НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

КУРСОВА РОБОТА

з дисципліни: "Інженерія програмного забезпечення"

(назва дисципліни)

на тему: Система програмно-цільового підходу. Функціональність для аналітика

Студента 2 курсу ІО-22 групи

спеціальності комп’ютерна інженерія\_\_\_\_\_\_\_\_\_\_

Щербини Микити Валерійовича

(прізвище та ініціали)

Керівник Абу Усбах О.Н

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів:\_\_\_\_ Оцінка: ECTS\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

Київ - 2014 рік

Зміст

[ВСТУП 3](#_Toc386886466)

[1 ОГЛЯД ПАТЕРНУ MVC 4](#_Toc386886467)

[1.1 Мета шаблону 4](#_Toc386886468)

[1.2 Компоненти 4](#_Toc386886469)

[1.3 Особливості реалізації шаблону 6](#_Toc386886470)

[2 ПРОЕКТУВАННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА 8](#_Toc386886471)

[2.1 Діаграма прецедентів 8](#_Toc386886472)

[2.2 Діаграма граничних класів 11](#_Toc386886473)

[2.3 Макет зовнішнього вигляду програми 12](#_Toc386886474)

[2.4 Компоненти бібліотеки SWING, що були використані в програмі 15](#_Toc386886475)

[3 ДОКУМЕНТАЦІЯ 16](#_Toc386886476)

[3.1 Опис пакетів та класів 16](#_Toc386886477)

[3.2 Діаграма класів 61](#_Toc386886478)

[3.3 Інстукція користування програмним додатком 61](#_Toc386886479)

[4 ТЕСТУВАННЯ ПРОГРАМНОГО ДОДАТКУ 63](#_Toc386886480)

[ВИСНОВОК 67](#_Toc386886481)

[ЛІТЕРАТУРА 68](#_Toc386886482)

[ДОДАТКИ 69](#_Toc386886483)

[Додаток А. Код програми 69](#_Toc386886484)

# ВСТУП

Метою курсової роботи є закріплення теоретичних знань і отримання практичних навичок з проектування, моделювання, розробки та тестування програмного забезпечення, у використанні шаблонів проектування. Закріпити шаблон проектування MVC, багатопотоковість у Java, бібліотеку для створення графічного інтерфейсу на прикладі програми, що призначена для управління ієрархією задач.

## 1 ОГЛЯД ПАТЕРНУ MVC

## 1.1 Мета шаблону

Моде́ль-вид-контро́лер (MVC) — архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

Цей шаблон поділяє систему на три частини: модель даних, вигляд даних та керування. Застосовується для відокремлення даних (модель) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.

Мета шаблону — гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того використання цього шаблону у великих системах призводить до певної впорядкованості їх структури і робить їх зрозумілішими завдяки зменшенню складності.

## 1.2 Компоненти

Архітектурний шаблон Модель-Вид-Контролер (MVC) поділяє програму на три частини. У тріаді до обов'язків компоненту Модель (Model) входить зберігання даних і забезпечення інтерфейсу до них. Вигляд (View) відповідальний за представлення цих даних користувачеві. Контролер (Controller) керує компонентами, отримує сигнали у вигляді реакції на дії користувача, і повідомляє про зміни компоненту Модель. Така внутрішня структура в цілому поділяє систему на самостійні частини і розподіляє відповідальність між різними компонентами.

MVC поділяє цю частину системи на три самостійні частини: введення даних, компонент обробки даних і виведення інформації. Модель, як вже було відмічено, інкапсулює ядро даних і основний функціонал з їх обробки. Також компонент Модель не залежить від процесу введення або виведення даних. Компонент виводу Вигляд може мати декілька взаємопов'язаних областей, наприклад, різні таблиці і поля форм, в яких відображається інформація. У функції Контролера входить моніторинг за подіями, що виникають в результаті дій.

