**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Національний Технічний Університет України**

**«Київський Політехнічний Інститут»**

*Факультет інформатики та обчислювальної техніки*

*Кафедра обчислювальної техніки*

**Лабораторна робота №8**

*з дисципліни «Інженерія програмного забезпечення»*

*на тему: «Шаблони, що породжують. Шаблони Prototype, Singleton, Factory Method»*

**Виконав:**

студент 2-го курсу ФІОТ

групи ІО-44

*Барабаш Т.А.*

**Номер залікової книжки:** 4403

**Варіант:** 3

**Перевірив:**

Старший викладач

к.т.н, с.н.с.

*Антонюк А.І.*

**Київ – 2015**

**ЛАБОРАТОРНА РОБОТА №8**

*Шаблони, що породжують. Шаблони Prototype, Singleton, Factory Method*

**Мета:** Вивчення шаблонів, що породжують. Отримання базових навичок з застосування шаблонів Prototype, Singleton та Factory Method.

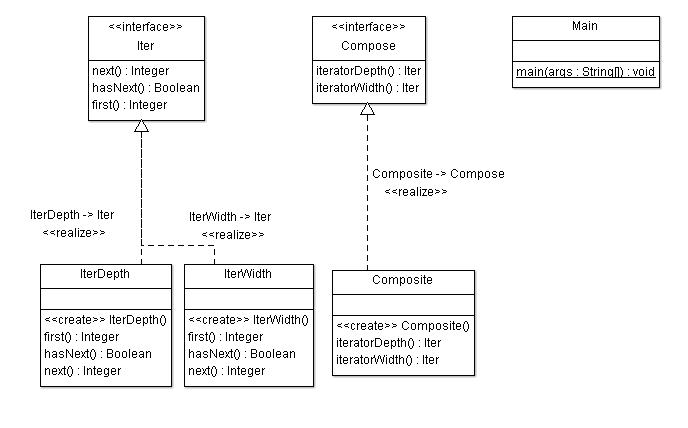
**I. Завдання**

**Визначення варіанту завдання:**

**Номер залікової книжки:** 4403;

**Варіант:** 4403 *mod* 11 = **3** – *Визначити специфікації класів для реалізації композиту та його ітераторів — для обходу структури методами пошуку в глибину (DFS) та ширину (BFS).*

**ІI. Діаграма класів**



**IIІ. Код програми**

**package com.lab111.labwork8;**

**/\*\***

**\* In this labwork I used pattern 'Factory method'**

**\***

**\* @author Taras Barabash**

**\* @version 8.10**

**\*/**

**public class Main {**

**public static void main(String[] args) {**

**Composite c = new Composite();**

**Iter i1 = c.iteratorDepth();**

**Iter i2 = c.iteratorWidth();**

**}**

**}**

**package com.lab111.labwork8;**

**/\*\***

**\* Defines interface for object.**

**\***

**\* @author Taras Barabash**

**\* @version 8.10**

**\*/**

**public interface Compose {**

**/\*\***

**\* Method of iterator by depth**

**\*/**

**public Iter iteratorDepth();**

**/\*\***

**\* Method of iterator by width**

**\*/**

**public Iter iteratorWidth();**

**}**

**package com.lab111.labwork8;**

**/\*\***

**\* Class Composite which implements Compose and has its methods and constructor.**

**\***

**\* @author Taras Barabash**

**\* @version 8.10**

**\*/**

**public class Composite implements Compose {**

**/\*\***

**\* Constructor**

**\*/**

**Composite() {**

**System.out.println("Realization of composite");**

**}**

**/\*\***

**\* Method of iterator by depth**

**\*/**

**public Iter iteratorDepth() {**

**System.out.println("Iterator by Depth");**

**return null;**

**}**

**/\*\***

**\* Method of iterator by width**

**\*/**

**public Iter iteratorWidth() {**

**System.out.println("Iterator by Width");**

**return null;**

**}**

**}**

**package com.lab111.labwork8;**

**/\*\***

**\* Defines interface for iterators.**

**\***

**\* @author Taras Barabash**

**\* @version 8.10**

**\*/**

**public interface Iter {**

**/\*\***

**\* Method that returns next element if it is**

**\*/**

**public Integer next();**

**/\*\***

**\* Method that define if there is next element**

**\*/**

**public Boolean hasNext();**

**/\*\***

**\* Method that returns first element**

**\*/**

**public Integer first();**

**}**

**package com.lab111.labwork8;**

**/\*\***

**\* Class IterWidth implements Iter, has constructor and specialization of method**

**\* first, hasNext,next.**

**\***

**\* @author Taras Barabash**

**\* @version 8.10**

**\*/**

**public class IterDepth implements Iter {**

**public IterDepth() {**

**System.out.println("Creation iterator by width");**

**}**

**/\*\***

**\* Method that returns first element**

**\*/**

**public Integer first() {**

**System.out.println("IteratorDepth.first");**

**return 0;**

**}**

**/\*\***

**\* Method that define if there is next element**

**\*/**

**public Boolean hasNext() {**

**System.out.println("IteratorDepth.hasNext");**

**return null;**

**}**

**/\*\***

**\* Method that returns next element if it is**

**\*/**

**public Integer next() {**

**System.out.println("IteratorDepth.next");**

**return null;**

**}**

**}**

**package com.lab111.labwork8;**

**/\*\***

**\* Class IterWidth implements Iter, has constructor and specialization of method**

**\* first, hasNext,next.**

**\***

**\* @author Taras Barabash**

**\* @version 8.10**

**\*/**

**public class IterWidth implements Iter {**

**/\*\***

**\* Constructor**

**\*/**

**public IterWidth() {**

**System.out.println("Creation iterator by width");**

**}**

**/\*\***

**\* Method that returns first element**

**\*/**

**public Integer first() {**

**System.out.println("IteratorWidth.first");**

**return 0;**

**}**

**/\*\***

**\* Method that define if there is next element**

**\*/**

**public Boolean hasNext() {**

**System.out.println("IteratorWidth.hasNext");**

**return null;**

**}**

**/\*\***

**\* Method that returns next element if it is**

**\*/**

**public Integer next() {**

**System.out.println("IteratorWidth.next");**

**return null;**

**}**

**}**

**ІV. Висновок**

Ознайомились з видами шаблонів проектування програмного забезпечення. Розроблена відповідна тестова програма. Результати успішної роботи тестової програми наведені нижче підтверджують правильність обраних рішень.

Realization of composite

Iterator by Depth

Iterator by Width