побітове виключне АБО (XOR). Пріоритет логічних операцій. Складні умови. Високосний рік. Перевірка коректності дати та часу. 2.3. Умовний тернарний оператор. Оператор вибору <i>switch</i>, лейбли <i>case</i> та <i>default</i>, оператор <i>break</i>. Геометричні задачі у декартовій системі координат: перевірка приналежності точки різним геометричним фігурам, площа перетину двох прямокутників. 2.4. Конвертація чисел з двійкової системи числення в десяткову і навпаки. Побітові оператори. Правила де Моргана. Формула включень-виключень. Прості числа з діапазону від 1 до 100. Задачі на шаховій дошці: клітинки одного кольору, коректні ходи шахових фігур.	Розробка та презентація коротких інтерактивних лекцій з поясненням синтаксису та можливостей C++. Демонстрація написання фрагментів коду та розв'язання прикладних задач. Практичні вправи на застосування умовних операторів, логічних виразів і побітових операцій. Виконання індивідуальних практичних завдань на перевірку умов, обчислення та обробку даних. Робота в групах над задачами підвищеної складності (геометричні, комбінаторні, шахові). Організація обговорення проблемних питань та різних способів розв'язання задач. Створення й використання наочних прикладів, схем і блок-схем алгоритмів. Надання конструктивного зворотного зв'язку та аналіз помилок у роботах учнів/учениць. Проведення діагностичних робіт у формі тестів, анкетувань, листків самоконтролю. Використання онлайн-компіляторів та

середовищі та описує його наслідки [12 ІФО 4.1.1-1 П].		середовищ розробки для перевірки власних рішень. Самостійне опрацювання додаткових матеріалів (статей, відео, документації). <i>Список орієнтовних практичних завдань.</i> - Перевірка парності числа. - Пошук мінімального та максимального числа серед введених значень. - Сортювання трьох і більше чисел за зростанням/спаданням - Визначення більшого з двох чисел за допомогою тернарного оператора. - Обчислення оцінки/категорії за введеним балом (умовне розгалуження). - Перевірка високосного року. - Перевірка правильності введеної дати та часу. - Визначення кількості днів у місяці з урахуванням високосного року. - Визначення пори року за номером місяця. - Перевірка належності точки до квадрата чи кола у декартовій системі координат. - Обчислення площі перетину двох прямокутників. - Переведення чисел з двійкової системи в десяткову і навпаки. - Перевірка простих чисел у діапазоні 1–100. - Визначення кольору клітинки на шахівниці за координатами. - Перевірка коректності ходу шахових фігур (кінь, тура, ферзь).
Тема 3. Циклічна алгоритмічна структура.		

Кількість годин: 4.		
<i>Учень / учениця:</i> - обирає ефективний спосіб пошуку, опрацювання та візуалізації структурованих даних [12 ІФО 1.2.3-2 П]; - описує можливі причини типових апаратних або програмних збоїв і способи їх усунення [12 ІФО 3.2.2-1 П]; - аналізує інформаційні системи та їх компоненти, описує та порівнює їх функціонування і принципи організації [12 ІФО 3.2.1-1 П]; - будує і використовує інформаційні моделі для розв'язання проблем реального та віртуального світу, зокрема моделі «сутність – зв'язок», із застосуванням інформаційних технологій [12 ІФО 1.3.1-2 П]; - пропонує критерії вибору, обґрунтовано обирає, налаштовує та ефективно використовує технологічні засоби для індивідуальної або групової діяльності та комунікації [12 ІФО 3.3.1-1 П]; - діагностує апаратні або програмні збої, усуває типові несправності під контролем учителя [12 ІФО 3.2.2-2 П]; - надає перевагу програмним засобам і ресурсам з інтерфейсом державною мовою (за наявності) [12 ІФО 3.2.1-3 П]; - описує способи регулювання власного інформаційного навантаження і застосовує їх (обмеження, тайм-менеджмент, відключення певних функцій програм, перемикання на інші види діяльності, фізичні вправи тощо)	3.1. Поняття циклічного алгоритму. Блок-схеми циклічних алгоритмів. Цикл з передумовою <i>while</i>. Цикл з післяумовою <i>do ... while</i>. Нескінченні цикли. Оператори <i>break</i> і <i>continue</i>. Теоретичні задачі: кількість цифр у числі, сума цифр числа, сума цифр двійкового представлення числа, переворот числа. 3.2. Зчитування заданої кількості даних. Зчитування даних до термінатора. Зчитування даних до кінця файлу. Обробка послідовностей: кількість елементів, сума, максимум, мінімум, другий максимум, кількість максимумів тощо. 3.3. Цикл з параметром <i>for</i>. Нарахування складних відсотків. Похибки комп'ютерних обчислень. Послідовності та закономірності. Обчислення сум та добутків елементів послідовностей. Генерація ступенів двійки. Факторіал числа, подвійний факторіал. Числа Фібоначчі. 3.4. Вкладені цикли. Геометричні фігури з символів у консолі: квадрат, прямокутник, трикутник. Ітераційні процеси. Гіпотеза Коллатца. Організація перебору для розв'язування діофантових рівнянь з двома та трьома невідомими.	Короткі інтерактивних мінілекцій (із прикладами на C++). Демонстрація написання коду в середовищі розробки (пояснення помилок, тестування). Колективне складання блок-схем алгоритмів циклів. Розв'язування задач на дошці/екрані з покроковим аналізом. Робота в групах для обговорення проблемних ситуацій (наприклад, «нескінченний цикл»). Виконання практичних вправ у середовищі програмування. Створення власних алгоритмів і порівняння різних способів реалізації. Тестування та діагностика програм (виявлення помилок, налагодження). Виконання домашніх завдань з поступовим ускладненням рівня. Проведення мінітестів та листків самоконтролю. Формування практичних навичок роботи з IDE, налаштування компілятора. Інструктаж з техніки безпеки перед практичною роботою. <i>Список орієнтовних практичних завдань.</i> - Визначення кількості цифр у числі, суми цифр числа. - Визначення чисел-паліндромів. - Створення нескінченного циклу та завершення його оператором <i>break</i>.

[12 ІФО 4.1.1-2 П].		- Використання оператора <i>continue</i> для пропуску ітерації. - Зчитування N чисел, знаходження суми та середнього арифметичного. - Пошук найбільшого та найменшого елемента. - Пошук другого максимуму. - Визначення кількості елементів, що дорівнюють максимуму. - Обчислення складних відсотків (модель банківського депозиту). - Обчислення суми квадратів перших N чисел. - Обчислення факторіала, подвійного факторіала. - Обчислення чисел Фібоначчі. - Генерація ступенів двійки. - Дослідження гіпотези Коллатца. - Перебір значень для розв'язування діофантових рівнянь (2–3 невідомих). - Ітераційні методи обчислень (наприклад, наближення до квадратного кореня).
Тема 4. Структурне програмування. Процедурне програмування. Кількість годин: 4.		
<i>Учень / учениця:</i> - пояснює та наводить аргументи доцільності та ефективності використання цифрових пристроїв / інформаційних технологій в обраній галузі та для вирішення життєвих задач [12 ІФО 1.1.1-2 П]; - пояснює та враховує переваги і ризики використання конкретних сучасних інформаційних технологій в обраній галузі	4.1. Вкладені алгоритмічні конструкції. Концепція структурного програмування: пакетування та вкладення. Теорема Бьома-Якопіні. Налаштування програм: степпінг і точки зупину. Поняття складності алгоритму. Приклади реалізації функцій: сума цифр числа, переворот числа, паліндром. 4.2. Підпрограми, функції. Блоки стейтментів. Локальні та глобальні змінні, область видимості	Короткі інтерактивні лекції з прикладами використання вкладених алгоритмічних конструкцій та підпрограм. Демонстрація написання та покрокового налаштування фрагментів коду (<i>stepping, breakpoints</i>). Розробка та виконання практичних завдань, що ілюструють складність алгоритмів і роботу функцій з різними

[12 ІФО 1.1.2-2 П]; - висловлює прогнози щодо розвитку обраної галузі під впливом інформаційних технологій [12 ІФО 1.1.2-4 П]; - наводить приклади факторів, які впливають на розподіл та використання цифрових ресурсів у глобальному суспільстві [12 ІФО 1.1.3-1 П]; - використовує галузеву термінологію та розширює словниковий запас, спілкуючись державною, рідною та іноземною мовою на тему інформаційних технологій [12 ІФО 1.1.3-4 П]; - відстежує появу нових способів і інструментів, а також пропонує власні стратегії перевірки достовірності інформації та надійності джерел [12 ІФО 1.4.1-2 П]; - обґрунтовано обирає бібліотеки або інші зовнішні модулі для розв'язування задач з обраної галузі [12 ІФО 2.3.1-2 П]; - використовує цифрові засоби та технології для забезпечення особистої та громадської безпеки в умовах надзвичайних ситуацій [12 ІФО 4.1.1-3 П].	та тривалість життя змінної. Статичні змінні. Приклади реалізації функцій: коректність дати, час через 1 секунду, видалення найменшої цифри. 4.3. Параметри й аргументи функцій. Передача аргументів за значенням, за посиланням та за адресою. Приклади реалізації функцій: найближче менше, кратне Р; найближче більше, кратне Р; кількість чисел, кратних Р; сума чисел, кратних Р.	параметрами. Організація дискусій щодо ефективності використання різних підходів у програмуванні (структурне програмування, використання підпрограм, бібліотек). Робота в групах над створенням і тестуванням невеликих проєктів (наприклад, «програмний калькулятор» із застосуванням функцій). Використання середовищ розробки для закріплення практичних навичок. Проведення діагностичних робіт: тести з теоретичного матеріалу, листки самоконтролю з аналізом помилок. Надання взаємооцінювання та конструктивного зворотного зв'язку в парах / групах. Виконання практичних і домашніх завдань з поступовим ускладненням. <i>Список орієнтовних практичних завдань.</i> - Створення бібліотеки користувацьких функцій для роботи з числами. - Розробка власної функції для перевірки спеціальних умов (наприклад, «щасливий квиток»). - Аналіз складності обраного алгоритму та пояснення вибору його реалізації. - Підготовка короткого огляду про переваги та ризики використання алгоритмів чи бібліотек у сучасному програмуванні. - Самостійне дослідження й представлення нового інструмента налагодження або
---	--	--

		бібліотеки для C++. - Створення інтерактивної вправиб(наприклад, тест чи гру для перевірки знань).
--	--	---

10 клас. II семестр.

