Задачі

**Наслідування та поліморфізм**

1. Створити клас для представлення довільної матриці TMatrix, визначити в ньому метод форматованого виведення матриці. На основі нього створити класи нащадки, які представляють квадратну матрицю та матрицю стовпець: для квадратної матриці перевантажити оператор + для знаходження суми матриць, а для матриці-стовпця перевантажити оператор \* для знаходження добутку матриці на число. Згенерувати випадковим чином дві квадратні матриці, отримавши від користувача дані про розмірність, транспонувати першу з них, та знайти їх суму; згенерувати випадковим чином матрицю стовпець та помножити її на задане число.
2. Створити інтерфейс для представлення функціоналу двовимірної фігури Figure та оголосити в ньому методи для знаходження периметру і площі, і метод для форматованого виведення інформації про об’єкт. Імплементувати цей інтерфейс у класах для представлення трикутника і прямокутника, додати у ці класи атрибути довжин сторін, як мінімум один конструктор, гетери, сетери. У класі трикутника ввести у конструкторі перевірку на можливість існування трикутника, що створюється, а у класі прямокутника визначити метод для обрахунку довжини діагоналі. Створити прямокутник, знайти його периметр, площу, довжину діагоналі і створити трикутник, у якого сторони будуть рівними трьом показникам прямокутника. Знайти площу створеного трикутника, якщо він існує, інакше вивести форматоване повідомлення про неможливість існування.
3. Створити структуру Product, яка містить назву та ціну продукту, та клас Person, який містить інформацію про ім’я людини, її вік та стать. На основі класу людини створити класи, для представлення покупця і продавця. У класі покупця визначити кількість грошей у гаманці, список покупок, декілька перевантажених конструкторів ініціалізації, методи для доповнення або прибирання продуктів зі списку та перевантажити оператор - для зняття грошей з рахунку. У класі продавця визначити список наявних у продажі товарів, методи для продажі предметів та поповнення асортименту. Створити покупця та продавця за введеною інформацією, знайти максимальну кількість товарів, яку може купити покупець.
4. Створити клас для представлення банківського рахунку TBankAccount, визначити у ньому атрибути ім’я власника рахунку, кількості грошей та методи для списання та нарахування коштів. Створити два класи нащадки для представлення гривневих та доларових рахунків, визначити у кожному класі як мінімум один конструктор, гетери, а також методи для конвертації з однієї валюти у іншу. Створити гривневий та доларовий рахунок за введеними значеннями, після чого конвертувати на одному з рахунків частину валюти і перевести на інший рахунок. Після цього визначити на якому з рахунків більше грошей (з урахуванням валюти).
5. Створити інтерфейс для представлення функціоналу двовимірної фігури Figure та оголосити у ньому метод знаходження площі та периметру фігури і метод для форматованого виведення інформації про об’єкт. Імплементувати цей інтерфейс у класах для представлення кола (замість периметра довжину кола) і квадрата, додати у ці як мінімум один конструктор, гетери, сетери. Для класа кола визначити атрибут радіусу, для квадрата визначити атрибут довжини сторони та перевантажити оператор / для перевірки можливості вписати іншого об’єкта коло у квадрат. Створити за введеними даними об’єкт квадрата та об’єкт кола, перевірити, чи можливо вписати коло в квадрат, якщо так, то вивести інформацію про коло, інакше – вивести інформацію про квадрат.
