

Практическое занятие № 14

Тема практического занятия: Алгоритмы поиска и сортировки данных.

Цель практического занятия: сформировать навыки эффективно работать с данными, искать и сортировать их с высокой производительностью и эффективностью, а также настраивать и оптимизировать алгоритмы для удовлетворения требований конкретных задач.

В результате выполнения данной работы обучающийся должен уметь:

1. Реализовывать алгоритмы поиска и сортировки на практике с использованием языка программирования
2. Выбирать наиболее подходящий алгоритм поиска или сортировки для конкретной задачи с учетом требований эффективности и производительности
3. Адаптировать и модифицировать алгоритмы поиска и сортировки для решения конкретных задач

знать:

1. Основные алгоритмы поиска
2. Основные алгоритмы сортировки
3. Особенности и принципы работы различных алгоритмов поиска и сортировки и их применение в реальных сценариях

Перечень оборудования, необходимого для выполнения задания:

- Автоматизированные рабочие места по количеству обучающихся (процессор Intel Core i7 или аналогичный, БП 700 Вт, 32 Гб ОЗУ, SSD 512 Гб, HDD 4 ТБ SATA 7200 rpm, RTX 3060 12GB);
 - Автоматизированное рабочее место преподавателя (процессор Intel Core i7 или аналогичный, БП 700 Вт, 32 Гб ОЗУ, SSD 512 Гб, HDD 4 ТБ SATA 7200 rpm, RTX 3060 12GB);
 - Монитор 34", изогнутый, 3440x1440, 6 ms, 178°/178°, 300 cd/m2, 20M:1, HDMI, DP, USB, регулировка по высоте (1 монитор на одно рабочее место).
 - Интерактивная панель 86" с OPS ПК
- Программное обеспечение:
- Microsoft Visual Studio.

Общие теоретические сведения:

Существует несколько основных алгоритмов поиска. Вот некоторые из них:

1. Линейный поиск: при этом алгоритме элементы последовательно проверяются один за другим до нахождения искомого элемента или достижения конца последовательности.

2. Бинарный поиск: применяется к отсортированной последовательности. Поиск начинается с середины, а затем искомый элемент сравнивается с элементом в середине. Если элемент равен искомому, то поиск заканчивается. Если элемент меньше искомого, то поиск продолжается в левой половине последовательности, иначе в правой половине.

3. Поиск с использованием хэш-таблиц: основан на хэш-функциях, которые используются для получения индекса элемента в хэш-таблице. Данный алгоритм обеспечивает быстрый доступ к данным, но требует больше памяти.

Существует множество алгоритмов сортировки, каждый из которых имеет свои особенности и эффективность в различных ситуациях. Некоторые из наиболее распространенных алгоритмов сортировки включают:

1. Сортировка пузырьком: Этот алгоритм проходит по списку несколько раз, меняя местами соседние элементы, если они находятся в неправильном порядке. Он продолжает выполнение до тех пор, пока весь список не будет отсортирован.

2. Сортировка выбором: Алгоритм находит наименьший элемент в списке и помещает его в начало. Затем он ищет наименьший элемент в оставшейся части списка и помещает его после первого элемента. Процесс продолжается до полной сортировки списка.

3. Сортировка вставками: Этот алгоритм проходит по списку и в каждой итерации выбирает элемент и вставляет его в правильную позицию в уже отсортированной части списка. Он продолжается до полной сортировки списка.

4. Сортировка слиянием: Этот алгоритм рекурсивно разбивает список на две части, сортирует каждую из них отдельно, а затем объединяет их в один упорядоченный список. Он использует метод "разделяй и властвуй".

5. Быстрая сортировка: Этот алгоритм также использует метод "разделяй и властвуй". Он выбирает элемент, называемый опорным, и разбивает список на две части - элементы, меньшие опорного, и элементы, большие опорного. Затем он рекурсивно применяет этот процесс к каждой из частей, пока не будет достигнута полная сортировка.

Задание:

1. Доработайте библиотеку классов, созданную ранее во время практического занятия №9: добавьте класс «Достопримечательность» с полями и свойствами: название, год возникновения.
2. Замените в классе «Город» число достопримечательностей на список достопримечательностей.
3. Добавьте возможность инициализации списка достопримечательностей в конструктор класса.
4. Разработайте приложение, которое позволит создать город со списком достопримечательностей, искать самую старую достопримечательность в

городе, вывести все достопримечательности в порядке от самой старой к самой новой.