Кількість годин на тиждень: 3.

Кількість годин на семестр: 57.

Очікувані результати навчання	Пропонований зміст навчального предмета	Види навчальної діяльності
Тема 1. Масив – фундаментальна структура даних. Кількість годин: 20.		
<i>Учень / учениця:</i> - класифікує, структурує та узагальнює дані з обраної галузі, використовуючи відповідні методи та засоби [12 ІФО 1.2.3-1 П]; - аналізує та оцінює доказовість і вагу (сильні – слабкі) аргументів у текстах, своїх твердженнях і судженнях інших [12 ІФО 1.4.1-3 П]; - аналізує інформаційні системи та їхні компоненти, описує й порівнює їх функціонування та принципи організації [12 ІФО 3.2.1-1 П]; - будує індивідуальну траєкторію навчання, зокрема для розвитку власної цифрової компетентності та інформаційної культури, використовуючи можливості сучасних цифрових освітніх середовищ [12 ІФО 3.2.1-4 П]; - пояснює принципи функціонування різних цифрових пристроїв і їх розвиток [12 ІФО 3.1.1-1 П]; - описує можливі причини типових апаратних чи програмних збоїв та способи їх усунення [12 ІФО 3.2.2-1 П]; - обґрунтовано обирає, налаштовує та	1.1. Статичні масиви. Адресна арифметика і індексація масивів. Введення та виведення одновимірних числових масивів. Базові алгоритми для роботи з масивами: сума, максимум, мінімум, середнє арифметичне. Реалізація власних функцій для обробки масивів. 1.2. Лінійний пошук в одновимірних масивах. Вставка елемента в масив. Видалення елемента масиву. Переворот масиву. Обробка підмасивів. 1.3. Динамічні масиви. Виділення пам'яті під масив, видалення масиву (оператори <i>new</i> та <i>delete</i>). Контейнери <i>std::array</i> та <i>std::vector</i>. Цикл <i>foreach</i> та ключове слово <i>auto</i>. Поняття про ітератори. Ітератори стандартної бібліотеки C++. 1.4. Квадратичні алгоритми сортування одновимірних масивів: сортування вибором, сортування «бульбашкою», сортування вставками. Поняття компаратору. Сортування з компаратором: сортування модулів, сортування за сумою цифр тощо..	Мінілекцій із прикладами використання масивів у C++. Демонстрація написання фрагментів коду: введення/виведення масивів, пошук, сортування. Виконання індивідуальних та групових практичних завдань у середовищі програмування. Розв'язання навчальних задач із поступовим ускладненням (від базових операцій до алгоритмів сортування). Організація обговорення проблемних питань (чому деякі алгоритми працюють повільніше/швидше). Проведення інструктажів із техніки безпеки та правильного використання IDE. Надання індивідуального зворотного зв'язку, колективний аналіз помилок. Створення схем і порівняльних таблиць (наприклад, різниця між <i>std::array</i> та <i>std::vector</i>). Проведення діагностичних робіт (онлайн-тести, листки самоконтролю). Розробка власних мініпроектів, що включають використання масивів

ефективно використовує технологічні засоби для індивідуальної або групової діяльності та комунікації [12 ІФО 3.3.1-1 П]; інформаційних систем і можливих наслідків зловмисного їх використання [12 ІФО 4.1.2-1 П].		(самостійна діяльність учнів/учениць). Список орієнтовних практичних завдань. - Базові операції з масивами: обчислити суму, мінімум, максимум, середнє. - Лінійний пошук: знайти перший і останній елемент, що задовольняє умові. - Модифікація масиву: вставка та видалення елементів, переворот масиву, робота з підмасивами. - Динамічні масиви: створення за допомогою <i>new/delete</i>, використання <i>std::array</i> і <i>std::vector</i>. - Ітератори: демонстрація обходу масиву через цикл <i>foreach</i> та <i>auto</i>. - Алгоритми сортування: реалізація сортування вибором, «бульбашкою», вставками. - Сортування з компараторами: сортування масиву за модулем, за сумою цифр. - Мініпроект: програма, яка вводять масив чисел і виводить набір характеристик (мін/макс, середнє, відсортований список).
Тема 2. Обробка текстової інформації. Символи та рядки. Кількість годин: 17.		
<i>Учень / учениця:</i> - класифікує, структурує та узагальнює дані з обраної галузі, використовуючи відповідні методи та засоби [12 ІФО 1.2.3-1 П]; - аргументовано обстоює використання надійних джерел і достовірної інформації [12 ІФО 1.4.1-4 П]; - обґрунтовано обирає формати файлів під час створення та збереження інформаційних	2.1. Символьний тип даних <i>char</i>. Американський стандартний код для інформаційного обміну <i>ASCII</i>. Англійський алфавіт, цифри, регістр символів. Зміна регістру символів. 2.2. Рядкові класи <i>std::string</i> і <i>std::wstring</i>. Створення, знищення і конвертація <i>std::string</i>. Довжина і ємність <i>std::string</i>. Доступ до символів <i>std::string</i>. Конвертація <i>std::string</i> в рядки <i>C-style</i>. Присвоювання і перестановка значень з	Інтерактивні мінілекції з поясненням принципів роботи з <i>char</i>, <i>std::string</i> та <i>std::wstring</i>. Демонстрація написання фрагментів коду (змін регістру символів, конвертація між форматами, робота з файлами). Розробка практичних вправ для закріплення навичок роботи з рядками й символами.

продуктів з урахуванням принципів кодування цифрових даних, що використовуються в обраній галузі [12 ІФО 1.2.2-4 П]; - пояснює різні способи подання об'єктів галузі [12 ІФО 1.3.1-1 П]; - обґрунтовано та доцільно використовує функції, класи або інші компоненти в програмних проектах [12 ІФО 2.3.1-3 П]; - розуміє та пояснює принципи кодування та декодування цифрових даних, що використовуються в обраній галузі, та враховує їх під час створення інформаційних продуктів [12 ІФО 1.2.2-1 П]; - обґрунтовано обирає, налаштовує та ефективно використовує технологічні засоби для індивідуальної або групової діяльності та комунікації [12 ІФО 3.3.1-1 П]; - розуміє проблему інформаційного перевантаження під час роботи в цифровому середовищі та описує його наслідки [12 ІФО 4.1.1-1 П]; - критично оцінює нові інформаційні технології щодо їх відповідності засадам кібербезпеки [12 ІФО 4.1.2-5 П].	<i>std::string</i>. Вставка символів і рядків у <i>std::string</i>. 2.3. Виділення фрагментів у рядку. Обробка фрагментів тексту. Виділення слів у рядку. Робота з текстовими файлами. Арифметика багаторозрядних чисел: реалізація основних арифметичних операцій.	Організація обговорення проблемних питань. Надання зворотного зв'язку й аналіз розв'язків учнів/учениць. Формування практичних навичок роботи з IDE та компіляторами C++. Проведення діагностичних робіт (онлайн-тестів або листків самоперевірки). Робота в групах для реалізації невеликих проектів (наприклад, текстовий калькулятор або пошук слова в рядку). Виконання практичних та домашніх завдань. <i>Список орієнтовних практичних завдань.</i> - Написати програму для відображення ASCII-кодів англійських літер та цифр. - Створити функцію для зміни регістру символів. - Розробити мініпрограму, яка перевіряє, чи символ є цифрою/літерою/знаком пунктуації. - Написати програму, що приймає рядок від користувача й виводить його довжину та ємність. - Реалізувати функцію пошуку, вставки й заміни підрядка у визначену позицію рядка. - Виконати обмін значень між двома рядками. - Програма для виділення окремих слів у рядку. - Зчитати з файлу текст, порахувати кількість слів, речень, символів.
--	---	--