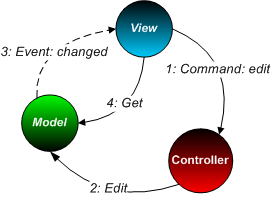


Рис.1.2.1 Взаємодія компонентів MVC

Зареєстровані події транслюються в різні запити, що спрямовуються компонентам Моделі або об'єктам, відповідальним за відображення даних. Відокремлення моделі від вигляду даних дозволяє незалежно використовувати різні компоненти для відображення інформації. Таким чином, якщо користувач через Контролер внесе зміни до Моделі даних, то інформація, подана одним або декількома візуальними компонентами, буде автоматично відкоригована відповідно до змін, що відбулися [1].

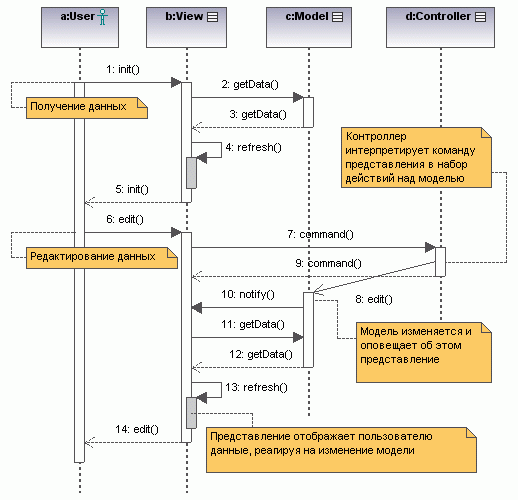


Рис. 1.2.2 Діаграма послідовності роботи MVC

## 1.3 Особливості реалізації шаблону

Для реалізації Model-View-Controller використовується велика кількість шаблонів проектування (в залежності від складності архітектурного вирішення). Основними шаблонами є Observer, Strategy та Composite.

Основною реалізацією є відокремлення *вигляд*у від моделі шляхом встановлення між ними протоколу взаємодії, використовуючи апарат подій (підписка/сповіщення). При кожній зміні внутрішніх даних у моделі вона сповіщає усі представлення, які від неї залежать. Ті ж у свою чергу оновлюються. Для виконання таких операцій використовується шаблон проектування Observer. При обробці реакції користувача *вигляд* обирає, залежно від потрібної реакції, потрібний *контролер*, який забезпечить той чи інший зв’язок з моделлю. Для цього використовується шаблон Strategy (також, при певній модифікації, можна використовувати шаблон Command). Для можливості однотипного звернення з підоб’єктами складного ієрархічного *вигляд*у може використовуватись шаблон Composite. Крім того, також можуть використовуватись інші шаблони проектування, наприклад, Factory Method, який дозволить задати за замовчуванням тип *контролер*у для відповідного *вигляд*у.

MVC задає не стільки правила поділу програми на окремі компоненти, скільки правила їх взаємодії.

Отже, вибір та використання шаблонів проектування, під час реалізації архітектурного шаблону MVC, повністю залежить від розробника.

Тому в різноманітних реалізаціях цього шаблону є певні відступи від оригінального. По-перше, відсутній чіткий поділ між *вигляд*ом та *контролер*ом. По-друге, деякі реалізації не потребують виділення бізнес-логіки програми в окремий компонент – *модель*. Програміст самостійно приймає рішення : витрачати зусилля на виділення в програмі моделі або ж відразу описувати у *контролер*і обробку даних, наприклад, звернення до бази даних. Проте таке рішення призведе до проблеми у проектуванні, для боротьби з яким і був створений шаблон MVC: якщо не виділити окрему сутність – *модель*, то уникнути залежності бізнес-логіки від *контролеру* неможливо. Даний вибір реалізації частіше всього помилково обирають недосвідчені програмісти. Хоча певні відхилення від оригіналу не є порушенням основної ідеї шаблону.

