

Київський національний університет імені Тараса Шевченка

Факультет комп’ютерних наук та кібернетики

Кафедра інформаційних систем

Алгоритми та складність

Лабораторний проект №6 «Реалізація В+ дерева»

Звіт

*Виконав:*

*студент групи К-29*

***Кожановський Олександр Сергійович***

Київ-2018

1. **Умова завдання.**

Реалізувати В+ дерево.

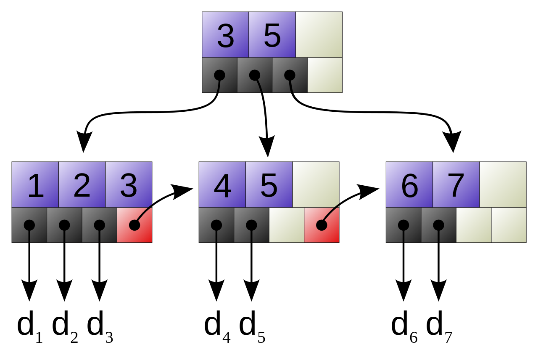
1. **Основні поняття**

*Дерево (англ. tree)* — в інформатиці та програмуванні одна з найпоширеніших структур даних. Формально дерево визначається як скінченна множина Т з однією або більше вершин (вузлів, nodes), яка задовольняє наступним вимогам:

1. існує один відокремлений вузол — корінь (root) дерева
2. інші вузли (за винятком кореня) розподілені серед m ≥ 0 непересічних множин T1…Tm і кожна з цих множин, в свою чергу, є деревом. Дерева T1…Tm мають назву піддерев (subtrees) даного кореня.
3. **Опис алгоритму**

B + -деревом називається збалансоване n-арне дерево пошуку порядку t, що задовольняє наступним властивостям:

* Кожен вузол містить хоча б один ключ; ключі в кожному вузлі впорядковані, корінь містить від 1 до 2t-1 ключів, будь-який інший вузол містить від t-1 до 2t-1 ключів; листя не є винятком з цього правила. Тут t - параметр дерева, не менший 2 (і зазвичай приймає значення від 50 до 2000 в залежності від розміру ключа щодо розміру сторінки, в свою чергу, визначається розміром змістовної записи).
* У листків немає нащадків; для всіх інших вузлів, що містять ключі, заданий вузол містить (n + 1) синів. При цьому:
* перший нащадок і всі його нащадки містять ключі з інтервалу;
* n i-й нащадок і всі його нащадки містять ключі з інтервалу;
* (N + 1) -й син і всі його нащадки містять ключі з інтервалу.

 *(приклад b+ дерева)*

1. **Аналіз алгоритму**

* Максимальна кількість записів, що зберігаються
* Мінімальна кількість записів, що зберігаються
* Мінімальна кількість ключів
* Максимальна кількість
* Простір, необхідний для зберігання дерева
* Вставка запису потребує операцій
* Пошук запису потребує операцій
* Видалення (вже знайденого) запису потребує операцій
* Виконання запиту, який отримує діапазон значень, що містить k елементів, потребує операцій
* Виконання сторінкового запиту з розміром сторінки s та номером сторінки p потребує операцій

1. **Реалізація алгоритму**

Реалізовано на мові С# в середовищі розробки Microsoft [Visual Studio 2017](https://www.visualstudio.com/ru/).

Вибір мови зроблено з урахуванням підтримки Windows Forms.

1. **Розрахунки**

Висота В-дерева Т з n ≥ 1 вузлами та мінімальною степінню t ≥ 2 не перевищує logt(n+1)/2.

Доведення. Нехай В-дерево має висоту h. Корінь містить мінімум 1 ключ, решта вузлів – не менше (t–1) ключа кожен. Отже на глибині 1 маємо мінімум 2 вузли. На глибині 2 мінімум 2 вузлів. На глибині 3 мінімум 2t2 вузлів. ……… На глибині h мінімум 2th–1 вузол. Кількість ключів задовольняє нерівність

Тобто 𝑡h ≤ (𝑛 + 1)/2.

1. **Висновки**

*Свою популярність B+ дерева набули в часи розвитку жорстких дисків. Високий рівень розгалуженості системи та її ієрархія дозволяє мінімізувати час на пошук елементів та компенсувати низьку швидкодію механічних частин накопичувачів. З роками hdd пам’ять витісняється більш швидкою твердотільною пам’яттю, в якій відсутні дані недоліки. Однак, технологія b+ дерева активно використовується в сучасних файлових системах та базах даних, наприклад* [*NTFS*](https://uk.wikipedia.org/wiki/NTFS)*,*[*ReiserFS*](https://uk.wikipedia.org/wiki/ReiserFS)*,*[*NSS*](https://uk.wikipedia.org/w/index.php?title=Novell_Storage_Services&action=edit&redlink=1)*,* [*Microsoft SQL Server*](https://uk.wikipedia.org/wiki/Microsoft_SQL_Server)*,*[*Oracle 8*](https://uk.wikipedia.org/wiki/Oracle_Database)*,*[*Sybase ASE*](https://uk.wikipedia.org/w/index.php?title=Adaptive_Server_Enterprise&action=edit&redlink=1) *та інші, а, отже, є актуальною й на сьогоднішній день.*

1. **Особливості реалізації даного варіанту**

Структурних відмінностей між даними варіантами не виявлено.