6. Створити абстрактний клас для представлення клієнта банку TClient з атрибутом прізвища та з методами, що дозволяють вивести на екран інформацію про клієнтів банку, а також визначити відповідність клієнта критерію пошуку. Створити похідні класи вкладника з такими додатковими атрибутами, як дата відкриття вкладу, розмір вкладу, відсоток вкладу та кредитора з атрибутами дати видачі кредиту, розмірe кредиту, залишку боргу. Реалізувати у класах-нащадках як мінімум по одному конструктору, гетери, сетери. Створити масив з заданою кількістю клієнтів, вивести повну інформацію про всіх клієнтів на екран і організувати пошук клієнтів, котрі почали співпрацювати з банком в задану дату.
7. Створити клас для представлення функції TFunction, що має методи для знаходження значення функції у заданій точці та форматованого виводу інформації про функцію. Створити похідні класи лінійної та квадратичної функції, задати в них атрибути відповідних коефіцієнтів, у класі квадратичної функції перевантажити оператор ~ для знаходження похідної від квадратичної функції. Створити за введеними даними квадратичну функцію, знайти від неї похідну, і зробити з неї об’єкт лінійної функції, після чого знайти значення лінійної функції у заданій точці.
8. Створити клас для представлення довільного трикутника TTriangle з методом для знаходження периметра і методом для виведення інформації про трикутник. На основі цього класу створити класи-нащадки для представлення прямокутного та рівностороннього трикутника. У класі прямокутного трикутника задати атрибути двох катетів та метод для знаходження гіпотенузи. У класі рівностороннього трикутника задати атрибут довжини сторони та метод знаходження довжини висоти трикутника. У кожному класі створити відповідні конструктори, гетери та сетери. Створити за введеними даними прямокутний трикутник, знайти його гіпотенузу та за отриманим значенням створити рівносторонній трикутник, знайти його висоту, і створити новий прямокутний трикутник, у якого довжини катетів дорівнюють довжині гіпотенузи і висоти, знайти периметр отриманого трикутника.
9. Створити клас для представлення дробу TFraction, задати у ньому атрибут цілої частини та метод для форматованого виводу дробу на екран. На основі цього класу створити класи для представлення десяткових та простих дробів, у класі десяткового дробу задати додатковий атрибут десяткової частини та перевантажити оператор \* для збільшення дробу у n раз, а у класі простого дробу задати додаткові атрибути чисельника і знаменника, перевантажити оператор + для збільшення дробу на n одиниць. У кожному класі створити відповідні конструктори, гетери, сетери. За введеними даними створити десятковий та простий дріб, збільшити десятковий дріб у задане число разів, збільшити простий дріб на інше задане число та перевірити у якого дробу ціла частина більше.
10. Створити абстрактний клас для представлення транспортного засобу TVehicle, у якому оголосити атрибути максимальної швидкості і наявної кількості пасажирів та метод для виведення інформації про транспортний засіб. Створити класи нащадки для представлення автомобіля і автобусу. У класі автомобіля перевантажити оператори + та - для додавання та прибирання пасажирів, причому з кожним доданим пасажиром швидкість авто падає на 3 км/г, і максимальна кількість пасажирів автомобіля складає 7 людей, також додати конструктор і гетери. У класі автобусу перевантажити оператори + та - для додавання та прибирання пасажирів, причому з кожним доданим пасажиром швидкість автобусу падає на 2 км/г, і максимальна кількість пасажирів автобусу складає 25 людей , також додати конструктор і гетери. Створити за введеними значеннями об’єкти цих двох класів, додати задану кількість пасажирів у автомобіль та прибрати задану кількість пасажирів з автобусу, після чого визначити, швидкість якого з об’єктів більше.