		- Реалізувати додавання багаторозрядних чисел, записаних у вигляді рядків, множення двох великих чисел (символьне подання).
Тема 3. Двовимірні масиви. Структури, масиви структур. Кількість годин: 20.		
<i>Учень / учениця:</i> - задовольняє власні інформаційні потреби, активно шукає нові рішення та ідеї, зокрема читаючи та аналізуючи текстові джерела [12 ІФО 1.2.1-2 П]; - пояснює принципи стиснення даних, обґрунтовано обирає метод стиснення, урахувуючи можливість їх втрати [12 ІФО 1.2.2-2 П]; - пояснює особливості та загальні принципи подання даних у файлах різних форматів [12 ІФО 1.2.2-3 П]; - класифікує, структурує та узагальнює дані з обраної галузі, використовуючи відповідні методи та засоби [12 ІФО 1.2.3-1 П]; - аналізує та оцінює інформаційну модель для розв'язування задачі за власними критеріями [12 ІФО 1.3.1-3 П]; - пояснює можливості впливу цифрової репутації на взаємини в реальному світі [12 ІФО 4.1.3-2 П]; - конфіденційності, цілісності та доступності даних [12 ІФО 4.1.2-2 П]; - інтегрує компоненти програмного проекту [12 ІФО 2.3.1-4 П].	3.1. Двовимірний масив або матриця. Введення та виведення матриць. Індксація елементів двовимірних масивів. Заповнення двовимірних масивів. Базові алгоритми для роботи з матрицями: сума, максимум, мінімум, середнє арифметичне, обробка окремих стовпчиків та рядків. 3.2. Робота з динамічними двовимірними масивами: виділення пам'яті, введення, виведення. Сортування рядків / стовпчиків двовимірного масиву. Транспонування, поворот, віддзеркалення двовимірних масивів. Реалізація власних функцій для обробки матриць. 3.3. Перерахування (перелічуваний тип даних). Класи <i>enum</i>. Структури та пари. Масиви структур та пар. Індксація та доступ до елементів масиву структур. Двовимірні масиви з символів, рядків, структур тощо. 3.4. Підсумкова лекція з основ програмування мовою C++. Анонс курсу «Алгоритми та структури даних».	Інтерактивні мінілекції із поясненням можливостей C++ (двовимірні масиви, <i>enum</i>, структури). Демонстрація написання та покрокового розбору фрагментів коду. Виконання практичних завдань: введення та виведення матриць, пошук суми/максимуму/мінімуму, середнє арифметичне, обробка рядків і стовпчиків. Завдання на динамічні двовимірні масиви: виділення та звільнення пам'яті, сортування рядків і стовпців, транспонування, поворот, віддзеркалення. Реалізація власних функцій для обробки матриць. Завдання з <i>enum</i>, структур та масивів структур: опис, індксація, доступ до елементів. Робота з двовимірними масивами символів і рядків. Створення простих програм, що демонструють принципи стиснення даних та роботу з файлами різних форматів. Групова робота над складними практичними завданнями (наприклад, створення невеликої системи для обліку даних у вигляді структур і масивів).

		Організація дискусій і обговорення проблемних питань (оптимальні алгоритми обробки матриць, порівняння способів збереження даних). Самостійна розробка мініпроектів: програма-«калькулятор матриць», редактор таблиць на масивах структур, простий аналізатор текстових даних. Підготовка та захист презентацій учнівських проектів. Проведення діагностичних тестів, анкетувань та листків самоконтролю для оцінки засвоєння матеріалу. Надання індивідуального та групового зворотного зв'язку, аналіз помилок. Виконання домашніх і практичних завдань для закріплення тем. <i>Список орієнтовних практичних завдань.</i> - Введення та виведення двовимірних масивів (матриць). - Знаходження суми, мінімального та максимального елементів у матриці. - Обчислення середнього арифметичного рядка/стовпця. - Транспонування матриці. - Поворот та віддзеркалення матриці. - Сортування рядків і стовпців матриці. - Робота з динамічними двовимірними масивами (виділення та звільнення пам'яті). - Реалізація функцій для обробки матриць (пошук, обчислення, сортування). - Опис та використання перерахувань
--	--	---

		(enum). - Створення та обробка структур. - Використання масивів структур для збереження та пошуку даних. - Робота з двовимірними масивами символів і рядків. - Створення програми для обліку даних (таблиця на масивах структур). - Створення простого редактора таблиць на основі масивів. - Завдання на роботу з файлами: запис та зчитування даних.
--	--	--

11 клас. I семестр.
Кількість годин на тиждень: 3.
Кількість годин на семестр: 48.

Очікувані результати навчання	Пропонований зміст навчального предмета	Види навчальної діяльності
Тема 1. Прості структури даних. Кількість годин: 14.		
<i>Учень / учениця:</i> - аналізує та оцінює інформаційну модель для розв'язування задачі за власними критеріями [12 ІФО 1.3.1-3 П]; - обирає ефективний спосіб пошуку, опрацювання та візуалізації структурованих даних [12 ІФО 1.2.3-2 П]; - інформаційні моделі для розв'язання проблем реального та віртуального світу, зокрема моделі «сутність – зв'язок», із застосуванням інформаційних технологій [12 ІФО 1.3.1-2 П]; - описує етапи виконання комп'ютером коду високого рівня (компіляція, інтерпретація, оптимізація) [12 ІФО 2.2.1-3 П]; - обґрунтовано обирає комплекс програмних засобів для опрацювання даних різних типів у процесі створення інформаційних продуктів [12 ІФО 2.4.1-1 П]; - інформації для особистої репутації або репутації групи, яку представляє [12 ІФО 4.1.3-3 П]; - конструктивно реагує на агресивні	1.1. Актуалізація знань за 10-й клас. Структура програми з інформатики за 11-й клас, критерії оцінювання. Знайомство із системним середовищем: операційна система, облікові записи, тестувальна система. Встановлення та налаштування компіляторів та середовищ розробки для програмування на C/C++. Онлайн-компілятори C++. Інформаційні джерела для вивчення мови програмування C++. Перші програми мовою програмування C++, елементарні обчислення. 1.2. Структура даних <i>стек (stack)</i>: реалізація на масиві, стек із захистом від помилок. Правильна дужкова послідовність (перевірка). Обчислення виразу в польській нотації. Знаходження мінімального елемента в структурі даних <i>stack</i>. STL-контейнер <i>stack</i>. 1.3. Структура даних <i>черга (queue)</i> реалізація на масиві, реалізація на кільцевому списку, черга із захистом від помилок. Знаходження мінімального елемента в структурі даних <i>queue</i>. Структура даних <i>дек (deque)</i>. STL-контейнери <i>queue</i> та <i>deque</i>. Черга з пріоритетами. STL-контейнер <i>priority_queue</i>. 1.4. Ітератори. Зв'язаний список та двобічно	Інтерактивні лекції про структури даних у C++. Демонстрація написання фрагментів коду для роботи зі стеком, чергою, списком та асоціативними контейнерами. Організація практичних робіт з реалізації структур даних на масивах та за допомогою STL-контейнерів. Обговорення алгоритмів обробки даних та методів оптимізації коду. Надання конструктивного зворотного зв'язку та аналіз роботи учнів/учениць. Формування практичних навичок роботи із середовищами розробки C++ (IDE, компілятори, онлайн-компілятори). Створення та розподіл навчальних матеріалів для самостійного опрацювання (шпаргалки, приклади коду, інтерактивні схеми). Проведення діагностичних робіт: тести, анкетування, листки самоконтролю для перевірки засвоєння структури даних. Проведення інструктажів з техніки безпеки перед практичними роботами. Робота в групах для розв'язання складних задач на структури даних. Виконання практичних та домашніх завдань із

висловлювання, зокрема в глобальному інформаційному просторі [12 ІФО 4.2.1-4 П]; - описує способи регулювання власного інформаційного навантаження і застосовує їх (обмеження, тайм-менеджмент, відключення певних функцій програм, перемикання на інші види діяльності, фізичні вправи тощо) [12 ІФО 4.1.1-2 П].	зв'язаний список. STL-контейнер <i>list</i>. Асоціативний масив (<i>map, unordered_map, multimap, unordered_multimap</i>). Множина, мультимножина (<i>set, multiset, unordered_set, unordered_multiset</i>).	самостійним дослідженням алгоритмів. Самостійна реалізація невеликих проєктів із використанням STL-контейнерів. Експерименти з оптимізацією алгоритмів та аналізом продуктивності програм. Обговорення кібербезпеки, цифрової репутації та управління інформаційним навантаженням під час роботи з даними. <i>Список орієнтовних практичних завдань.</i> - Реалізація стеку на масиві: додавання, видалення, перегляд елементів. - Обчислення виразу у польській нотації за допомогою стеку. - Пошук мінімального елемента в стеку, в черзі. - Реалізація черги на масиві та кільцевому списку. - Реалізація дека (<i>deque</i>) та черги з пріоритетом (<i>priority_queue</i>). - Робота з ітераторами: проходження контейнерів, модифікація елементів. - Реалізація однібічного та двобічного зв'язного списку. - Створення масиву структур та обробка даних у ньому. - Використання асоціативних контейнерів (<i>map, unordered_map</i>) для пошуку та зберігання даних. - Робота з множинами (<i>set, multiset, unordered_set</i>) для обробки унікальних елементів. - Проєкт: облік даних користувачів або товарів за допомогою STL-контейнерів. - Використання контейнерів для обробки
---	---	---

		реальних даних (наприклад, статистика, текстові дані, списки завдань).
Тема 2. Сортування та пошук. Кількість годин: 17.		
<i>Учень / учениця:</i> - оцінює можливості, функціонал, обмеження, переваги використання інформаційних технологій для вирішення проблем обраної галузі [12 ІФО 1.1.1-1 П]; - використовує доступні передові інформаційні технології для розв'язування задач з обраної галузі [12 ІФО 1.1.2-3 П]; - пропонує критерії вибору, обґрунтовано обирає, налаштовує та ефективно використовує технологічні засоби для індивідуальної або групової діяльності і комунікації [12 ІФО 3.3.1-1 П]; - обґрунтовано обирає інформаційні технології для реалізації програмних проектів [12 ІФО 2.2.1-2 П]; - створює та вдосконалює інформаційні продукти з використанням кількох інформаційних технологій [12 ІФО 2.4.1-2 П]; - адаптується під потреби аудиторії під час цифрової комунікації, зокрема з використанням програмних засобів штучного інтелекту та перекладу текстів [12 ІФО 4.2.1-2 П]; - порівнює заходи та засоби інформаційної безпеки за наданими критеріями та аргументовано обирає їх [12 ІФО 4.1.2-3 П];	2.1. Постановка задачі сортування даних. Застосування квадратичні алгоритми сортування. 2.2. Сортування злиттям. Принцип «Розділяй та володарюй». Аналіз алгоритму та асимптотика. Модифікації та оптимізації алгоритму. Швидке сортування, алгоритм Гоара. 2.3. Лінійний пошук. Двійковий пошук. Двійкове дерево пошуку. Ітеративна та рекурсивна реалізації. Аналіз алгоритму та асимптотика. Двійковий пошук за відповіддю. Використання двійкового пошуку для розв'язку рівнянь. 2.4. Тернарний пошук. Знаходження максимумів та мінімумів функцій. Дискретний та неперервний варіанти тернарного пошуку. Приклади реалізації тернарного пошуку мовою програмування C++.	Інтерактивні лекції про алгоритми сортування та пошуку (квадратичні алгоритми, злиттям, швидке сортування, лінійний, двійковий та тернарний пошук). Демонстрація написання фрагментів коду з різними алгоритмами, включаючи використання компараторів. Організація практичних вправ з реалізації алгоритмів сортування та пошуку на одновимірних та двовимірних масивах, а також у структурах даних STL. Обговорення переваг та обмежень різних алгоритмів, аналіз їх асимптотики та ефективності. Формування практичних навичок роботи із середовищами розробки та налагодження програм. Створення та розподіл навчальних матеріалів: блок-схеми алгоритмів, порівняльні таблиці, приклади коду. Робота в групах для розв'язання складних практичних завдань та порівняння результатів. Виконання практичних та домашніх завдань з кодування алгоритмів. <i>Список орієнтовних практичних завдань.</i> - Реалізація сортування «бульбашкою», вставками та вибором для числових масивів. - Сортування масивів із використанням компаратора (за модулем, за сумою цифр).

