

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФРАСТРУКТУРИ ТА ТЕХНОЛОГІЙ
ІНСТИТУТ УПРАВЛІННЯ, ТЕХНОЛОГІЙ ТА ПРАВА
ФАКУЛЬТЕТ УПРАВЛІННЯ І ТЕХНОЛОГІЙ

Кафедра інформаційних технологій

Лабораторна робота №5

з дисципліни «Фундаментальні комп'ютерні алгоритми»

з теми: «ДОСЛІДЖЕННЯ АЛГОРИТМІВ ПОШУКУ»

Варіант 13

Виконав:

Студент групи

ІПЗд-23121 маг.

Петренко Д.М.

Перевірів:

Доцент кафедри ІТ

Ткаченко О.А.

Київ – 2024

Лабораторна робота №5

Мета: дослідити алгоритми пошуку та набуті практичних навичок з пошуку елементів у лінійних та нелінійних структурах даних.

ЗАВДАННЯ:

Виконати такі дії:

- описати студента згідно з варіантом завдання (Прізвище, ім'я, день, місяць і рік народження, хобі);
- описати метод, який виконує пошук в одновимірному масиві студентів за заданим алгоритмом згідно з варіантом завдання (Послідовний);
- створити та ініціювати екземпляр лінійної структури даних (одновимірний масив студентів розміром не менш 20 елементів) з урахуванням правила формування (Невпорядкований). Правило формування структури даних застосовується при додаванні нового елемента під час ініціювання одновимірного масиву. Якщо правило формування - «Упорядкований ...», то це означає, що при введенні чергового -довільного випадкового елемента йому треба знайти відповідне місце у масиві так, щоб в результаті формування масив був упорядкований.
- виконати завдання (Видалити студентів, народжених улітку, що захоплюються туризмом);
- вивести вміст одновимірного масиву в разі його зміни (до і після видалення) або після виконання всього завдання.

ХІД РОБОТИ:

Створюємо програму на мові програмування Java, яка виконуватиме поставлене завдання. Розглянемо весь код:

В класі Main присутній Student. Розглянемо весь блок коду окремо

```

import java.util.ArrayList;

public class Main {

    public static class Student {
        private static int nextId = 1; // Змінна для створення унікального ID для кожного студента
        private int studentId; // ID студента
        private String lastName;
        private String firstName;
        private int dayOfBirth;
        private int monthOfBirth;
        private int yearOfBirth;
        private String hobby;

        public Student(String lastName, String firstName, int dayOfBirth, int monthOfBirth, int yearOfBirth, String hobby) {
            this.studentId = nextId++; // Присвоєння ID та інкрементування для наступного студента
            this.lastName = lastName;
            this.firstName = firstName;
            this.dayOfBirth = dayOfBirth;
            this.monthOfBirth = monthOfBirth;
            this.yearOfBirth = yearOfBirth;
            this.hobby = hobby;
        }

        public String getLastName() {
            return lastName;
        }

        public void setLastName(String lastName) {
            this.lastName = lastName;
        }

        public String getFirstName() {
            return firstName;
        }

        public void setFirstName(String firstName) {
            this.firstName = firstName;
        }

        public int getDayOfBirth() {
            return dayOfBirth;
        }

        public void setDayOfBirth(int dayOfBirth) {
            this.dayOfBirth = dayOfBirth;
        }

        public int getMonthOfBirth() {
            return monthOfBirth;
        }

        public void setMonthOfBirth(int monthOfBirth) {
            this.monthOfBirth = monthOfBirth;
        }

        public int getYearOfBirth() {
            return yearOfBirth;
        }

        public void setYearOfBirth(int yearOfBirth) {
            this.yearOfBirth = yearOfBirth;
        }

        public String getHobby() {
            return hobby;
        }

        public void setHobby(String hobby) {
            this.hobby = hobby;
        }

        @Override
        public String toString() {
            return String.format("ID: (%d), %s, data народження: %d/%d/%d, hobby: %s", studentId, lastName, firstName, dayOfBirth, monthOfBirth, yearOfBirth, hobby);
        }
    }
}