Бізнес-модель додатків, майже завжди, складається більше ніж з одного класу. Тому можлива реалізація спискової моделі. Описується загальний інтерфейс моделі, який дає можливість додавати/видаляти та сповіщати підписників. Всі класи, що належать до модель реалізують цей інтерфейс.

Створюється додатковий клас СписокМоделей, який реалізує інтерфейс простої моделі та має додаткові методи для пересилання повідомлення, створення та управлінням колекцією з простих моделей. Така архітектура дозволяє працювати з багатьма моделями як з однією. Можливість створювати композитні структури з моделей покращить повторну використовуванність програмного коду завдяки незалежності однієї групи моделей від інших[2].

## 2 ПРОЕКТУВАННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА

## 2.1 Діаграма прецедентів

В діаграмі прецедентів наведено порядок взаємодії користувача з програмним додатком.



Рис.2.1 Діаграма прецедентів

При відкритті програми,відкривається 2 вікна: «Вікно задач» та «Вікно груп».

Задопомогою головного меню користувач може переходити від одного вікна до іншого.



Рис.2.2 Прецендент переходу до вікна управління групами



Рис.2.3 Прецендент переходу до вікна управління задачами

Завершення програми відбувається після натиснення кнопки «Вихід» головного меню, або при закритті обох вікон.



Рис.2.4 Прецендент завершення програми натисненням кнопки головного меню

У вікні управління групами взаємодія користувача з програмою відбувається через список груп.

При виділенні групи у списку відображається повна інформація про групу.

Всі зміни відбуваються через контесне меню.



Рис.2.5 Прецендент виклику контесного меню списку груп

Для видалення виділеної групи необхідно з контесного меню вибрати пунт «Видалити групу» та підтвердити видалення.



Рис.2.6 Прецендент видалення виділеної групи

Для редагування виділеної групи необхідно з контесного меню вибрати пункт «Редагувати групу», відкриється вікно редагування. Після внесення змін необхідно підтвердити зміни натиснувши кнопку «Ок».



Рис.2.7 Прецендент редагуванння виділеної групи

Для створення нової групи необхідно з контесного меню вибрати пункт «Додати групу», відкриється вікно редагування новох групи. Після внесення змін необхідно підтвердити зміни натиснувши кнопку «Ок».



Рис.2.8 Прецендент створення нової групи

У вікні управління задачами взаємодія користувача з програмою відбувається через дерево задач.

При виділенні задачі у дереві відображаються під задачі та повна інформація про задачу.

Всі зміни відбуваються через контесне меню дерева задач.



Рис.2.9 Прецендент виклику контесного меню дерева задач

Для видалення виділеної задачі необхідно з контесного меню вибрати пунт «Видалити задачі» та підтвердити видалення.При видаленні задачі, видаляються й всі її під задачі.



Рис.2.10 Прецендент видалення виділеної задачі

Для редагування виділеної задачі необхідно з контесного меню вибрати пункт «Редагувати задачу», відкриється вікно редагування. Після внесення змін необхідно підтвердити зміни натиснувши кнопку «Ок».



Рис.2.11 Прецендент редагуванння виділеної задачі

Для створення нової задачі необхідно з контесного меню вибрати пункт «Додати задачу», відкриється вікно редагування новох групи. Після внесення змін необхідно підтвердити зміни натиснувши кнопку «Ок».



Рис.2.12 Прецендент створення нової задачі

## 2.2 Діаграма граничних класів

В наведеній нижче діаграмі показані класи з якими взаємодіє користувач.



Рис. 2.13 Діаграма граничних класів

## 2.3 Макет зовнішнього вигляду програми

Вікно для роботи з задачами дає можливість переглядати ієрархію задач, дивитися докладну інформацію про конкретну задачу, видаляти задачі.



Рис. 2.14 Макет вікна «Задачі»

Вікно «Редагування задачі» дає можливість редагувати задачу. Для збереження змін потрібно натиснути кнонку «ОК».