- наводить приклади стратегій запобігання та протидії кібербулінгу та застосовує їх за потреби [12 ІФО 4.2.1-5 П].		- Реалізація сортування злиттям і швидкого сортування. - Лінійний пошук у масиві чисел та рядків. - Двійковий пошук (ітеративна та рекурсивна реалізації). - Застосування двійкового пошуку для розв'язку рівнянь. - Тернарний пошук для дискретних та неперервних функцій. - Порівняння ефективності різних алгоритмів сортування на великих масивах. - Робота з двовимірними масивами: сортування рядків та стовпчиків, пошук максимумів і мінімумів. - Реалізація алгоритмів на STL-контейнерах (<i>vector</i>, <i>array</i>).
Тема 3. Алгоритми теорії графів. Кількість годин: 17.		
<i>Учень / учениця:</i> - задовольняє власні інформаційні потреби, активно шукає нові рішення та ідеї, зокрема читаючи та аналізуючи текстові джерела [12 ІФО 1.2.1-2 П]; - пояснює особливості та загальні принципи подання даних у файлах різних форматів [12 ІФО 1.2.2-3 П]; - класифікує, структурує та узагальнює дані з обраної галузі, використовуючи відповідні методи та засоби [12 ІФО 1.2.3-1 П]; - компонує інформаційну систему під задачу та визначає ефективність її функціонування [12 ІФО 3.2.1-2 П];	3.1. Базові поняття теорії графів. Способи подання та зберігання графів: матриця суміжності, список ребер, список суміжності. Орієнтовані та неорієнтовані графи. Кратні ребра у неорієнтованому та в орієнтованому графі. Визначення ступенів та напівступенів вершин графа за матрицею суміжності та за списком ребер. Джерела та стоки в графі. Регулярні графи. Повні та напівповні графи, турніри. Транзитивність неорієнтованих та орієнтованих графів. Способи подання зважених графів. 3.2. Пошук у глибину (<i>DFS</i>, <i>Depth-First Search</i>). Реалізація алгоритму <i>DFS</i> мовою програмування C++. Компоненти зв'язності	Опрацювання основ теорії графів, способів подання графів, алгоритмів DFS, BFS, топологічного сортування та алгоритму Дейкстри на основі опорних конспектів. Демонстрація реалізації алгоритмів на C++ із використанням масивів, списків, STL-контейнерів та структур даних. Організація практичних вправ для відпрацювання побудови графів та застосування алгоритмів пошуку і сортування на графах. Обговорення переваг та обмежень різних способів зберігання графів і алгоритмів пошуку. Надання конструктивного зворотного зв'язку на виконані завдання та налагодження коду. Формування практичних навичок роботи із

- формулює очікуваний результат розробки та вимоги до програмного проекту для розв'язування задач з обраної галузі [12 ІФО 2.2.1-1 П]; - обирає доречні типи даних для створення інформаційних продуктів [12 ІФО 2.4.2-1 П]; - наводить приклади та розуміє упередження, пов'язані з використанням інформаційних технологій, і їх наслідки [12 ІФО 4.2.1-6 П]; - застосовує засоби та стратегії захисту конфіденційності та цілісності даних [12 ІФО 4.1.2-4 П].	графа. Перевірка, чи є граф деревом. Перевірка, чи є граф дводольним. 3.3. Топологічне сортування. Алгоритм Кана, приклад роботи алгоритму. Алгоритм Тарьяна. Реалізація алгоритмів мовою програмування. 3.4. Пошук у ширину (<i>BFS, Breadth-First search</i>). Хвильовий алгоритм. Знаходження виходу з лабіринту. Найкоротший шлях коня. Використання черги. Реалізація алгоритмів мовою програмування. 3.5. Алгоритм Дейкстри. Формулювання задачі на прикладі пошуку оптимального маршруту на карті. Обмеження алгоритму. Реалізація алгоритму з асимптотикою $O(V*V)$, де V – кількість вершин в графі.	середовищами розробки та налагодження програм. Створення та розподіл навчальних матеріалів: схеми графів, приклади коду, порівняльні таблиці алгоритмів. Робота в групах для розв'язання складних практичних завдань та обговорення результатів. Виконання практичних та домашніх завдань з побудови графів та реалізації алгоритмів. Проведення тестування та діагностичних робіт (тести, листки самоконтролю, анкетування). <i>Список орієнтовних практичних завдань.</i> - Створення орієнтованих і неорієнтованих графів. - Реалізація графів через матрицю суміжності, список ребер, список суміжності. - Визначення ступенів та напівступенів вершин, пошук джерел та стоків. - Реалізація алгоритму DFS та перевірка компонентів зв'язності графа. - Перевірка графа на дерево та на дводольність. - Реалізація топологічного сортування (алгоритми Кана та Тарьяна). - Реалізація BFS та хвильового алгоритму для пошуку найкоротшого шляху. - Знаходження виходу з лабіринту та найкоротшого шляху коня на шахівниці. - Реалізація алгоритму Дейкстри для знаходження оптимального маршруту на карті. - Аналіз асимптотики алгоритмів і порівняння ефективності різних методів пошуку.
--	---	---

11 клас. II семестр.
Кількість годин на тиждень: 3.
Кількість годин на семестр: 57.

Очікувані результати навчання	Пропонований зміст навчального предмета	Види навчальної діяльності
Тема 1. Алгоритми теорії чисел. Кількість годин: 19.		
<i>Учень / учениця:</i> - пояснює можливості та обмеження комп'ютерного моделювання [12 ІФО 1.3.1-4 П]; - демонструє здатність до самостійного опанування нових пристроїв і технологій [12 ІФО 3.1.1-3 П]; - аргументовано обирає структури даних для розв'язування задач з обраної галузі [12 ІФО 2.1.3-1 П]; - інтегрує різнотипні дані під час створення інформаційних продуктів [12 ІФО 2.4.2-2 П]; - застосовує принципи декомпозиції та синтезу під час розв'язування задач з обраної галузі [12 ІФО 2.3.1-1 П]; - застосування різних цифрових пристроїв для розв'язування задач з обраної галузі та використовує їх [12 ІФО 3.1.1-2 П]; - оцінює інформаційні технології та системи за критеріями інклюзивності, доступності та рівності [12 ІФО 4.2.2-3 П]; - описує способи регулювання власного інформаційного навантаження і застосовує їх (обмеження, тайм - менеджмент,	1.1. Прості числа. Перевірка числа на простоту. Факторизація. Арифметика залишків. Арифметичні мультиплікативні функції: кількість та сума дільників, функція Ейлера. Застосування теорії чисел у криптографії. 1.2. Найбільший спільний дільник. Найменше спільне кратне. Алгоритм Евкліда. Аналіз алгоритмів обчислення найбільшого спільного дільника. Реалізація алгоритмів мовою програмування C++. Бінарний алгоритм знаходження найбільшого спільного дільника. 1.3. Решето Ератосфена. Решето Сундарами. Решето Аtkінса. Аналіз алгоритмів, реалізація мовою програмування C++. 1.4. Швидке піднесення до степеня. Аналіз алгоритму, асимптотика. Ітераційна та рекурсивна реалізації. Використання швидкого піднесення до степеня для обчислення чисел Фібоначчі за модулем. 1.5. Мала теорема Ферма. Теорема Ейлера. Обчислення оберненого за модулем числа. Аналіз та реалізація алгоритмів. Обчислення сполуки за модулем. Китайська теорема про залишки.	Інтерактивні лекції про прості числа, алгоритми знаходження НСД/НСК. Демонстрація написання фрагментів коду на C++ для перевірки простоти числа, факторизації, обчислення НСД/НСК, решет Ератосфена, Сундарами, Аtkінса, швидкого піднесення до степеня та модульної арифметики. Організація обговорення прикладів застосування теорії чисел у криптографії та захисті даних. Розробка практичних вправ для застосування алгоритмів на прикладах числових задач. Надання конструктивного зворотного зв'язку на виконані завдання та аналіз помилок. Формування практичних навичок роботи із середовищами розробки та налагодження коду. Створення та розподіл навчальних матеріалів: блок-схеми алгоритмів, приклади коду, таблиці обчислень. Робота в групах для розв'язання складних практичних завдань. Виконання практичних та домашніх завдань із застосуванням різних структур даних та алгоритмів.