```

Цей код визначає клас Student, який представляє студента з унікальним ID та особистою інформацією. Ось детальне пояснення:

- `private static int nextId = 1;` - це статична змінна, яка використовується для створення унікального ID для кожного нового студента. Вона ініціалізується значенням 1 і збільшується на 1 кожного разу, коли створюється новий об'єкт Student (ця змінна була створена для кращого бачення видалених студентів, що мають хобі туризм та народились літом).
- `public Student(...)` - це конструктор класу, який приймає параметри для ініціалізації об'єкта. Він присвоює студенту унікальний ID, збільшуючи `nextId` на 1.
- `public String getLastName()` - це метод “getter”, який повертає прізвище студента.
- `public void setLastName(String lastName)` - це метод “setter”, який встановлює нове прізвище студента.
- `public String toString()` - це перевизначений метод, який повертає рядок, що представляє об'єкт Student. Він використовує `String.format` для створення форматowanego рядка з ID студента, прізвищем, ім'ям, датою народження та хобі.

Методи “getter” та “setter” використовуються для доступу та оновлення значень полів об'єкта, а `toString()` для представлення об'єкта у зручному для читання форматі.

Клас послідовного пошуку:

```
public static ArrayList<Student> sequentialSearch(Student[] students, String hobby) {  
    ArrayList<Student> foundStudents = new ArrayList<>();  
    for (Student student : students) {  
        if (student.getHobby().equals(hobby)) {  
            foundStudents.add(student);  
        }  
    }  
    return foundStudents;  
}
```

Метод `sequentialSearch` приймає масив студентів (`Student[] students`) та рядок `hobby`. Його завдання полягає в пошуку студентів, які мають задане хобі.

Детальне пояснення:

1. Параметри методу:

- `Student[] students`: Масив об'єктів класу `Student`, який містить інформацію про студентів.
- `String hobby`: Рядок, що представляє задане хобі, яке ми шукаємо серед студентів.

2. Створення об'єкту `ArrayList<Student>`:

- Створюється порожній список `foundStudents`, який буде містити знайдених студентів з відповідним хобі.

3. Цикл `for-each`:

- Проходимо через кожного студента в масиві `students`.
- Для кожного студента перевіряємо, чи співпадає його хобі з заданим `hobby`.

4. Умова перевірки хобі:

- `if (student.getHobby().equals(hobby))`: Перевіряємо, чи рядок `hobby` співпадає з хобі студента. Якщо так, то додаємо студента до списку `foundStudents`.

5. Повернення результату:

- Після завершення циклу повертаємо список `foundStudents`, який містить студентів з відповідним хобі.

Цей метод дозволяє знаходити студентів за певним критерієм (у цьому випадку - за хобі) та повертати їх у вигляді списку. Він може бути корисним,

наприклад, для фільтрації студентів за певними інтересами або для подальшого аналізу даних.

Клас для видалення студента-туриста, що народився влітку:

```
public static void removeSummerTourismStudents(Student[] students) {  
    ArrayList<Student> updatedStudents = new ArrayList<>();  
    for (Student student : students) {  
        if (!(student.getMonthOfBirth() >= 6 && student.getMonthOfBirth() <= 8) || !student.getHobby().equals("туризм")) {  
            updatedStudents.add(student);  
        }  
    }  
    students = updatedStudents.toArray(new Student[0]);  
}
```

Метод `removeSummerTourismStudents` приймає масив студентів (`Student[] students`). Його завдання полягає в видаленні студентів, які мають хобі “туризм” та народилися влітку (від червня до серпня).

Детальне пояснення:

1. Параметри методу:

- `Student[] students`: Масив об’єктів класу `Student`, який містить інформацію про студентів.

2. Створення об’єкту `ArrayList<Student>`:

- Створюється порожній список `updatedStudents`, який буде містити оновлену інформацію про студентів.

3. Цикл `for-each`:

- Проходимо через кожного студента в масиві `students`.
- Для кожного студента перевіряємо дві умови:
 - Чи не народився студент влітку (місяці з червня по серпень): `!(student.getMonthOfBirth() >= 6 && student.getMonthOfBirth() <= 8)`
 - Чи не має студент хобі “туризм”: `!student.getHobby().equals("туризм")`

4. Умова видалення студента:

- Якщо студент не відповідає жодній з умов, то додаємо його до списку `updatedStudents`.