**Базові структури даних**

1. Створити структуру даних зв’язний список для зберігання числових даних. У класі реалізувати метод пошуку елементів, сортування, знаходження мінімуму і максимуму та виведення у консоль. Користувачем на вхід подаються оцінки студентів, що можуть бути від 0 до 10, які записуються у список. Створити новий список, у який додати оцінки студентів, які вище за 4, та відсортувати їх у зростаючому порядку. Знайти у новому списку максимальну та мінімальну оцінку.
2. Створити структуру даних черга для зберігання рядків, яка підтримує операції додавання та видалення елементів з черги і метод виведення черги у консоль. Створити чергу з імен людей, заданих користувачем. Відомо, що одна людина обслуговується за 3 хвилини, після чого покидає чергу. Визначити, хто залишиться у черзі після заданої користувачем кількості хвилин.
3. Створити структуру даних стек для зберігання символьних даних, що підтримує операції додавання елемента, видалення елемента з отриманням, перевірки на заповненість та виведення стеку у консоль. Користувачем вводиться рядок, який розбивається на символи, та вводиться інший символ, після чого елементи по черзі додаються у стек, причому елементи, що співпадають з введеним символом заносяться у стек двічі. Якщо стек переповнюється, то виводиться відповідне повідомлення і створюється новий стек.
4. Створити структуру даних множина для зберігання унікальних числових значень, що підтримує операції додавання, видалення та перевірки наявності елементу та метод виведення множини у консоль. Створити множину з заданих користувачем елементів, після чого почати приймати на вхід від користувача елементи до тих пір, поки елемент не буде повторюватися три рази, причому кожен раз, коли елемент повторюється, видаляти його з множини.
5. Створити структуру даних двозв’язний список для зберігання символьних даних, яка підтримує операції додавання та видалення елементу зі списку, сортування, та виведення списку у консоль. Користувачем на вхід подається рядок, який після цього ділиться на символи і вноситься у список. Відсортувати список та сформувати новий рядок з непарних елементів, вивести його у консоль.
6. Створити структуру даних стек для зберігання рядків, що підтримує операції додавання елемента, видалення елемента з отриманням та виведення стеку у консоль. Користувачем на вхід подаються рядки, серед яких можуть бути однакові. Перевірити стек на те, чи знаходяться у ньому поруч два однакових рядка, і якщо так, то видалити обидва. Вивести отриманий стек.
7. Створити структуру даних черга для зберігання числових значень, яка підтримує операції додавання та видалення елементів з черги і метод виведення черги у консоль. Користувачем задаються числа, що являють собою оцінки студентів за семестр. Переглянути чергу і знайти кількість студентів з заданою оцінкою. Видалити всіх студентів з заданою оцінкою і вивести перетворену чергу.
8. Створити структуру даних множина для зберігання унікальних символьних значень, що підтримує операції додавання, видалення та перевірки наявності елементу та метод виведення множини у консоль. Користувачем на вхід подається рядок, який після цього ділиться на символи і вноситься у множину. Знайти усі елементи, що повторюються, та створити нову множину з ними, та перевірити нову множину на наявність заданого елементу.
9. Створити структуру даних однозв’язний список для зберігання числових значень, що підтримує пошук елементів, знаходження мінімуму і максимуму, сортування, та виведення у консоль. Користувачем на вхід подаються числа, що являють собою температури повітря у різні дні. Знайти середнє арифметичне між максимальною та мінімальною температурою та порівняти її з медіаною температури у списку.
10. Створити структуру даних двостороння черга для зберігання рядків, яка підтримує операції додавання та видалення елементів з черги і метод виведення черги у консоль. Створити чергу з імен людей, заданих користувачем. Відомо, що одна людина обслуговується 2 хвилини, після чого покидає чергу, а людині яка стоїть в кінці очікування набридає через 5 хвилин, після чого вона виходить з черги. Визначити, хто залишиться у черзі після заданої користувачем кількості хвилин.