відключення певних функцій програм, перемикання на інші види діяльності, фізичні вправи тощо) [12 ІФО 4.1.1-2 П]; - аргументовано пояснює суспільні та правові наслідки порушення прав інтелектуальної власності [12 ІФО 4.3.1-5 П].		<i>Список орієнтовних практичних завдань.</i> - Перевірка простих чисел та реалізація факторизації. - Обчислення арифметичних мультиплікативних функцій: кількість і сума дільників, функція Ейлера. - Реалізація алгоритму Евкліда та бінарного алгоритму знаходження НСД. - Обчислення НСК та його застосування для розв'язання практичних задач. - Реалізація решет Ератосфена, Сундарами та Аткинса. - Реалізація швидкого піднесення до степеня. - Використання швидкого піднесення до степеня для обчислення чисел Фібоначчі за модулем. - Обчислення оберненого числа за модулем і застосування малої теореми Ферма та теореми Ейлера. - Реалізація обчислення сполук за модулем та Китайської теореми про залишки. - Практичне застосування теорії чисел у криптографічних прикладах.
Тема 2. Комбінаторні алгоритми. Кількість годин: 19.		
<i>Учень / учениця:</i> - класифікує, структурує та узагальнює дані з обраної галузі, використовуючи відповідні методи та засоби [12 ІФО 1.2.3-1 П]; - обґрунтовано обирає формати файлів під час створення та збереження інформаційних продуктів з урахуванням	2.1. Поняття про комбінаторні об'єкти в програмуванні. Лексикографічний та металексикографічний порядок на множині комбінаторних об'єктів. Генерація комбінаторних об'єктів: перестановки, двійкові вектори, підмножини. Аналіз та реалізація алгоритмів. Двійкові вектори без двох одиниць поспіль.	Мінілекції та інтерактивні презентації з поясненням понять комбінаторики, порядків, алгоритмів генерації. Демонстрація коду (приклад на C++ для генерації перестановок, сполук, кодів Грея тощо). Практичне програмування (написання учнями власних реалізацій алгоритмів).

принципів кодування цифрових даних, що використовуються в обраній галузі [12 ІФО 1.2.2-4 П]; - розуміє та пояснює принципи кодування та декодування цифрових даних, що використовуються в обраній галузі, та враховує їх під час створення інформаційних продуктів [12 ІФО 1.2.2-1 П]; - пояснює принципи стиснення даних, обґрунтовано обирає метод стиснення, враховуючи можливість їхньої втрати [12 ІФО 1.2.2-2 П]; - пояснює поняття часової та ємнісної складності алгоритмів і способи їх оцінювання [12 ІФО 2.1.2-1 П]; оцінює ефективність алгоритмів розв'язування задач з обраної галузі, застосовуючи критерії часової та ємнісної складності [12 ІФО 2.1.2-2 П]; - пропонує критерії вибору, обґрунтовано обирає, налаштовує та ефективно використовує технологічні засоби для індивідуальної або групової діяльності та комунікації [12 ІФО 3.3.1-1 П].	2.2. Генерація комбінаторних об'єктів: сполуки, розбиття на доданки. Аналіз та реалізація алгоритмів. Коди Грея. 2.3. Генерація наступного та попереднього комбінаторних об'єктів: наступний та попередній двійковий вектор, наступна та попередня перестановка. 2.4. Генерація наступного та попереднього комбінаторних об'єктів: наступна та попередня сполука, наступне та попереднє розбиття на доданки. 2.5. Ідентифікація комбінаторного об'єкта за номером та номера за об'єктом: перестановка за номером та номер за перестановкою, сполука за номером та номер за сполукою, двійковий вектор за номером та номер за двійковим вектором, розбиття на доданки за номером та номер за розбиттям на доданки.	Групові проекти(аналіз та оптимізація алгоритмів, обговорення їх часової та ємнісної складності). Робота з інтерактивними середовищами розробки. Тренувальні завдання для самоконтролю (тести, вікторини, вправи на відповідність). Обговорення проблемних питань(чому один алгоритм ефективніший за інший). Аналіз результатів учнівських робіт і зворотний зв'язок. Домашні завдання на вибір(індивідуальні реалізації алгоритмів, підготовка мініпрезентацій, створення візуалізацій). <i>Список орієнтовних практичних завдань.</i> - Реалізувати програму для генерації перестановок у лексикографічному порядку. - Побудувати всі двійкові вектори довжини n, що не містять двох одиниць поспіль. - Реалізувати генерацію всіх k-елементних сполук множини. - Створити програму для генерації розбиттів числа на доданки. - Написати алгоритм генерації кодів Грея довжини n. - Створити програму для знаходження наступного двійкового вектора за умовою. - Реалізувати програму, яка за номером будує перестановку. - Задано перестановку – знайти її номер у лексикографічному порядку. - Задано k-сполуку – визначити її порядковий номер.
---	---	--

Тема 3. Дослідження алгоритмів. Проєкти.
Кількість годин: 19.

<i>Учень / учениця:</i> - пояснює та наводить аргументи доцільності та ефективності використання цифрових пристроїв / інформаційних технологій в обраній галузі для вирішення життєвих, зокрема професійних, задач [12 ІФО 1.1.1-2 П]; - оцінює можливості, функціонал, обмеження, переваги використання інформаційних технологій для вирішення проблем обраної галузі [12 ІФО 1.1.1-1 П]; - розробляє критерії оцінювання переваг та негативних впливів комп'ютерних інновацій на особистий розвиток, суспільство та довкілля, аргументує висновки [12 ІФО 1.1.3-2 П]; - описує суспільні проблеми, пов'язані з цифровим розривом та нерівним доступом до обчислювальних ресурсів [12 ІФО 1.1.3-3 П]; - аргументовано обирає та складає алгоритми для розв'язування задач з обраної галузі [12 ІФО 2.1.1-3 П]; - пояснює поняття універсального дизайну інформаційних продуктів і демонструє його реалізацію в обраній галузі [12 ІФО 4.2.2-1 П]; - активно шукає і пропонує конструктивні рішення для розвитку та перевірки ідей, удосконалення програмного проєкту на основі власних оцінок, відгуків	3.1. Методи аналізу, дослідження та порівняння алгоритмів. Автоматична генерація тестових даних. Презентація результатів дослідження у вигляді таблиць та графіків. Оформлення результатів дослідження. 3.2. Ідея проєкту: аналіз, дослідження та порівняння алгоритмів сортування: сортування «бульбашкою», сортування вставками, сортування вибором, швидке сортування, сортування злиттям. 3.3. Ідея проєкту: аналіз, дослідження та порівняння алгоритмів перевірки числа на простоту та алгоритмів факторизації. Ймовірнісні алгоритми перевірки числа на простоту. 3.4. Ідея проєкту: аналіз, дослідження та порівняння алгоритмів пошуку найбільшого спільного дільника. 3.5. Ідея проєкту: аналіз, дослідження та порівняння алгоритмів пошуку найкоротших шляхів у графі. 3.6. Ідея проєкту: аналіз, дослідження та порівняння простих структур даних.	Прослуховування інтерактивних мінілекцій про методи аналізу та порівняння алгоритмів. Обговорення прикладів застосування алгоритмів у життєвих та професійних сферах. Робота з навчальними матеріалами (таблиці, графіки, тестові дані). Виконання практичних завдань з аналізу та дослідження алгоритмів. Спільна робота в групах для порівняння результатів. Підготовка та проведення мініпрезентацій про дослідження алгоритмів. Обговорення етичних та суспільних проблем цифрового розриву. Самостійне створення критеріїв оцінювання ефективності алгоритмів. Участь у захисті проєктів (усно + у вигляді таблиць, графіків, презентацій). <i>Список орієнтовних практичних завдань.</i> - Реалізація та порівняння алгоритмів сортування (бульбашка, вставки, вибір, швидке сортування, злиття). - Аналіз і порівняння алгоритмів перевірки простоти числа та факторизації. - Дослідження алгоритмів пошуку найбільшого спільного дільника (Євкліда, бінарний алгоритм тощо). - Реалізація та порівняння алгоритмів пошуку найкоротшого шляху (Дейкстри, Беллмана–Форда, Флойда–Воршелла). - Аналіз часу виконання і використання пам'яті
---	---	---

користувачів і зміни умов його реалізації [12 ІФО 2.2.2-4 П]; - розпізнає різновиди прав інтелектуальної власності на інформаційні продукти та дотримується їх [12 ІФО 4.3.1-2 П].		алгоритмів. - Побудова таблиць та графіків результатів вимірювань. - Оформлення звіту про дослідження (структурований текст, таблиці, графіки, висновки).
--	--	--

12 клас. I семестр.
Кількість годин на тиждень: 3.
Кількість годин на семестр: 48.