5. Оновлення масиву `students`:

- Після завершення циклу перетворюємо список `updatedStudents` назад у масив `students` за допомогою `toArray(new Student[0])`.

```
Run | Debug
public static void main(String[] args) {
    // Ініціалізуємо масив студентів
    Student[] students = {
        new Student(lastName:"Петров", firstName:"Іван", dayOfBirth:15, monthOfBirth:6, ...2000, "туризм"),
        new Student(lastName:"Іванов", firstName:"Петро", dayOfBirth:20, monthOfBirth:8, ...2001, "малювання"),
        new Student(lastName:"Сидоров", firstName:"Олег", dayOfBirth:12, monthOfBirth:5, ...2002, "музика"),
        new Student(lastName:"Ковальчук", firstName:"Марія", dayOfBirth:10, monthOfBirth:10, ...2000, "спорт"),
        new Student(lastName:"Василенко", firstName:"Анна", dayOfBirth:5, monthOfBirth:2, ...1999, "література"),
        new Student(lastName:"Ткаченко", firstName:"Олександр", dayOfBirth:18, monthOfBirth:7, ...2001, "театр"),
        new Student(lastName:"Савченко", firstName:"Катерина", dayOfBirth:25, monthOfBirth:12, ...1998, "малювання"),
        new Student(lastName:"Павленко", firstName:"Юрій", dayOfBirth:8, monthOfBirth:9, ...2000, "спорт"),
        new Student(lastName:"Мельник", firstName:"Оксана", dayOfBirth:30, monthOfBirth:4, ...2001, "музика"),
        new Student(lastName:"Попов", firstName:"Вадим", dayOfBirth:14, monthOfBirth:3, ...1999, "спорт"),
        new Student(lastName:"Козлов", firstName:"Ігор", dayOfBirth:7, monthOfBirth:8, ...2002, "туризм"),
        new Student(lastName:"Григоренко", firstName:"Тетяна", dayOfBirth:3, monthOfBirth:1, ...2000, "література"),
        new Student(lastName:"Бойко", firstName:"Роман", dayOfBirth:28, monthOfBirth:8, ...2001, "малювання"),
        new Student(lastName:"Лисенко", firstName:"Ірина", dayOfBirth:19, monthOfBirth:6, ...1999, "спорт"),
        new Student(lastName:"Мороз", firstName:"Андрій", dayOfBirth:23, monthOfBirth:9, ...2000, "музика"),
        new Student(lastName:"Шевченко", firstName:"Віктор", dayOfBirth:11, monthOfBirth:7, ...2002, "театр"),
        new Student(lastName:"Коваленко", firstName:"Олена", dayOfBirth:9, monthOfBirth:5, ...2001, "туризм"),
        new Student(lastName:"Павлюк", firstName:"Наталія", dayOfBirth:17, monthOfBirth:4, ...1999, "малювання"),
        new Student(lastName:"Сепрієнко", firstName:"Максим", dayOfBirth:22, monthOfBirth:11, ...2000, "спорт"),
        new Student(lastName:"Кучеренко", firstName:"Лідія", dayOfBirth:6, monthOfBirth:10, ...2002, "музика")
    };
}
```

Ініціалізуємо одновимірний масив студентів, кількість яких не менше 20.

```
// Виводимо вміст попереднього масиву студентів
System.out.println(x:"Попередній масив студентів:");
for (Student student : students) {
    System.out.println(student);
}
```

Даний кусок коду виконує виведення базового масиву студентів. Це робиться за допомогою циклу for-each, який проходить через кожен об'єкт Student у масиві students і виводить його за допомогою методу toString().

```
// Виконуємо пошук студентів за заданим хобі (туризм)
ArrayList<Student> foundStudents = sequentialSearch(students, hobby:"туризм");
if (!foundStudents.isEmpty()) {
    System.out.println(x:"\nЗнайдено студентів з хобі туризм:");
    for (Student student : foundStudents) {
        System.out.println(student);
    }
} else {
    System.out.println(x:"\nСтудентів з вказаним хобі не знайдено.");
}
```

