Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

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 курсу

ФІОТ гр. ІО-81

Дудка Максим

Залікова книжка №8106

ПЕРЕВІРИВ:

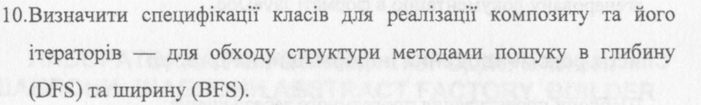
Старший викладач

к.т.н., с.н.с.

Антонюк А.І.

Київ 2019 р.

1. **Завдання**



1. **Код програми**

package com.lab111.labwork8;  
*/\*\*  
 \* In this labwork I used pattern 'Factory method'  
 \*  
 \** ***@author*** *Dudka Maxym  
 \** ***@version*** *12.0.1  
 \*/*public class Main {  
  
 public static void main(String[] args) {  
 Composite c = new Composite();  
 Iter i1 = c.iteratorDepth();  
 Iter i2 = c.iteratorWidth();  
  
 }  
  
}

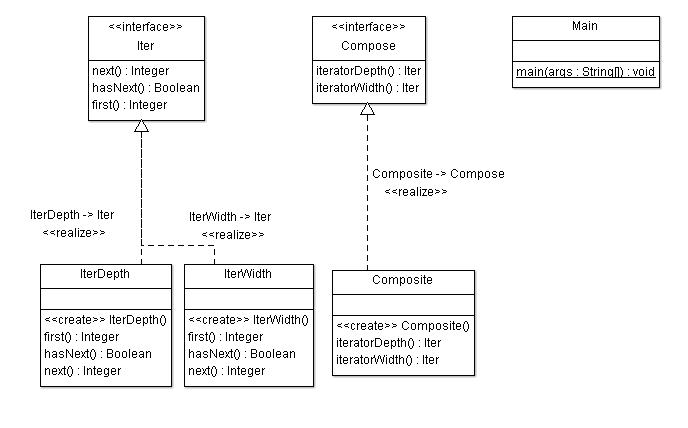
package com.lab111.labwork8;  
  
*/\*\*  
 \* Defines interface for object.  
 \*  
 \** ***@author*** *Dudka Maxym  
 \** ***@version*** *12.0.1  
 \*/*public interface Compose {  
 */\*\*  
 \* Method of iterator by depth  
 \*/* public Iter iteratorDepth();  
  
 */\*\*  
 \* Method of iterator by width  
 \*/* public Iter iteratorWidth();  
}package com.lab111.labwork8;  
  
*/\*\*  
 \* Class Composite which implements Compose and has its methods and constructor.  
 \*  
 \** ***@author*** *Dudka Maxym  
 \** ***@version*** *12.0.1  
 \*/*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*** *Dudka Maxym  
 \** ***@version*** *12.0.1  
 \*/*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();  
  
}

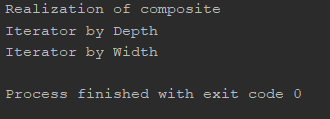
package com.lab111.labwork8;  
  
*/\*\*  
 \* Class IterWidth implements Iter, has constructor and specialization of method  
 \* first, hasNext,next.  
 \*  
 \** ***@author*** *Dudka Maxym  
 \** ***@version*** *12.0.1  
 \*/*public class IterDebth implements Iter {  
 public IterDebth() {  
 System.*out*.println("Creation 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*** *Dudka Maxym  
 \** ***@version*** *12.0.1  
 \*/*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;  
 }  
}

**IIІ. Діаграма класів**

**

**ІV. Вивід**



**V. Висновок**

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