Очікувані результати навчання	Пропонований зміст навчального предмета	Види навчальної діяльності
Тема 1. Динамічне програмування. Кількість годин: 16.		
<i>Учень / учениця:</i> - розпізнає та класифікує інформаційні системи різних типів (соціальні, біологічні, технічні тощо) та пояснює взаємозв'язки між ними [12 ІФО 1.3.2-1 П]; - аналізує необхідність оновлення компонентів інформаційної системи [12 ІФО 3.2.2-3 П]; - описує власне бачення світу як інформаційної системи [12 ІФО 1.3.2-2 П]; - обирає різні стратегії вирішення проблем (індуктивні, дедуктивні, евристичні, алгоритмічні) [12 ІФО 2.1.1-2 П]; - продукує нові ідеї щодо створення, удосконалення та застосування інформаційних продуктів, використовує ідеї інших і доопрацьовує їх [12 ІФО 2.4.3-1 П]; - аналізує наявні розробки та шукає інноваційні рішення під час створення інформаційних продуктів [12 ІФО 2.4.3-2 П]; - самостійно або в співпраці налагоджує програмні проекти [12 ІФО 2.2.2-3 П];	1.1. Знайомство з методами вирішення оптимізаційних задач. Метод динамічного програмування: запам'ятовування результатів, принцип оптимальності, незалежні підзадачі. Приклади застосування динамічного програмування. Переваги динамічного програмування. Приклади застосування динамічного програмування. 1.2. Проста лінійна динаміка. Запам'ятовування результатів в одновимірному масиві. Числа Фібоначчі. Динамічне програмування та рекурсія. Лінійні обчислення або виклик за потребою. 1.3. Квадратична динаміка. Задача про оптимальний шлях у таблиці та її варіації. 1.4. Задача про пошук найдовшої спільної підпослідовності (<i>longest common subsequence</i>). Аналіз та реалізація алгоритмів. 1.5. Задача про пакування рюкзака (<i>Knapsack problem</i>). Варіації постановки задачі про пакування рюкзака Застосування задачі про пакування рюкзака в промисловій та економічній галузі. Обмеження застосування методу динамічного програмування для розв'язку задачі про пакування рюкзака. Аналіз та реалізація алгоритму. Наближені методи розв'язання.	Інтерактивних лекцій про можливості C++ та принципи динамічного програмування. Демонстрація написання фрагментів коду для розв'язку оптимізаційних задач (Фібоначчі, шлях у таблиці, LCS). Виконання індивідуальних практичних завдань у середовищі розробки (написання програм, тестування, налагодження). Групове обговорення проблемних питань і стратегій вирішення задач (індуктивні, дедуктивні, евристичні, алгоритмічні). Створення інтерактивних матеріалів (схеми, таблиці, блок-схеми, порівняльні таблиці алгоритмів). Розробка власних мініпроектів із використанням динамічного програмування (індивідуально чи в парах). Презентація результатів власних програмних розробок і їх застосування до практичних задач. Робота в групах над складними завданнями з поділом ролей (аналітик, програміст, тестувальник, презентатор). Аналіз прикладів із промисловості та економіки, де застосовується задача про

- планує і організовує власну та командну діяльність під час розроблення проектів з використанням інформаційних технологій [12 ІФО 2.5.1-2 П]; - описує способи регулювання власного інформаційного навантаження і застосовує їх (обмеження, тайм - менеджмент, відключення певних функцій програм, перемикання на інші види діяльності, фізичні вправи тощо) [12 ІФО 4.1.1-2 П].	Метод гілок і меж. 1.6. Розбір складних задач на застосування методу динамічного програмування. Динаміка по профілю та по зламаному профілю, приклади.	рюкзак. Порівняння динамічного програмування з іншими методами розв'язання задач (жадібні алгоритми, метод гілок і меж). Проведення діагностичних робіт (тести, листки самоконтролю, анкетування). Надання та отримання зворотного зв'язку (самооцінка, взаємооцінка, коментарі вчителя). Створення навчальних матеріалів для інших (мініпосібники, відеоінструкції, приклади коду). <i>Список орієнтовних практичних завдань.</i> - Реалізація чисел Фібоначчі з рекурсією та збереженням результатів. - Знаходження оптимального шляху в таблиці. - Знаходження найдовшої спільної підпоследовності. - Розв'язання задачі про рюкзак (різні варіації).
Тема 2. Алгоритми на графах. Кількість годин: 16.		
<i>Учень / учениця:</i> - аналізує інформаційні процеси, що здійснюються в інформаційних системах різних галузей [12 ІФО 1.2.1-1 П]; - діє незалежно, уміє долати власні упередження [12 ІФО 1.4.2-5 П]; - задовольняє власні інформаційні потреби, активно шукає нові рішення та ідеї, зокрема читаючи та аналізуючи текстові джерела [12 ІФО 1.2.1-2 П]; - пояснює особливості та загальні	2.1. Актуалізація знань про графи за 11-й клас. Способи подання та зберігання графів. Алгоритм пошуку в глибину: компоненти зв'язності графа, перевірка графа на дводольність. Діаметр та радіус графа. 2.2. Алгоритми пошуку найкоротших шляхів у графі. Реалізація алгоритму Дейкстри за допомогою двійкової купи, збалансованого дерева пошуку або черги з пріоритетами. Алгоритм Флойда-Уоршелла. 2.3. Ребра та цикли від'ємної ваги. Алгоритм	Мінілекції з використанням інтерактивних матеріалів (схеми графів, візуалізації роботи алгоритмів). Демонстрація написання фрагментів коду мовою C++ (DFS, BFS, Дейкстра, Флойд-Уоршелл, Беллман-Форд, Прим, Крускал). Обговорення та аналіз проблемних питань. Виконання групових практичних завдань (розподіл ролей: програміст, тестувальник, аналітик). Самостійне розроблення та налагодження

принципи подання даних у файлах різних форматів [12 ІФО 1.2.2-3 П]; - класифікує, структурує та узагальнює дані з обраної галузі, використовуючи відповідні методи та засоби [12 ІФО 1.2.3-1 П]; - окреслює алгоритмічну складову задач з обраної галузі [12 ІФО 2.1.1-1 П]; - бере активну участь у розробленні інформаційних продуктів у складі команди, надає належну оцінку ідеям, підтримку та допомогу за потреби [12 ІФО 2.5.1-3 П]; - komponує інформаційну систему під задачу та визначає ефективність її функціонування [12 ІФО 3.2.1-2 П].	Форда-Беллмана. Оцінка складності алгоритму. Аналіз та реалізація алгоритму. 2.4. Мінімальне кістякове дерево. Система множин, що не перетинаються. Алгоритми Прима та Крускала. Оцінка складності алгоритмів. Аналіз та реалізація алгоритмів. 2.5. Дводольні графи. Паросполучення в графах. Теорема Холла. Найбільше паросполучення у дводольних графах. Оцінка складності алгоритмів. Аналіз та реалізація алгоритмів. 2.6. Узагальнення теми «Алгоритми на графах». Приклади застосування теорії графів. Розв'язування компетентнісних завдань.	програм для обраних алгоритмів. Тренування навичок роботи в IDE (середовище C++). Проведення тестів і опитувань для перевірки розуміння матеріалу. Робота з текстовими та візуальними джерелами (статті, відео, документація). Розробка інформаційних продуктів (порівняльні таблиці алгоритмів, інфографіка про складності). Рефлексія та зворотний зв'язок (короткі щоденники навчання, взаємооцінювання в групах). <i>Список орієнтовних практичних завдань.</i> - Реалізація алгоритму DFS для пошуку компонент зв'язності графа. - Програма перевірки графа на дводольність. - Знаходження діаметра та радіуса графа (застосування BFS). - Реалізація алгоритму Дейкстри з використанням черги з пріоритетами. - Порівняння швидкодії Дейкстри та Флойда-Уоршелла. - Реалізація алгоритму Беллмана-Форда для графів з від'ємними ребрами. - Побудова мінімального кістякового дерева: алгоритм Прима vs алгоритм Крускала. - Реалізація алгоритму знаходження найбільшого паросполучення у дводольному графі. - Оцінка складності та тестування всіх реалізованих алгоритмів на графах різної щільності.
---	--	---

		- Компетентнісне завдання: застосування алгоритмів на графах у задачах із реального життя (наприклад, оптимізація транспортної мережі).
Тема 3. Рядкові величини. Кількість годин: 16.		
<i>Учень / учениця:</i> - класифікує, структурує та узагальнює дані з обраної галузі, використовуючи відповідні методи та засоби [12 ІФО 1.2.3-1 П]; - описує, застосовує та за потреби розробляє стратегії захисту або протидії маніпуляціям, зокрема емоційним, у цифровому світі [12 ІФО 1.4.2-4 П]; - створює тестові приклади для перевірки різних сценаріїв роботи програмного проекту [12 ІФО 2.2.2-2 П]; - обґрунтовано обирає формати файлів під час створення та збереження інформаційних продуктів з урахуванням принципів кодування цифрових даних, що використовуються в обраній галузі [12 ІФО 1.2.2-4 П]; - розуміє та пояснює принципи кодування та декодування цифрових даних, що використовуються в обраній галузі, та враховує їх під час створення інформаційних продуктів [12 ІФО 1.2.2-1 П]; - пропонує критерії оцінювання інформаційних продуктів та їх відповідності поставленій задачі [12 ІФО	3.1. Базові поняття: рядок, підрядок, шаблон, суфікс, префікс. Задача пошуку підрядка в рядку. Наївний алгоритм. Оцінка складності алгоритму та реалізація мовою програмування C++. 3.2. Префікс-функція. Обчислення префікс-функції для рядка. Застосування префікс-функції: алгоритм Кнута-Морріса-Пратта. Оцінка складності алгоритму. Аналіз та реалізація алгоритму. 3.3. Z-функція від рядка. Обчислення Z-функції за префікс-функцією. Обчислення префікс-функції за Z-функцією рядка. 3.4. Хеш-функція. Обчислення хеш-функції для рядка. Використання хеш-функції для пошуку підрядка в рядку. Алгоритми хешування у завданнях на рядки. Колізії. Рядки <i>Тхе-Морса</i>. 3.5. Паліндроми. Алгоритми для пошуку паліндромів: з кубічною асимптотикою, з квадратичною асимптотикою, алгоритм Манакера. 3.6. Суфіксний масив, суфіксне дерево, суфіксний автомат. Пошук набору зразків. Алгоритм Ахо-Корасик.	Мінілекції та презентації з прикладами коду C++ (теорія + візуалізація алгоритмів). Демонстрація написання програм викладачем із поступовим поясненням. Практичні завдання індивідуальні й групові (поетапне ускладнення). Обговорення проблемних ситуацій: складність алгоритмів, вибір оптимального методу пошуку. Створення тестових прикладів для перевірки алгоритмів. Робота в групах: колективний пошук помилок і оптимізація коду. Діагностичні тести та листки самоконтролю для перевірки знань. Самостійні проєкти: мініпрограми на вибір (учень/учениця сам формулює задачу й реалізує). <i>Список орієнтовних практичних завдань.</i> - Написати програму, що знаходить підрядок у рядку наївним алгоритмом. - Порахувати кількість входжень підрядка. - Оцінити кількість операцій при різних довжинах рядка (аналіз складності). - Реалізувати обчислення префікс-функції. - Виконати пошук підрядка за КМП. - Порівняти швидкість роботи КМП і наївного