Указания по технике безопасности:

Инструкция по технике безопасности при работе в лаборатории, оборудованной компьютерной техникой.

Технология выполнения работы (этапы, последовательность действий):

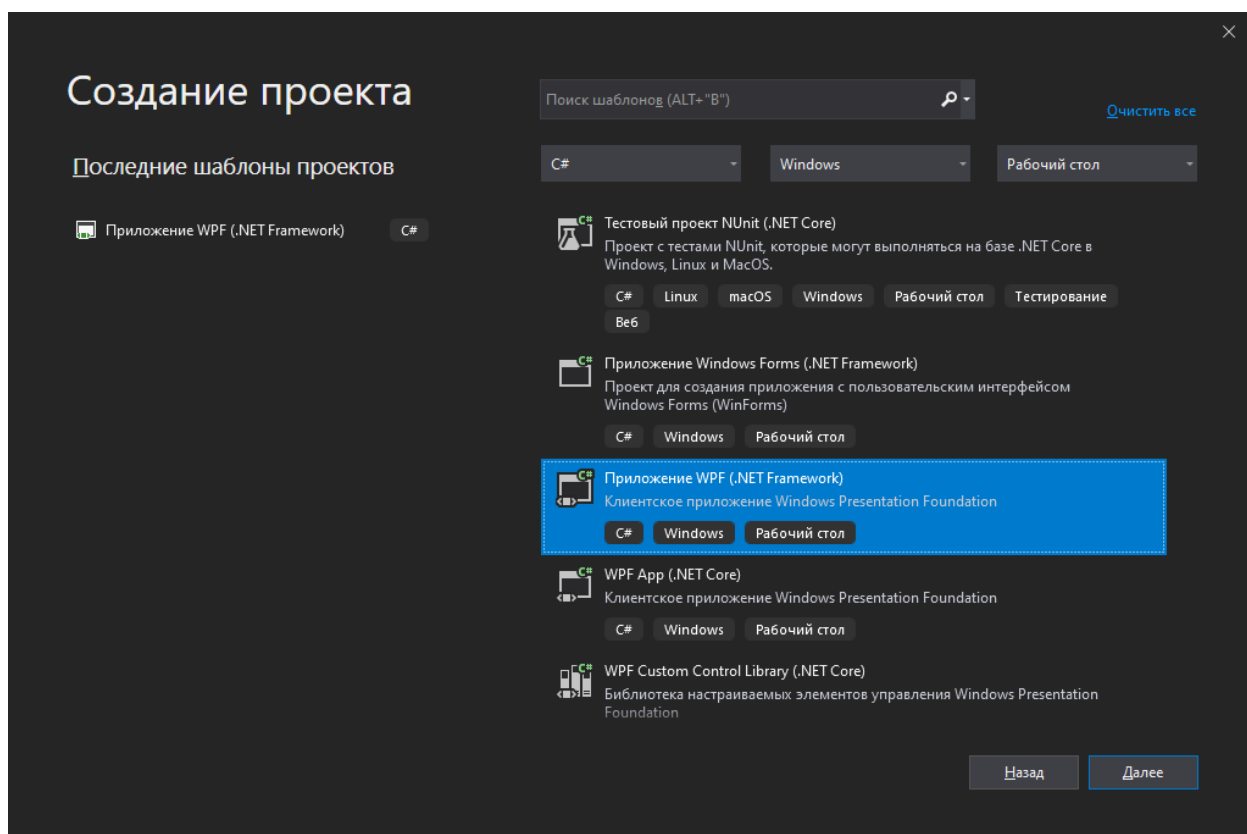
1. Добавьте в библиотеку классов новый класс – «Достопримечательность», опишите его в соответствии с заданием. Пример описания класса:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace RoomLibrary
8  {
9      Ссылка: 0
10     public class Room
11     {
12         double roomLength; //длина комнаты
13         double roomWidth; //ширина комнаты
14         Ссылка: 0
15         public double RoomLength
16         {
17             get { return roomLength; }
18             set { roomLength = value; }
19         }
20         Ссылка: 0
21         public double RoomWidth
22         {
23             get { return roomWidth; }
24             set { roomWidth = value; }
25         }
26         Ссылка: 0
27         public double RoomPerimeter()
28         {
29             return 2 * (roomLength + roomWidth);
30         }
31     }
32 }
```

В примере приведен другой класс, используйте его как аналогию для описания класса «Достопримечательность».

2. Модифицируйте класс «Город»: добавьте список достопримечательностей, модифицируйте конструктор.
3. Пересоберите библиотеку, убедитесь в успешности сборки. Чтобы пересобрать библиотеку, выберите пункт меню «Сборка», далее строку «Пересобрать решение».
4. Создайте проект для приложения.

Пример создания проекта приложения для MS Visual Studio:



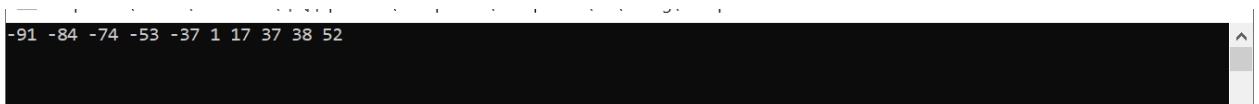
5. Разработайте интерфейс приложения, разместив все необходимые элементы управления.
6. Разработайте обработчик событий, который позволит сформировать список достопримечательностей
7. Убедитесь в работоспособности этой части приложения.
8. Разработайте обработчик событий, который позволит ввести информацию о городе.
9. Убедитесь в работоспособности этой части приложения.
10. Разработайте обработчик событий, который позволит найти самую старую достопримечательность. Используйте линейный поиск.
11. Убедитесь в работоспособности этой части приложения.
12. Разработайте обработчик событий, который позволит вывести упорядоченный список достопримечательностей. Можно использовать как метод вставки, так и метод выбора.
13. Убедитесь в работоспособности этой части приложения.

Пример кода, осуществляющего сортировку элементов массива:

```
static void Main(string[] args)
{
    int n = 10;
    Random rnd = new Random();
    int[] a = new int[n];
    for (int i = 0; i < n; i++)
        a[i] = rnd.Next(-100, 101);
    for (int i=0; i<n-1;i++)
    {
        int min = a[i];
        int minPos = i;
        for (int j=i+1; j<n; j++)
            if (a[j]<min)
            {
                min = a[j];
                minPos = j;
            }
        a[minPos] = a[i];
        a[i] = min;
    }
    foreach (int x in a)
        Console.Write(x + " ");
    Console.ReadKey();
}
```

Здесь используется так называемый «методы выбора» - сперва мы выбираем наименьший элемент, и ставим его на самое первое место, затем – наименьший из оставшихся, и ставим его на второе место и так далее, пока весь массив не будет отсортирован. Метод является довольно медленным, но интуитивно понятным. Сам массив при этом генерируется случайным образом. Варианты работы программы:

Первый вариант:



Второй вариант:

Можем убедиться в том, что во всех случаях выводится массив, элементы которого отсортированы в порядке возрастания.

Требование к отчету:

1. В библиотеку классов добавлен класс «Достопримечательность».
2. В класс «Город» добавлен список достопримечательностей.
3. В классе «Город» конструктор позволяет инициализировать список достопримечательностей.
4. Приложение позволяет вводить информацию о городе и его достопримечательностях.
5. Приложение выводит самую старую достопримечательность.
6. Приложение выводит список достопримечательностей в заданном порядке.

Контрольные вопросы:

1. Какие алгоритмы используются для поиска элемента в массиве?
2. Что такое линейный поиск?
3. Что такое бинарный поиск?
4. Что такое сортировка массива? Зачем она нужна?
5. Что такое сортировка пузырьком?
6. Что такое сортировка выбором?
7. Что такое сортировка вставками?
8. Что такое сортировка слиянием?
9. Что такое быстрая сортировка?

Основные и дополнительные источники, электронные ресурсы:

1. Подбельский, В. В. Язык C#. Базовый курс: учебное пособие / В. В. Подбельский. - 2-е изд., перераб. и доп. - Москва: Финансы и статистика, 2022. - 408 с. - ISBN 978-5-00184-079-4. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1913989>
2. Гуриков, С. Р. Основы алгоритмизации и программирования на Python: учебное пособие / С.Р. Гуриков. — Москва: ИНФРА-М, 2023. — 343 с. — (Среднее профессиональное образование). - ISBN 978-5-16-016906-4. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1927269>.