Обхід бінарних дерев (прямий, симетричний зворотній)

замовник: GO IT

виконавець: група №3 (VID)

мова програмування: JAVA

ментор групи: Анна Смірнова

учасники групи: Пясецький Дмитро

Подгорній Іван

Гудзуляк Богдан

Василистов Олександр

Грибовський Олег

Кошмарик Євган

Вареник Ілля

Документацію підготовлено: 02.02.2016

GO IT JAVA Online

2016

# ЗМІСТ

[ЗМІСТ](#h.207a43j8ihfp)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2

[Анотація](#h.m81n6iccxag)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

[Про авторів](#h.nsjbgv5ajxg8)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4

[Теорія](#h.wt18tifax7zi)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_5

[UML](#h.hpw3rspjp76l) Діаграма\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_7

[Клас Node](#h.nnr5agapyjb)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_8

[Клас Tree](#h.jr2kb0emh7h7)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_9

# Анотація

Пакет призначений для роботи з бінарними деревами. Реалізовано на мові програмування JAVA версії 8. Середа розробки - Intellij IDEA.

Основні задачі пакета: реалізувати навики програмування отримані під час проходження курсу JAVA Core.

# 

# Про авторів

Анна Смірнова

Програміст з великої букви Пі! Автор проектів “Воно шось не робить” та “Хочеш програмувати - не питай мене як?”. Розробник алгоритму “Що треба нажати, щоб воно заробило”.

Пясецький Дмитро

Потомственний віщун багів. Установить віндовс по фотогравфії, проведе відворот від вірусів, приворот безкоштовного інтернет трафіка, відремонтує будь-який мопед. Звертайтесь за телефом 3222-32-3222.

Подгорній Іван

Гудзуляк Богдан

Василистов Олександр

Грибовський Олег

Кошмарик Євган

Вареник Ілля

**Теорія**

Бінарне дерево - це структура даних, яка має корінь і певну кількість синів, які теж можуть мати своїх синів і т.д. Існує кілька термінів, які необхідно розглянути перед тим, як продовжити:

* Батько - це прямий попередник вузла дерева.
* Син - прямий спадкоємець вузла дерева.
* Попередник
* Спадкоємець

Бінарні дерева мають такі характеристики:

* Кожен вузол має по два сина.
* Вартість лівого сина менше вартості батька.
* Вартість правого сина більше вартості батька.

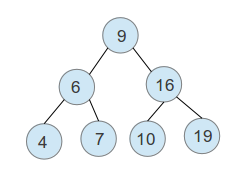


Рисунок 1. Збалансоване бінарне дерево

Дерево на цьому рисунку збалансовано, але це не є жорсткою вимогою для бінарних дерев. Хоча це бажана характеристика, бо дозволяє алгоритмам, базуватися на деревах, показувати оптимальну швидкість.

Операції над бінарними деревами

Знову ж таки є кілька основних операцій над бінарними деревами.

1. вставка
2. видалення
3. пошук
4. обхід

Обхід дерева

Обхід дерева, це процес знаходження всіх вузлів і їх оброблення. Існує три типи обходу бінарного дерева:

Inorder (зворотній) - відвідування лівого піддерева, кореня, правого піддерева.

Preorder (прямий) - відвідування кореня, лівого піддерева, правого піддерева.

Postorder (симетричний) - відвідування лівого піддерева, правого піддерева, кореня.

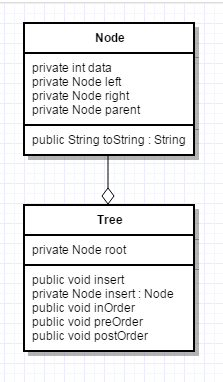
Ось як виглядає вихід для кожної з цих операцій на дереві, показаному на рисунку 1:

InOrder - 4, 6, 7, 9, 10, 16, 19. Як бачите, вартості сортовані. Це дуже зручна характеристика бінарного дерева.

PreOrder - 9, 6, 4, 7, 16, 10, 19.

PostOrder - 4, 7, 6, 10, 19, 16, 9.

# UML Діаграма



# Клас Node

Визначає основні поля і методи вузла (сина).

### Поля(властивості):

**private int data –** інформаційне поле;

**private Node left** – посилання на лівого сина;

**private Node right** – посилання на правого сина;

**private Node parent** – посилання на батька;

### Методи:

**public int[] getData()** - вертає числове значеннями інформації;

**public Node getLeft()** - вертає посилання на лівого сина;

**public Node getRight()** - вертає посилання на правого сина;

**public Node getParent()** - вертає посилання на батька;

**public void setData() -** записує інформаційну властивість;

**public void setLeft() -** записує посилання на лівого сина;

**public void setRight() -** записує посилання на правого сина;

**public void setParent() -** записує посилання на батька;

**public String toString()** - повертає опис основних характеристик вузла у вигляді текстової стрічки.

# Клас Tree

Описує дерево з його вузлами**.**

### Поля:

**private Node root** – зберігаеться основна інформація дерева;

### Конструктори:

**public Tree()** – створюється корінь дерева;

### Методи:

**public Node getRoot() –** повертає посилання на батька;

**public void insert()** – записується необхідна інформація в корінь дерева;

**private Node insert()** – записується необхідна інформація в вузол;

**public void inOrder()** – відбувається обхід дерева у зворотньому порядку;

**public void preOrder()** – відбувається обхід дерева у прямому порядку;

**public void postOrder()** – відбувається обхід у симетричному порядку;