Даний кусок коду виконує пошук студентів-туристів. В ньому викликається метод sequentialSearch, який повертає список студентів (foundStudents), що мають хобі “туризм”. Якщо список не порожній, виводяться знайдені студенти. Якщо порожній, виводиться повідомлення про те, що студентів з вказаним хобі не знайдено.

```
// Видаляємо студентів, народжених улітку та захоплюються туризмом
ArrayList<Student> updatedStudents = new ArrayList<>();
for (Student student : students) {
    if (!(student.getMonthOfBirth() >= 6 && student.getMonthOfBirth() <= 8) || !student.getHobby().equals(anObject:"туризм")) {
        updatedStudents.add(student);
    }
}
students = updatedStudents.toArray(new Student[0]);
```

Даний кусок коду видаляє студента за заданим критерієм в завданні. Створюється новий список updatedStudents, до якого додаються студенти, що не відповідають умовам видалення (не народилися влітку або не мають хобі “туризм”). Після цього масив students оновлюється, і в нього записуються лише ті студенти, що залишилися в updatedStudents.

```
// Виводимо вміст масиву після видалення
System.out.println(x:"\nМасив після видалення студентів, що народилися літом та захоплюються туризмом:");
for (Student student : students) {
    System.out.println(student);
}
```

Виводиться вміст оновленого масиву студентів після видалення. Це робиться так само, як і у виводі базового масиву, за допомогою циклу for-each і методу toString().

РЕЗУЛЬТАТИ РОБОТИ ПРОГРАМИ:

Виконаємо перевірку роботи програми. Першим прикладом буде список студентів, де буде 3 студенти-туристи та тільки 2 студента з літньою датою народження.

```
Student[] students = {
    new Student(lastName:"Петров", firstName:"Іван", dayOfBirth:15, monthOfBirth:6, _2000, "туризм"),
    new Student(lastName:"Іванов", firstName:"Петро", dayOfBirth:20, monthOfBirth:8, _2001, "малювання"),
    new Student(lastName:"Сидоров", firstName:"Олер", dayOfBirth:12, monthOfBirth:5, _2002, "музика"),
    new Student(lastName:"Ковальчук", firstName:"Марія", dayOfBirth:10, monthOfBirth:10, _2000, "спорт"),
    new Student(lastName:"Василенко", firstName:"Анна", dayOfBirth:5, monthOfBirth:2, _1999, "література"),
    new Student(lastName:"Ткаченко", firstName:"Олександр", dayOfBirth:18, monthOfBirth:7, _2001, "театр"),
    new Student(lastName:"Савченко", firstName:"Катерина", dayOfBirth:25, monthOfBirth:12, _1998, "малювання"),
    new Student(lastName:"Павленко", firstName:"Юрій", dayOfBirth:8, monthOfBirth:9, _2000, "спорт"),
    new Student(lastName:"Мельник", firstName:"Оксана", dayOfBirth:30, monthOfBirth:4, _2001, "музика"),
    new Student(lastName:"Попов", firstName:"Вадим", dayOfBirth:14, monthOfBirth:3, _1999, "спорт"),
    new Student(lastName:"Козлов", firstName:"Ігор", dayOfBirth:7, monthOfBirth:8, _2002, "туризм"),
    new Student(lastName:"Григоренко", firstName:"Тетяна", dayOfBirth:3, monthOfBirth:1, _2000, "література"),
    new Student(lastName:"Бойко", firstName:"Роман", dayOfBirth:28, monthOfBirth:8, _2001, "малювання"),
    new Student(lastName:"Лисенко", firstName:"Ірина", dayOfBirth:19, monthOfBirth:6, _1999, "спорт"),
    new Student(lastName:"Мороз", firstName:"Андрій", dayOfBirth:23, monthOfBirth:9, _2000, "музика"),
    new Student(lastName:"Шевченко", firstName:"Віктор", dayOfBirth:11, monthOfBirth:7, _2002, "театр"),
    new Student(lastName:"Коваленко", firstName:"Олена", dayOfBirth:9, monthOfBirth:5, _2001, "туризм"),
    new Student(lastName:"Павлюк", firstName:"Наталія", dayOfBirth:17, monthOfBirth:4, _1999, "малювання"),
    new Student(lastName:"Сепієнко", firstName:"Максим", dayOfBirth:22, monthOfBirth:11, _2000, "спорт"),
    new Student(lastName:"Кучеренко", firstName:"Лідія", dayOfBirth:6, monthOfBirth:10, _2002, "музика")
};
```