1. **Основні модулі програми**

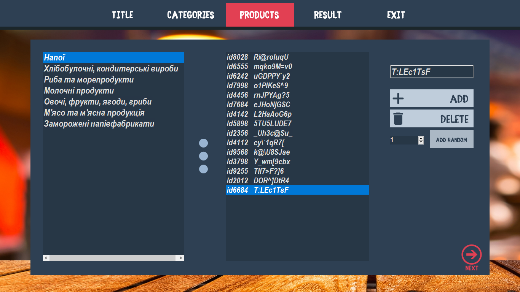
Основна функція “static void Main()” в файлі Program.cs. Функціонал програми прив’язаний до подій з візуальними компонентами. Забезпечення роботи всіх даних знаходиться в файлі main.cs. Реалізація алгоритму в файлі BTree.cs та інших.

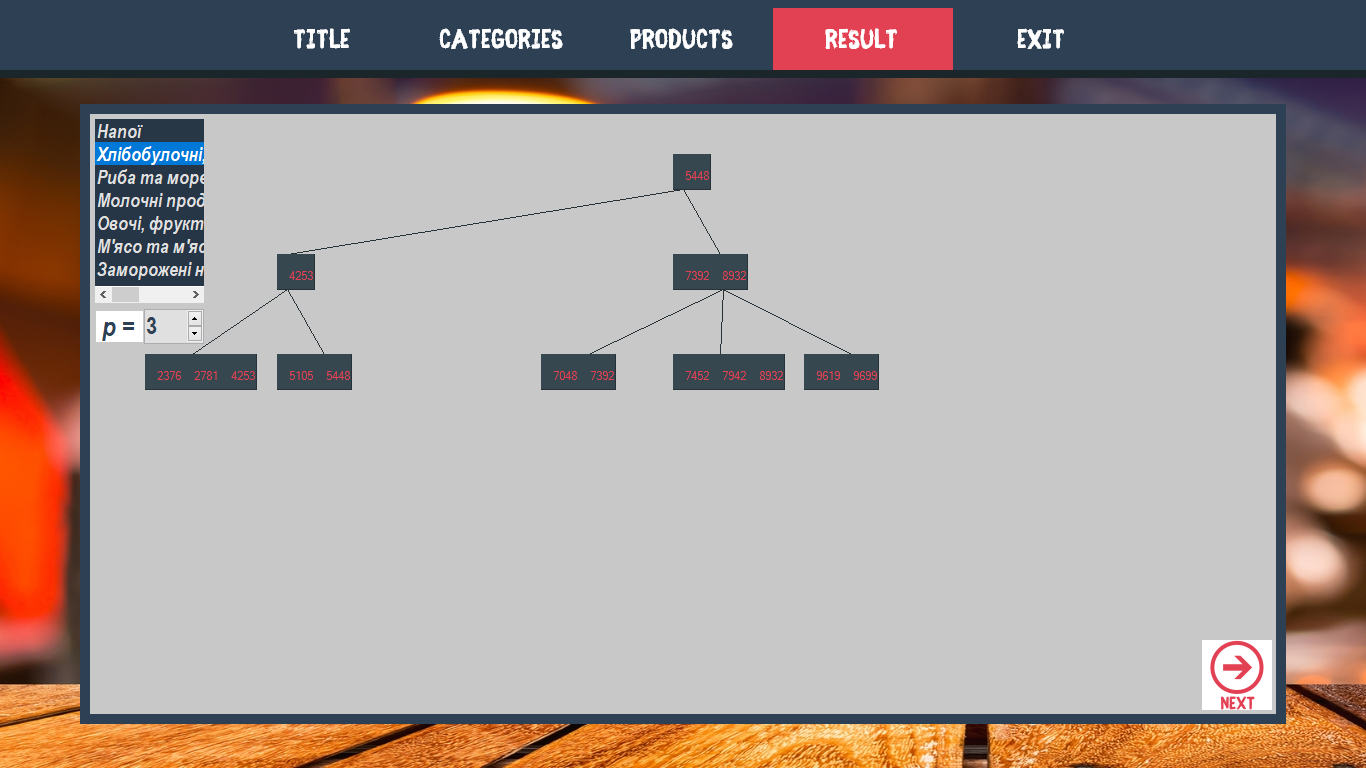
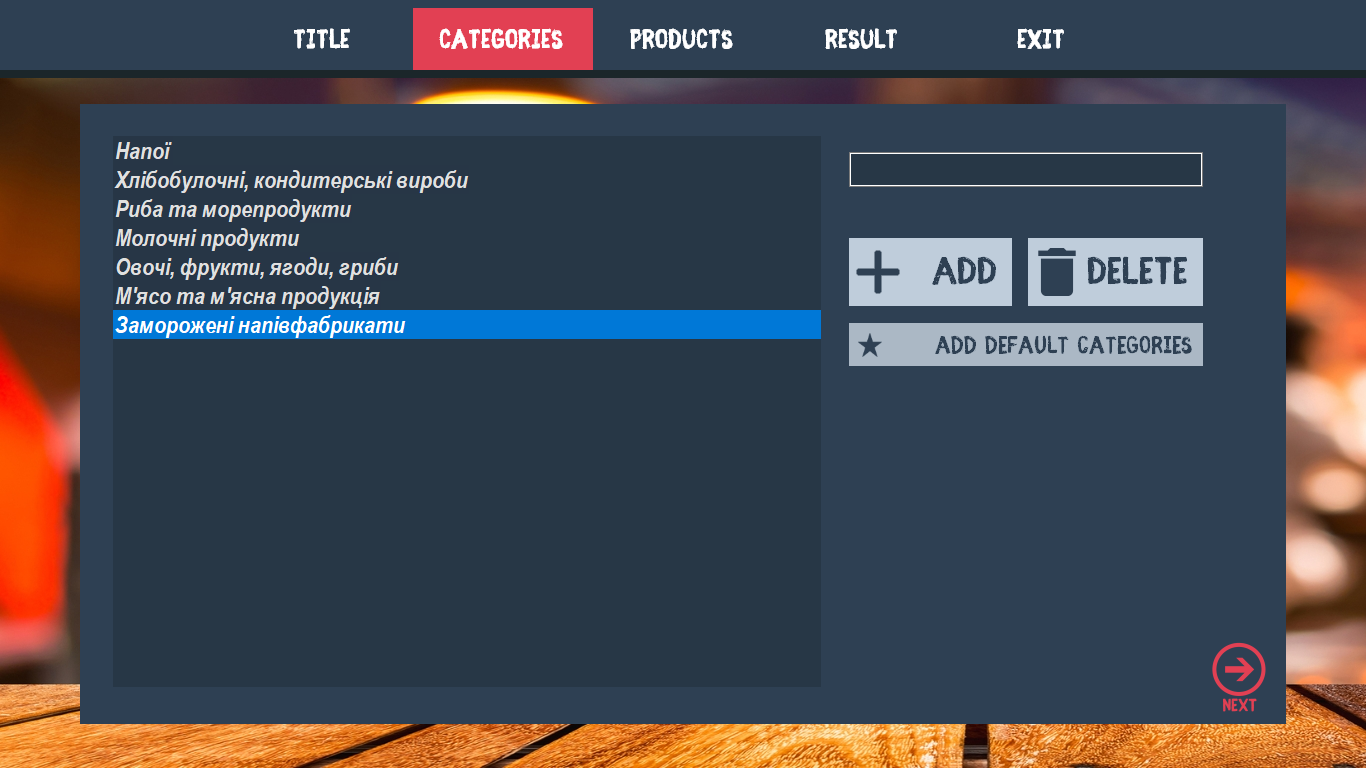
1. **Використані структури даних**

Початкові дані були збережені в класах та їх масивах. Обробка дерева відбувалась з використанням векторів.

1. **Інтерфейс програми**

Графічний:





1. **Тестовий приклад**
2. Запустити програму. Можна відкрити Visual Studio або запустити файл Lab1.exe з папки /bin/Debug/..
3. Натиснути Next. Відкриється вкладка Categories.
4. Натиснути Add default categories. Натиснути Next. Відкриється вкладка Product.
5. Вибрати пункт Напої. Натиснути на Add random декілька разів. Натиснути Next. Відкриється вкладка Result.
6. Вибрати пункт Напої на панелі зліва.
7. Натиснути Exit або Next. Натиснути Exit.
8. **Використані джерела**
9. https://uk.wikipedia.org/wiki/B%2B\_дерево
10. <http://algolist.manual.ru/ds/s_btr.php>
11. <https://www.youtube.com/watch?v=CYKRMz8yzVU>
12. Т. Кормен. Алгоритми: побудова й аналіз. Третє видання (Розділ 18)
13. Дональд Кнут 4. Генерація всіх дерев. Історія комбінаторної генерації // Мистецтво програмування = The Art of Computer Programming. — М.: «Вільямс», 2007. — Т. 4. — С. 160. — ISBN 0-321-33570-8