2.4.4-1 П]; - пояснює принципи стиснення даних, обґрунтовано обирає метод стиснення, враховуючи можливість їхньої втрати [12 ІФО 1.2.2-2 П]; - гнучко реагує на зміни умов і вимог під час створення інформаційних продуктів [12 ІФО 2.5.1-4 П]; - враховує різні погляди, національні та культурні особливості для прийняття зважених рішень під час створення інформаційного продукту [12 ІФО 2.5.2-1 П].		алгоритму на тестових прикладах. - Реалізувати Z-функцію. - Перетворити префікс-функцію на Z-функцію та навпаки (на прикладах). - Виконати пошук підрядка за допомогою Z-функції. - Обчислити поліноміальний хеш для рядка. - Реалізувати алгоритм Рабіна–Карпа для пошуку підрядка. - Продемонструвати приклад колізії та пояснити, як її усунути. - Скласти рядок Туе–Морса довжини n. - Написати програму, що перевіряє, чи є рядок паліндромом. - Знайти всі паліндромні підрядки у рядку (кубічний і квадратичний метод). - Реалізувати алгоритм Манакера та порівняти час виконання. - Реалізувати суфіксний масив (побудова + приклад пошуку). - Написати програму для пошуку множини зразків у рядку (Ахо-Корасик).
--	--	---

12 клас. II семестр.
Кількість годин на тиждень: 3.
Кількість годин на семестр: 57.

Очікувані результати навчання	Пропонований зміст навчального предмета	Види навчальної діяльності
Тема 1. Обчислювальна геометрія. Кількість годин: 19.		
<i>Учень / учениця:</i> - обґрунтовує власну позицію щодо достовірності інформації та надійності джерел, застосовуючи стратегії критичного мислення [12 ІФО 1.4.1-1П]; - розпізнає маніпулятивні впливи, які здійснюються цифровими та іншими каналами передавання інформації [12 ІФО 1.4.2-3 П]; - випробовує нові ідеї, обґрунтовано ризикує під час розроблення програмних проектів [12 ІФО 2.2.1-5 П]; - пропонує і застосовує методи перевірки коректності роботи програмного проекту [12 ІФО 2.2.2-1 П]; - оцінює якість інформаційних продуктів на основі вироблених критеріїв [12 ІФО 2.4.4-2 П]; - описує способи регулювання власного інформаційного навантаження і застосовує їх (обмеження, тайм-менеджмент, відключення певних функцій програм, перемикання на інші види діяльності, фізичні вправи тощо) [12 ІФО 4.1.1-2 П]; - використовує та ефективно опрацьовує	1.1. Декартова система координат. Геометричні примітиви: точка, пряма, відрізок. Відстань між точками на площині та в просторі. Пошук перетину прямих, променів, відрізків. 1.2. Вектори. Скалярний та векторний добуток векторів. Використання тригонометричних та обернених тригонометричних функцій. 1.3. Орієнтована площа трикутника та багатокутника. Застосування орієнтованої площі. Геометричні перетворення. Перетворення координат. 1.4. Теорема Піка. Підрахунок кількості точок з цілими координатами всередині багатокутника. 1.5. Оупкла оболонка. Алгоритм Джарвіса. Аналіз алгоритмів. Реалізація алгоритмів мовою програмування C++. 1.6. Скануюча пряма. Точка, вкрита найбільшою кількістю відрізків. Площа об'єднання прямокутників та кіл. Обертання скануючої прямої.	Мінілекцій з поясненням ключових понять (вектори, орієнтована площа, скануюча пряма). Демонстрація написання фрагментів коду на C++ із покроковим поясненням алгоритмів. Виконання практичних завдань у середовищі розробки з подальшою перевіркою коректності. Групове розв'язання задач із поділом ролей (алгоритмізація, кодування, тестування, презентація). Організація дискусій і обговорень щодо достовірності використаних алгоритмів та джерел. Самооцінювання та взаємооцінювання результатів програмних проектів. Розробка індивідуальних навчальних матеріалів (шпаргалки, мінігайди). Виконання діагностичних робіт (тести, контрольні запитання, анкети). Аналіз помилок та отримання зворотного зв'язку. <i>Список орієнтовних практичних завдань.</i> - Обчислення відстані між точками на площині та в просторі.

ідеї інших членів команди під час розроблення програмного проекту з належними посиланнями [12 ІФО 2.5.2-2 П]; - знає норми законодавства, що регулюють правові відносини у сфері інтелектуальної власності та використання інформаційних технологій [12 ІФО 4.3.1-1 П].		- Реалізація функції пошуку перетину двох відрізків. - Обчислення скалярного та векторного добутку векторів, застосування для знаходження кута між ними. - Написання програми для знаходження орієнтованої площі трикутника/багатокутника. - Використання теореми Піка для обчислення кількості точок з цілими координатами всередині багатокутника. - Реалізація алгоритму побудови опуклої оболонки (алгоритм Джарвіса). - Застосування методу «скануючої прямої» для знаходження точки, вкритої найбільшою кількістю відрізків - Обчислення площі об'єднання кількох прямокутників. - Оптимізація та тестування реалізованих алгоритмів.
Тема 2. Введення в об'єктно-орієнтоване програмування, бібліотеки та фреймворки. Кількість годин: 19.		
<i>Учень / учениця:</i> - пояснює та враховує переваги та ризики використання конкретних сучасних інформаційних технологій в обраній галузі [12 ІФО 1.1.2-2 П]; - наводить приклади факторів, які впливають на розподіл і використання цифрових ресурсів у глобальному суспільстві [12 ІФО 1.1.3-1 П]; - аргументовано обирає методи і стратегії для розпізнавання маніпулятивних технік	2.1. Причини виникнення, переваги та парадигми об'єктно-орієнтованого підходу до програмування. Складові об'єктно-орієнтованої методології. Принцип інкапсуляції даних і дій. Наслідування. Поліморфізм. 2.2. Класи, об'єкти та методи. Специфікатори доступу <i>public</i> і <i>private</i>. Інкапсуляція. Функції доступу. Конструктори та деструктори. Класи і заголовкові файли. 2.3. Знайомство з мультимедійною бібліотекою <i>SFML</i>. Встановлення та налаштування бібліотеки	Інтерактивні лекції про об'єктно-орієнтоване програмування (ООП) та сучасні бібліотеки/фреймворки. Демонстрація написання фрагментів коду на C++ для створення класів, об'єктів, використання конструкторів/деструкторів, інкапсуляції та поліморфізму. Виконання практичних завдань у середовищі розробки з подальшою перевіркою та налагодженням коду. Групове розв'язання проєктних завдань з

[12 ІФО 1.4.2-2 П]; - прогнозує наслідки дій у цифровому середовищі, їх вплив на власну цифрову репутацію та цифрову репутацію інших осіб [12 ІФО 4.1.3-1 П]; - дотримується норм етичної поведінки під час цифрової комунікації [12 ІФО 4.2.1-3 П]; - пояснює поняття універсального дизайну інформаційних продуктів та демонструє його реалізацію в обраній галузі [12 ІФО 4.2.2-1 П]; - аргументовано обирає вид ліцензії програмних засобів та інформаційних продуктів для розв'язування задач із обраної галузі [12 ІФО 4.3.1-4 П]; - пояснює особливості реалізації та захисту прав людини в контексті інформаційних технологій [12 ІФО 4.3.1-6 П]; - розпізнає власний емоційний стан і емоційний стан членів команди, ураховує його вплив під час розроблення проєктів з використанням інформаційних технологій [12 ІФО 2.5.2-3 П]; - обстоює особисті права на інтелектуальну власність з дотриманням норм права та етики [12 ІФО 4.3.1-7 П].	<i>SFML</i>. Створення та компіляція проєкту <i>SFML</i>. Виведення простих геометричних фігур C++/<i>SFML</i>. Розробка ігор. 2.4. Знайомство з кросплатформним фреймворком <i>Qt5</i>. Функціонал і можливості <i>Qt5</i>. Встановлення та налаштування фреймворку. Створення та компіляція проєкту <i>Qt5</i>. 2.5.Бібліотека <i>OpenGL</i>. Функціонал і можливості бібліотеки <i>OpenGL</i>. Створення та компіляція проєкту з <i>OpenGL</i>.	розподілом ролей (кодинг, дизайн інтерфейсу, тестування, презентація). Обговорення та аналіз переваг і ризиків використання сучасних бібліотек та фреймворків. Надання та отримання конструктивного зворотного зв'язку щодо реалізації та дизайну проєктів. Самостійне створення навчальних матеріалів (конспекти, мінігайди) для закріплення знань. Проведення діагностичних робіт (тести, контрольні завдання, анкети) для оцінки засвоєння матеріалу. Проведення інструктажів з техніки безпеки при роботі з програмним забезпеченням та комп'ютерними лабораторіями. Виконання домашніх завдань та комплексних практичних проєктів. <i>Список орієнтовних практичних завдань.</i> - Створення класів та об'єктів у C++ з застосуванням інкапсуляції та специфікаторів доступу (<i>public/private</i>). - Реалізація конструкторів і деструкторів для управління ресурсами. - Розробка простих мультимедійних проєктів на <i>SFML</i> (виведення фігур, обробка подій). - Створення графічних інтерфейсів та проєктів на <i>Qt5</i> (кнопки, текстові поля, взаємодія з користувачем). - Реалізація 3D-графіки та об'єктів на <i>OpenGL</i> (створення сцени, рендеринг об'єктів). - Побудова інтерактивної програми з використанням ООП та бібліотек.
---	---	--