Отримуємо такі результати:

Попередній масив студентів:

```
(1) (Петров, Іван, Дата народження: 15/6/2000, Хобі: туризм)
(2) (Іванов, Петро, Дата народження: 20/8/2001, Хобі: малювання)
(3) (Сидоров, Олег, Дата народження: 12/5/2002, Хобі: музика)
(4) (Ковальчук, Марія, Дата народження: 10/10/2000, Хобі: спорт)
(5) (Василенко, Анна, Дата народження: 5/2/1999, Хобі: література)
(6) (Ткаченко, Олександр, Дата народження: 18/7/2001, Хобі: театр)
(7) (Савченко, Катерина, Дата народження: 25/12/1998, Хобі: малювання)
(8) (Павленко, Юрій, Дата народження: 8/9/2000, Хобі: спорт)
(9) (Мельник, Оксана, Дата народження: 30/4/2001, Хобі: музика)
(10) (Попов, Вадим, Дата народження: 14/3/1999, Хобі: спорт)
(11) (Козлов, Ігор, Дата народження: 7/8/2002, Хобі: туризм)
(12) (Григоренко, Тетяна, Дата народження: 3/1/2000, Хобі: література)
(13) (Бойко, Роман, Дата народження: 28/8/2001, Хобі: малювання)
(14) (Лисенко, Ірина, Дата народження: 19/6/1999, Хобі: спорт)
(15) (Мороз, Андрій, Дата народження: 23/9/2000, Хобі: музика)
(16) (Шевченко, Віктор, Дата народження: 11/7/2002, Хобі: театр)
(17) (Коваленко, Олена, Дата народження: 9/5/2001, Хобі: туризм)
(18) (Павлюк, Наталія, Дата народження: 17/4/1999, Хобі: малювання)
(19) (Сергієнко, Максим, Дата народження: 22/11/2000, Хобі: спорт)
(20) (Кучеренко, Лідія, Дата народження: 6/10/2002, Хобі: музика)
```

Масив після видалення студентів, що народилися літом та захоплюються туризмом:

```
(2) (Іванов, Петро, Дата народження: 20/8/2001, Хобі: малювання)
(3) (Сидоров, Олег, Дата народження: 12/5/2002, Хобі: музика)
(4) (Ковальчук, Марія, Дата народження: 10/10/2000, Хобі: спорт)
(5) (Василенко, Анна, Дата народження: 5/2/1999, Хобі: література)
(6) (Ткаченко, Олександр, Дата народження: 18/7/2001, Хобі: театр)
(7) (Савченко, Катерина, Дата народження: 25/12/1998, Хобі: малювання)
(8) (Павленко, Юрій, Дата народження: 8/9/2000, Хобі: спорт)
(9) (Мельник, Оксана, Дата народження: 30/4/2001, Хобі: музика)
(10) (Попов, Вадим, Дата народження: 14/3/1999, Хобі: спорт)
(12) (Григоренко, Тетяна, Дата народження: 3/1/2000, Хобі: література)
(13) (Бойко, Роман, Дата народження: 28/8/2001, Хобі: малювання)
(14) (Лисенко, Ірина, Дата народження: 19/6/1999, Хобі: спорт)
(15) (Мороз, Андрій, Дата народження: 23/9/2000, Хобі: музика)
(16) (Шевченко, Віктор, Дата народження: 11/7/2002, Хобі: театр)
(17) (Коваленко, Олена, Дата народження: 9/5/2001, Хобі: туризм)
(18) (Павлюк, Наталія, Дата народження: 17/4/1999, Хобі: малювання)
(19) (Сергієнко, Максим, Дата народження: 22/11/2000, Хобі: спорт)
(20) (Кучеренко, Лідія, Дата народження: 6/10/2002, Хобі: музика)
```