**Базові структури даних + класи**

1. Створити клас для представлення книги з атрибутами назви, автору та рік видання. Створити структуру даних список, що містить об’єкти класу книга та підтримує операції пошуку за різними атрибутами і виведення списку у консоль. Ввести задану кількість книг, внести їх у список і знайти у ньому книги, у яких збігається або виключно рік видання, або виключно автор. Створити інший список та внести у нього знайдені книги.
2. Створити клас для представлення автомобіля з атрибутами марка, року виготовлення та максимальної швидкості. Створити структуру даних список, що містить об’єкти класу автомобіль та підтримує операції пошуку, знаходження максимального і мінімального елемента за різними атрибутами і виведення списку у консоль. Ввести задану кількість автомобілів, внести їх у список і знайти автомобіль з максимальною швидкістю і автомобіль з найпізнішою датою виходу, порівняти їх між собою. Створити новий список з усіма автомобілями, що вийшли в той же рік, що й автомобіль з найбільшою швидкістю.
3. Створити клас для представлення трикутника з атрибутами довжинсторін та методами для знаходження периметру і для формування строкового представлення трикутника, як (a, b, c). Створити структуру даних список, що містить об’єкти класу трикутник та підтримує операції пошуку за різними атрибутами, та сортування за периметром, і виведення списку у консоль. Прийняти на вхід дані про задану кількість трикутників, внести створені об’єкти у список та відсортувати трикутники за перметром. Знайти трикутник, у якого периметр дорівнює найбільшій стороні трикутника з максимальним периметром, якщо такого не існує, то вивести відповідне повідомлення.
4. Створити клас для представлення прямокутника з атрибутами двох сторін та методами для знаходження площі і для формування строкового представлення трикутника, як (a, b, c). Створити структуру даних список, що містить об’єкти класу прямокутник та підтримує операції пошуку, знаходження максимального і мінімального елемента за різними атрибутами елементів, і виведення списку у консоль. Прийняти на вхід дані про задану кількість прямокутників, внести створені об’єкти у список та відсортувати прямокутники за площею. Знайти прямокутник, у якого площа дорівнює сумі двох сторін найбільшого прямокутника, якщо такого не існує, то вивести відповідне повідомлення.
5. Створити структуру для представлення продукту з атрибутами назви і рік виготовлення. Створити структуру даних список, що містить об’єкти структури продукт та підтримує операції пошуку та сортування елементів за різними атрибутами, і виведення списку у консоль. Прийняти на вхід дані про задану кількість продуктів, внести створені об’єкти у список та знайти у ньому продукти, у яких назва співпадає з заданим користувачем рядком. Внести отримані продукти у новий список та відсортувати за роками виготовлення.
6. Створити структуру для представлення дому з атрибутами міста, назви вулиці та номеру будинку. Створити структуру даних множина, що містить унікальні об’єкти структури дім та підтримує операції додавання, видалення та перевірки наявності елементу та метод виведення множини у консоль. Прийняти на вхід дані про задану кількість домів, внести створені об’єкти у множину, але, якщо елементи повторюються, то вносити об’єкти у іншу множину і видаляти з першої. Після цього попросити користувача змінити інформацію про доми, що повторюються, і внести змінені об’єкти домів у першу множину.
7. Створити структуру для представлення студента з атрибутами прізвища, ім’я та середнього бала. Створити структуру даних список, що містить об’єкти структури студент та підтримує операції пошуку та сортування елементів за різними атрибутами, і виведення списку у консоль. Прийняти на вхід дані про задану кількість студентів, внести створені об’єкти у список та знайти у ньому студентів, у яких середній бал вищий ніж 85. Утворити новий список із знайдених студентів та відсортувати його за прізвищем.
8. Створити структуру для представлення фрукта з атрибутами назви, кольору та кількості калорій. Створити структуру даних список, що містить об’єкти структури фрукт та підтримує операції додавання елемента, знаходження максимального і мінімального елемента за атрибутом калорійності та виведення списку у консоль. Прийняти на вхід дані про задану кількість фруктів, внести створені об’єкти у список та знайти у ньому фрукти з максимальною і мінімальною кількістю калорій.
9. Створити структуру для представлення абітурієнта з атрибутами прізвища, ім’я та рейтингового балу. Створити структуру даних список, що містить об’єкти структури студент та підтримує операції сортування елементів за різними атрибутами, видалення елементів і виведення списку у консоль. Прийняти на вхід дані про задану кількість абітурієнтів, внести створені об’єкти у список та видалити з нього абітурієнтів з рейтинговим балом менше за 50. Утворити новий список із студентів, що залишилися та відсортувати його за прізвищем.
10. Створити клас для представлення лінійної функції з атрибутами коефіцієнтів, та методами для здвигу функції по осі ОY на задану кількість одиниць. Створити структуру даних множина, що містить унікальні об’єкти структури класу функції та підтримує операції перевірки наявності елементу та метод виведення множини у консоль. Прийняти на вхід дані про задану кількість функцій, внести створені об’єкти у множину, та знайти об’єкт у списку за введеними користувачем коефіцієнтами. Після цього зсунути функцію на задану кількість одиниць.

