**Методичні вказівки до лабораторних робіт з курсу**

**«Інженерія програмного забезпечення»**

**Лабораторна робота №9**

**Тема**: Шаблони, що породжують - 2. Шаблони Abstract Factory та Builder.

**Мета**: Вивчення шаблонів, що породжують. Отримання базових навичок з застосування шаблонів Abstract Factory та Builder.

**Завдання**

1. Повторити шаблони, що породжують. Знати загальну характеристику шаблонів, що породжують та призначення кожного з них.

2. Детально вивчити шаблони, що породжують - Abstract Factory та Builder. Для кожного з них:

* вивчити Шаблон, його призначення, альтернативні назви, мотивацію, випадки коли його застосування є доцільним та результати такого застосування;
* знати особливості реалізації Шаблону, споріднені шаблони, відомі випадки його застосування в програмних додатках;
* вільно володіти структурою Шаблону, призначенням його класів та відносинами між ними;
* вміти розпізнавати Шаблон в UML діаграмі класів та будувати сирцеві коди Java-класів, що реалізують шаблон.

3. В підготованому проекті (ЛР1) створити програмний пакет com.lab111.labwork9. В пакеті розробити інтерфейси і класи, що реалізують завдання (згідно варіанту) з застосуванням одного чи декількох шаблонів (п.2). В розроблюваних класах повністю реалізувати методи, пов'язані з функціюванням Шаблону. Методи, що реалізують бізнес-логіку закрити заглушками з виводом на консоль інформації про викликаний метод та його аргументи. Приклад реалізації бізнес-методу:

void draw(int x, int y){

System.out.println(“Метод draw з параметрами x=”+x+” y=”+y);

}

4. За допомогою автоматизованих засобів виконати повне документування розроблених класів (також методів і полів), при цьому документація має в достатній мірі висвітлювати роль певного класу в загальній структурі Шаблону та особливості конкретної реалізації.

**Варіанти** (№зк **mod** 11)

0. Визначити специфікації класів для подання сімейства віджетів графічного інтерфейсу користувача з реалізацією на різних API (WinAPI, GTK). Забезпечити можливість прозорого для клієнта розширення реалізацією для інших API (Qt, OSX).

1. Визначити специфікації класів для подання сімейства інструментів роботи з об'єктними даними через різні API (DB, File). Забезпечити можливість прозорого для клієнта розширення реалізацією для інших API (WebService).

2. Визначити специфікації класів для подання сімейства інструментів універсального інтерактивного середовища розробки (Validator, Compiler, Debugger) з їх реалізацією для різних мов (Java, C++). Забезпечити можливість прозорого для клієнта розширення реалізацією для мов (ObjectPascal).

3. Визначити специфікації класів для прямокутного ігрового простору та завантажувача його конфігурації із зовнішнього файлу.

4. Визначити специфікації класів для подання блок-схем алгоритмів (у відповідності до семантичної діаграми)

та її завантажувача із зовнішнього файлу.

5. Визначити специфікації класів для будівника дерева розбору складного виразу (у відповідності до БНФ) на основі його символьного подання.

<вираз>::=<простий вираз> | <складний вираз>

<простий вираз>::=<константа> | <змінна>

<константа>::=(<число>)

<змінна>::=(<ім’я>)

<складний вираз>::=(<вираз><знак операції><вираз>)

<знак операції>::=+|-|\*|/

6. Визначити специфікації класів для подання запису, реляційної таблиці та її завантажувача із зовнішнього файлу.

7. Визначити специфікації класів для подання реляційної таблиці та будівника прямого добутку таблиць.

8. Визначити специфікації класів для подання реляційної таблиці та будівника проекції таблиць.

9. Визначити специфікації класів для подання реляційної таблиці, схеми бази даних та відповідного завантажувача. Забезпечити можливість створення тільки одного примірника схеми бази даних.

10.  Визначити специфікації класів для подання елементів векторного графічного редактору (примітив і композит). Реалізувати можливість побудови композитного зображення на основі завантаженого файлу-специфікації.

**Протокол**

Протокол має містити титульну сторінку (з номером залікової книжки), завдання, роздруківку діаграми класів, розроблений сирцевий код та згенеровану документацію в форматі JavaDoc.

**Матеріали**

Підготовка до лабораторної роботи здійснюється за допомогою книги:

*Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес* Приемы объектно-ориентированного проектирования. Паттерны проектирования = Design Patterns: Elements of Reusable Object-Oriented Software. — СПб: [«Питер»](http://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D1%82%D0%B5%D1%80_%28%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE%29), 2007. — С. 366. — [ISBN 978-5-469-01136-1](http://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:BookSources/9785469011361) (также [ISBN 5-272-00355-1](http://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:BookSources/5272003551))

За необхідності додаткової інформації можливо використання матеріалів з мережі Інтернет, наприклад:

Шаблони проектування програмного забезпечення

* [Шаблони проектування програмного забезпечення](http://uk.wikipedia.org/wiki/%D0%A8%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D0%B8_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F)
* [Шаблон проектирования](http://ru.wikipedia.org/wiki/%D0%A8%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD_%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)
* [Обзор паттернов проектирования](http://citforum.ru/SE/project/pattern/)
* [Объектно-ориентированное проектирование, паттерны проектирования (Шаблоны)](http://www.javenue.info/themes/ood/)
* [David Gallardo. Шаблоны проектирования Java](http://khpi-iip.mipk.kharkiv.edu/library/extent/prog/jdp101/index.html)
* [Design pattern (computer science)](http://en.wikipedia.org/wiki/Design_pattern_%28computer_science%29)
* [Подготовка к собеседованию по Java/J2EE](http://cloud-cuckoo.com.ua/category/java/podgotovka-k-sobesedovaniyu-po-java/j2ee)

Шаблони, що породжують

* [Твірні шаблони](http://uk.wikipedia.org/wiki/%D0%A2%D0%B2%D1%96%D1%80%D0%BD%D1%96_%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D0%B8)
* [Creational pattern](http://en.wikipedia.org/wiki/Creational_pattern)
* [Шаблоны создания](http://khpi-iip.mipk.kharkiv.edu/library/extent/prog/jdp101/part4.html)