Знайдено студентів з хобі туризм:

```
(1) (Петров, Іван, Дата народження: 15/6/2000, Хобі: туризм)
(11) (Козлов, Ігор, Дата народження: 7/8/2002, Хобі: туризм)
(17) (Коваленко, Олена, Дата народження: 9/5/2001, Хобі: туризм)
```

Наступним прикладом роботи програми буде 0 студентів-туристів:

```
// Ініціалізуємо масив студентів
Student[] students = {
    new Student(lastName:"Петров", firstName:"Іван", dayOfBirth:15, monthOfBirth:6, _2000, "музика"),
    new Student(lastName:"Іванов", firstName:"Петро", dayOfBirth:20, monthOfBirth:8, _2001, "малювання"),
    new Student(lastName:"Сидоров", firstName:"Олер", dayOfBirth:12, monthOfBirth:5, _2002, "музика"),
    new Student(lastName:"Ковальчук", firstName:"Марія", dayOfBirth:10, monthOfBirth:10, _2000, "спорт"),
    new Student(lastName:"Василенко", firstName:"Анна", dayOfBirth:5, monthOfBirth:2, _1999, "література"),
    new Student(lastName:"Ткаченко", firstName:"Олександр", dayOfBirth:18, monthOfBirth:7, _2001, "театр"),
    new Student(lastName:"Савченко", firstName:"Катерина", dayOfBirth:25, monthOfBirth:12, _1998, "малювання"),
    new Student(lastName:"Павленко", firstName:"Юрій", dayOfBirth:8, monthOfBirth:9, _2000, "спорт"),
    new Student(lastName:"Мельник", firstName:"Оксана", dayOfBirth:30, monthOfBirth:4, _2001, "музика"),
    new Student(lastName:"Попов", firstName:"Вадим", dayOfBirth:14, monthOfBirth:3, _1999, "спорт"),
    new Student(lastName:"Козлов", firstName:"Ігор", dayOfBirth:7, monthOfBirth:8, _2002, "музика"),
    new Student(lastName:"Григоренко", firstName:"Тетяна", dayOfBirth:3, monthOfBirth:1, _2000, "література"),
    new Student(lastName:"Бойко", firstName:"Роман", dayOfBirth:28, monthOfBirth:8, _2001, "малювання"),
    new Student(lastName:"Лисенко", firstName:"Ірина", dayOfBirth:19, monthOfBirth:6, _1999, "спорт"),
    new Student(lastName:"Мороз", firstName:"Андрій", dayOfBirth:23, monthOfBirth:9, _2000, "музика"),
    new Student(lastName:"Шевченко", firstName:"Віктор", dayOfBirth:11, monthOfBirth:7, _2002, "театр"),
    new Student(lastName:"Коваленко", firstName:"Олена", dayOfBirth:9, monthOfBirth:5, _2001, "музика"),
    new Student(lastName:"Павлюк", firstName:"Наталія", dayOfBirth:17, monthOfBirth:4, _1999, "малювання"),
    new Student(lastName:"Сергієнко", firstName:"Максим", dayOfBirth:22, monthOfBirth:11, _2000, "спорт"),
    new Student(lastName:"Кучеренко", firstName:"Лідія", dayOfBirth:6, monthOfBirth:10, _2002, "музика")
};
```

Отримуємо такі результати:

Попередній масив студентів:

```
(1) (Петров, Іван, Дата народження: 15/6/2000, Хобі: музика)
(2) (Іванов, Петро, Дата народження: 20/8/2001, Хобі: малювання)
(3) (Сидоров, Олег, Дата народження: 12/5/2002, Хобі: музика)
(4) (Ковальчук, Марія, Дата народження: 10/10/2000, Хобі: спорт)
(5) (Василенко, Анна, Дата народження: 5/2/1999, Хобі: література)
(6) (Ткаченко, Олександр, Дата народження: 18/7/2001, Хобі: театр)
(7) (Савченко, Катерина, Дата народження: 25/12/1998, Хобі: малювання)
(8) (Павленко, Юрій, Дата народження: 8/9/2000, Хобі: спорт)
(9) (Мельник, Оксана, Дата народження: 30/4/2001, Хобі: музика)
(10) (Попов, Вадим, Дата народження: 14/3/1999, Хобі: спорт)
(11) (Козлов, Ігор, Дата народження: 7/8/2002, Хобі: музика)
(12) (Григоренко, Тетяна, Дата народження: 3/1/2000, Хобі: література)
(13) (Бойко, Роман, Дата народження: 28/8/2001, Хобі: малювання)
(14) (Лисенко, Ірина, Дата народження: 19/6/1999, Хобі: спорт)
(15) (Мороз, Андрій, Дата народження: 23/9/2000, Хобі: музика)
(16) (Шевченко, Віктор, Дата народження: 11/7/2002, Хобі: театр)
(17) (Коваленко, Олена, Дата народження: 9/5/2001, Хобі: музика)
(18) (Павлюк, Наталія, Дата народження: 17/4/1999, Хобі: малювання)
(19) (Сергієнко, Максим, Дата народження: 22/11/2000, Хобі: спорт)
(20) (Кучеренко, Лідія, Дата народження: 6/10/2002, Хобі: музика)
```

Масив після видалення студентів, що народилися літом та захоплюються туризмом:

```
(1) (Петров, Іван, Дата народження: 15/6/2000, Хобі: музика)
(2) (Іванов, Петро, Дата народження: 20/8/2001, Хобі: малювання)
(3) (Сидоров, Олег, Дата народження: 12/5/2002, Хобі: музика)
(4) (Ковальчук, Марія, Дата народження: 10/10/2000, Хобі: спорт)
(5) (Василенко, Анна, Дата народження: 5/2/1999, Хобі: література)
(6) (Ткаченко, Олександр, Дата народження: 18/7/2001, Хобі: театр)
(7) (Савченко, Катерина, Дата народження: 25/12/1998, Хобі: малювання)
(8) (Павленко, Юрій, Дата народження: 8/9/2000, Хобі: спорт)
(9) (Мельник, Оксана, Дата народження: 30/4/2001, Хобі: музика)
(10) (Попов, Вадим, Дата народження: 14/3/1999, Хобі: спорт)
(11) (Козлов, Ігор, Дата народження: 7/8/2002, Хобі: музика)
(12) (Григоренко, Тетяна, Дата народження: 3/1/2000, Хобі: література)
(13) (Бойко, Роман, Дата народження: 28/8/2001, Хобі: малювання)
(14) (Лисенко, Ірина, Дата народження: 19/6/1999, Хобі: спорт)
(15) (Мороз, Андрій, Дата народження: 23/9/2000, Хобі: музика)
(16) (Шевченко, Віктор, Дата народження: 11/7/2002, Хобі: театр)
(17) (Коваленко, Олена, Дата народження: 9/5/2001, Хобі: музика)
(18) (Павлюк, Наталія, Дата народження: 17/4/1999, Хобі: малювання)
(19) (Сергієнко, Максим, Дата народження: 22/11/2000, Хобі: спорт)
(20) (Кучеренко, Лідія, Дата народження: 6/10/2002, Хобі: музика)
```

Студентів з вказаним хобі не знайдено.

ВИСНОВОК:

В ході лабораторної роботи було досліджено алгоритми пошуку та набуто практичних навичок з пошуку елементів у лінійних та нелінійних структурах даних. Описано студентів згідно з варіантом завдання (Прізвище, ім'я, день, місяць і рік народження, хобі).

Описано метод, який виконує пошук в одновимірному масиві студентів за заданим алгоритмом згідно з варіантом завдання (Послідовний). Створено та ініційовано екземпляр лінійної структури даних (одновимірний масив студентів розміром не менш 20 елементів) з урахуванням правила формування (Невпорядкований).

Виконано завдання (Видалити студентів, народжених улітку, що захоплюються туризмом). Виведено вміст одновимірного масиву в разі його зміни (до і після видалення) або після виконання всього завдання.