**Дерева**

1. Користувачем задається рядок, який розбивається на символи, і з цих символів формується бінарне дерево. Сформувати нові рядки через прямий та симетричний обхід дерева та вивести їх.
2. Користувач вводить масив з n чисел, потрібно сформувати з них бінарне дерево. Знайти суму лівого та правого піддерева та порівняти їх. Вставити декілька заданих користувачем нових елементів у дерево
3. Користувачем вводиться математичний вираз, який потрібно розбити на бінарне дерево. Реалізувати префіксний обхід дерева, вивести вираз у консоль та обчислити його.
4. Користувач вводить двозначне число n, потрібно сформувати бінарне дерево з усіх дільників цього числа. За допомогою зворотного обходу визначити, чи є інше задане користувачем число дільником числа n та вивести всі відвідані вершини.
5. Згенерувати масив з n випадкових елементів з інтервалу [-50; 50]. Сформувати бінарне дерево та за допомогою симетричного обходу визначити кількість від’ємних чисел. Вивести всі відвідані вершини.
6. Користувачем вводиться речення, з слів якого потрібно сформувати бінарне дерево. Знайти на якому рівні знаходиться задане користувачем слово. Вставити декілька слів та за допомогою прямого обходу сформувати нове речення.
7. Користувач вводить масив з n чисел, потрібно сформувати з них бінарне дерево пошуку. Вставити у дерево декілька елементів так, щоб зберіглась властивість дерева. Знайти у дереві кількість входжень заданого елемента.
8. Користувач вводить арифметичний вираз, який потрібно розбити на бінарне дерево. Реалізувати постфіксний обхід дерева, вивести вираз у консоль та обчислити його.
9. Користувач вводить значення n та m, які будуть підставлятися у формулу . Розбити вираз на дерево і реалізувати префіксний обхід дерева, вивести результат обчислення у консоль.
10. Користувач вводить масив чисел, з якого потрібно сформувати бінарне дерево. Пройтись по всім шляхам дерева з кореня до листків і знайти шлях, на якому сума елементів буде максимальною. Вивести знайдену суму і шлях.
11. Програмою випадково генеруються n чисел з інтервалу [-10; 10], з яких треба сформувати бінарне дерево. Перевірити, чи є це дерево бінарним деревом пошуку, і, якщо ні, то відсортувати масив і побудувати дерево пошуку.
12. Користувач вводить значення a та b, які будуть підставлятися у формулу . Розбити вираз на дерево і реалізувати префіксний обхід дерева, вивести результат обчислення у консоль.
13. Користувач вводить масив з n чисел, потрібно сформувати з них бінарне дерево. За допомогою прямого обходу знайти шлях між двома заданими вершинами дерева. Вивести шлях у консоль.
14. Користувачем вводиться речення, з слів якого потрібно сформувати бінарне дерево. За допомогою зворотного обходу від найкоротшого до найдовшого вузла сформувати нове речення і вивести його у консоль.
15. Користувачем задається рядок, який розбивається на символи, і з цих символів формується бінарне дерево. . Пройтись по всім шляхам дерева з кореня до листків, зберегти всі рядки, які можливо утворити таким чином і вивести їх.

**Хеш-таблиці**

1. Створити хеш-таблицю з закритою адресацією і хеш-функцією ділення, яка підтримує процедури пошуку, вставки за ключом і видалення за значенням. Прийняти на вхід декілька значень від користувача, в яких будуть значення, які спровокують колізію. Отримати список значень у таблиці за заданим ключем і видалити їх.
2. Створити хеш-таблицю з відкритою адресацією, використавши лінійний пошук для вирішення колізій. Хеш-таблиця повинна підтримувати процедури пошуку та вставки за ключом. Прийняти на вхід декілька значень від користувача, в яких будуть значення, які спровокують колізію. Знайти за яким ключем знаходиться введене значення.
3. Створити хеш-таблицю з відкритою адресацією, використавши квадратичний пошук для вирішення колізій. Хеш-таблиця повинна підтримувати процедури пошуку та вставки за ключом. Прийняти на вхід декілька значень від користувача, в яких будуть значення, які спровокують колізію. Отримати список значень у таблиці за заданим ключем.
4. Створити хеш-таблицю з відкритою адресацією, використавши лінійний пошук для вирішення колізій. Хеш-таблиця повинна підтримувати процедури пошуку та вставки за ключом. Прийняти на вхід декілька значень від користувача, в яких будуть значення, які спровокують колізію. Отримати список значень у таблиці за заданим ключем
5. Створити хеш-таблицю з закритою адресацією і хеш-функцією множення, яка підтримує процедури пошуку, вставки за. Прийняти на вхід декілька значень від користувача, в яких будуть значення, які спровокують колізію. Отримати список значень у таблиці за заданим ключем.