		- Аналіз і оптимізація коду, використання поліморфізму та наслідування для розширюваності проєктів.
Тема 3. Дослідження та візуалізація алгоритмів. Проєкти. Кількість годин: 19.		
<i>Учень / учениця:</i> - оцінює можливості, функціонал, обмеження, переваги використання інформаційних технологій для вирішення проблем обраної галузі [12 ІФО 1.1.1-1 П]; - оцінює роль і вплив маніпуляцій і пропаганди в особистому та суспільному житті, навчальній діяльності, цифровому світі [12 ІФО 1.4.2-1 П]; - враховує засади універсального дизайну та ергономічності під час створення інформаційних продуктів [12 ІФО 4.2.2-2 П]; - розпізнає різновиди прав інтелектуальної власності на інформаційні продукти та дотримується їх [12 ІФО 4.3.1-2 П]; - створює нові, розширює або змінює наявні програмні проєкти [12 ІФО 2.2.1-4 П]; - документує програмний проєкт, що складається з окремих компонентів [12 ІФО 2.3.1-5 П]; - оцінює роль і внесок кожного члена команди під час створення інформаційних продуктів [12 ІФО 2.5.1-1 П]; - пояснює призначення простих засобів захисту авторських прав і доречно	3.1. Методи аналізу, дослідження та порівняння алгоритмів. Презентація результатів дослідження у вигляді таблиць та графіків. Оформлення результатів дослідження. Візуалізація алгоритмів. 3.2. Ідея проєкту: аналіз, дослідження та візуалізація алгоритмів сортування: сортування «бульбашкою», сортування вставками, сортування вибором, швидке сортування, сортування злиттям. 3.3. Ідея проєкту: аналіз, дослідження та візуалізація алгоритмів на графах. 3.4. Ідея проєкту: аналіз, дослідження та візуалізація алгоритмів з рядковими величинами. 3.5. Ідея проєкту: аналіз, дослідження та візуалізація геометричних алгоритмів. 3.6. Ідея проєкту: аналіз, дослідження та візуалізація простих структур даних.	Розробка та презентація інтерактивних матеріалів, що пояснюють методи аналізу, дослідження та порівняння алгоритмів. Демонстрація прикладів реалізації алгоритмів (сортування, графи, рядкові алгоритми, геометричні алгоритми, прості структури даних). Обговорення проблемних питань та спільний аналіз результатів. Надання конструктивного зворотного зв'язку щодо реалізації проєктів. Формування практичних навичок презентації та захисту учнівських проєктів. Розробка і розподіл навчальних матеріалів для кращого засвоєння матеріалу та підготовки проєктів. Організація захисту проєктів із оцінкою вкладку кожного учасника. Робота в групах для розв'язання складних елементів проєктів. Презентація та захист власних та командних проєктів. <i>Список орієнтовних практичних завдань.</i> - Візуалізація алгоритмів сортування («бульбашка», вставки, вибір, швидке сортування, сортування злиттям) із демонстрацією кроків виконання.

застосовує їх [12 ІФО 4.3.1-3 П].		- Аналіз та візуалізація алгоритмів на графах (пошук найкоротшого шляху, обходи, мінімальні кістякові дерева). - Візуалізація алгоритмів роботи з рядковими величинами (пошук підрядка, префікс-функція, Z-функція, хешування, суфіксні структури). - Візуалізація геометричних алгоритмів (опукла оболонка, скануюча пряма, площа об'єднання фігур). - Демонстрація реалізації простих структур даних та їх алгоритмів (стек, черга, список, дерево). - Порівняння ефективності різних алгоритмів за швидкістю та використанням пам'яті, створення таблиць та графіків.
-----------------------------------	--	--

Прикінцева частина

Основними видами оцінювання результатів навчання учнів є формувальне, поточне та підсумкове (тематичне, семестрове, річне).

Формувальне оцінювання здійснюється на всіх етапах роботи учнів і передбачає систематичний зворотний зв'язок щодо процесу засвоєння знань і вироблення вмінь. Оцінюванню підлягають як процес, так і результат навчальної діяльності.

Поточне оцінювання передбачає відстеження рівня засвоєння навчального матеріалу на уроці та між уроками. Здійснюється через:

- усне опитування (індивідуальне, групове, фронтальне);
- письмові завдання (самостійні, контрольні, вправи з розв'язування задач з алгоритмізації, програмування, логіки);
- цифрові форми (онлайн-тести, робота з інтерактивними тренажерами, автоматизоване оцінювання коду в середовищах програмування);
- практичні роботи (робота з алгоритмами, створення програмних проєктів, моделювання даних, застосування інструментів цифрової обробки інформації);

Підсумкове оцінювання проводиться у формі контрольних і тематичних робіт, а також через виконання учнівських проєктів.

Оцінюванню підлягають:

- здатність учня/учениці застосовувати теоретичні знання на практиці;
- уміння працювати з цифровими інструментами та програмним забезпеченням;
- навички алгоритмічного мислення та програмування;
- вміння аналізувати, відбирати та коректно використовувати інформацію;
- здатність до роботи з даними (структурування, візуалізація, пошук закономірностей);
- культура цифрової комунікації, безпечна поведінка в інформаційному середовищі;
- вміння працювати індивідуально та в команді над спільними завданнями.

Особливістю цієї Програми є можливість виконання учнівського проєкту як альтернативи традиційній контрольній роботі чи тестуванню. Такий формат дозволяє оцінювати не лише рівень засвоєння знань, а й уміння застосовувати їх у практичній діяльності, здатність до самостійного пошуку та аналізу інформації, творчість і командну взаємодію.

Орієнтовні критерії оцінювання учнівських проєктів.

1. Актуальність і постановка проблеми (0–2 бали):

- чітке визначення теми, завдань і цілей проєкту;
- обґрунтування вибору інструментів (мови програмування, середовища, сервісів).

2. Теоретична база та використані знання (0–2 бали):

- застосування вивчених понять (алгоритми, структури даних, графіка, тощо);
- використання базових і розширених знань інформатики.

3. Технічна реалізація (0–4 бали):

- якість коду (структурованість, коментарі, оптимальність алгоритмів);
- виконання вимог до інтерфейсу продукту (якщо він передбачений) (зручність, логічність);
- функціональність і відсутність критичних помилок.

4. Творчість і самостійність (0–2 бали):

- оригінальність ідеї реалізації мети проєкту;
- ступінь самостійного виконання роботи;
- дотримання принципів академічної доброчесності.

5. Презентація та захист проєкту (0–2 бали):

- здатність учня/учениці пояснити принципи роботи та обґрунтувати обрані рішення;
- якість візуалізації результатів (схеми, графіки, скріншоти, демонстрація роботи програми).

Максимальна кількість балів – 12.

Список пропонованої літератури

1. Веклич Р. А., Карнаух Т. О., Ставровський А. Б. Вступ до програмування мовою C++. Структури даних. — Київ : ВПЦ «Київський університет», 2018.
2. Рудий Т. В., Паранчук Я. С., Сенік В. В. Алгоритмізація та програмування. Частина 1. Структурне програмування : навчальний посібник. — Львів : Львівський державний університет внутрішніх справ, 2023. — 240 с.
3. Бублик В. В. Об'єктно-орієнтоване програмування : підручник. — Київ : ІТ-книга, 2015. — 624 с.
4. Грицюк Ю. І., Рак Т. Є. Програмування мовою C++ : навчальний посібник. — Львів : Вид-во Львівського ДУ БЖД, 2011. — 292 с.
5. Ментинський С. М., Пелех Я. М. Основи програмування на C++ : навчальний посібник. — Львів : Галицька Видавнича Спілка, 2021. — 256 с.
6. Ришковець Ю. В., Висоцька В. А. Алгоритмізація та програмування. Частина 1 : навчальний посібник. — Львів : Видавництво «Новий Світ-2000», 2021. — 337 с.
7. Ришковець Ю. В., Висоцька В. А. Алгоритмізація та програмування. Частина 2 : навчальний посібник. — Львів : Видавництво «Новий Світ-2000», 2021. — 315 с.
8. Трофименко О. Г., Прокоп Ю. В., Задерейко О. В. Алгоритмізація та програмування : навчально-методичний посібник. — Одеса : Фенікс, 2020. — 310 с. — Режим доступу: <http://dspace.onua.edu.ua/handle/11300/12345>.
9. Ярошко С. А. Методи розробки алгоритмів. Програмування мовою C++ : навчальний посібник. — Львів : ЛНУ імені Івана Франка, 2022. — 248 с.
10. C++ Reference [Електронний ресурс] / en.cppreference.com. — Режим доступу: <https://en.cppreference.com>. — Дата звернення: 22.08.2025.
11. C++ Language Tutorial [Електронний ресурс] / cplusplus.com. — Режим доступу: <https://cplusplus.com/doc/tutorial/>. — Дата звернення: 22.08.2025.
12. Online C++ Compiler [Електронний ресурс] / onlinegdb.com. — Режим доступу: https://www.onlinegdb.com/online_cplusplus_compiler. — Дата звернення: 22.08.2025.
13. Ideone [Електронний ресурс] / ideone.com. — Режим доступу: <https://ideone.com>. — Дата звернення: 22.08.2025.
14. ISO C++ Standard [Електронний ресурс] / isocpp.org. — Режим доступу: <https://isocpp.org/std/the-standard>. — Дата звернення: 22.08.2025.
15. Кублій А. О. Алгоритмізація та програмування. Практикум. — 2019. — Режим доступу: https://ela.kpi.ua/bitstream/123456789/28216/1/Alhorytmizatsiya-ta-prohramuvannia-Praktykum_2019Kublii.pdf.
16. Бандоріна Ю. О. Навчальний посібник з програмування. — Дніпро, 2019. — Режим доступу: <http://eadnurt.diit.edu.ua/bitstream/123456789/15729/1/Bandorina.pdf>.