Рис. 2.15 Макет вікна «Редагування задачі»

Вікно для роботи з групами дає можливість переглядати список наявних груп, дізнатися додаткову інформацію про групу, а також видаляти групи.



Рис. 2.16 Макет вікна «Групи»

Вікно «Редагування групи» дає можливість редагувати групу. Для збереження змін потрібно натиснути кнонку «ОК».



Рис. 2.17 Макет вікна «Редагування групи»

## 2.4 Компоненти бібліотеки SWING, що були використані в програмі

В наведеній нижче таблиці описується відповідність між сутностями діаграми прецидентів та бібліотечними класами.

таблиця 2.1 відповідність між сутностями та класами

|  |  |
| --- | --- |
| Вікно «Задачі» | javax.swing.JFrame |
| Вікно «Групи» | javax.swing.JFrame |
| Головне меню | javax.swing.JMenuBar |
| Елемент меню «Перейти до груп» | javax.swing.JMenuItem |
| Елемент меню «Перейти до задач» | javax.swing.JMenuItem |
| Елемент меню «Вихід» | javax.swing.JMenuItem |
| Панель показу групи | javax.swing.JPanel |
| Панель показу задачі | javax.swing.JPanel |
| Дерево задач | javax.swing.JTree |
| Список груп | javax.swing.JTable |
| Текстове поле «Назва задачі» | javax.swing.JTextField |
| Текстове поле «Завдання» у вікні Задачі | javax.swing.JTextArea |
| Текстове поле «Необхідні ресурси» | javax.swing.JTextArea |
| Текстове поле «Назва групи» | javax.swing.JTextField |
| Текстове поле «Завдання» у вікні Груп | javax.swing.JTextArea |
| Склад групи | javax.swing.JTable |
| Контексте меню задач | javax.swing.JPopupMenu |
| Контексте меню груп | javax.swing.JPopupMenu |
| Елемент меню «Додати задачу» | javax.swing.JMenuItem |
| Елемент меню «Видалити задачу» | javax.swing.JMenuItem |
| Елемент меню «Редагувати задачу» | javax.swing.JMenuItem |
| Елемент меню «Додати групу» | javax.swing.JMenuItem |
| Елемент меню «Редагувати групу» | javax.swing.JMenuItem |
| Елемент меню «Видалити групу» | javax.swing.JMenuItem |

# 3 ДОКУМЕНТАЦІЯ

## 3.1 Опис пакетів та класів

|  |  |
| --- | --- |
| **Пакети** | |
| **Пакет** | **Опис** |
| [**configuration**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\package-summary.html) | Конфігурація програми |
| [**configuration.utils**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\package-summary.html) | Інтерфейси для використання конфігурації |
| [**configuration.utils.xml**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\xml\package-summary.html) | Реалізації класів для обробки конфігурації у форматі XML |
| [**core.controller**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\package-summary.html) | Контроллер паттерну MVC |
| [**core.controller.annotation**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\package-summary.html) | Інтерфейси анотації, які використовуються контроллером |
| [**core.controller.annotation.utils**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\utils\package-summary.html) | Утиліти для обробки аннотацій |
| [**core.controller.Commands**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\package-summary.html) | Інтерфейс опису команди |
| [**core.controller.Commands.AllCommands**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\AllCommands\package-summary.html) | Команди, які будуть виконуватися контроллером |
| [**core.controller.exception**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\exception\package-summary.html) | Виключні ситуації при роботі контроллера |
| [**dao**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\package-summary.html) | Шаблон Data Access Object |
| [**dao.annotation**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\annotation\package-summary.html) | Аннотації для розмітки класів,що будут оброблюватися DAO |
| [**dao.annotation.utils**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\annotation\utils\package-summary.html) | Утилиты для работы с аннотациями |
| [**dao.annotation.utils.converter**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\annotation\utils\converter\package-summary.html) | Конверторы типов данных модели для DAO |
| [**dao.filters**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\filters\package-summary.html) | Фільтри для роботи з DAO |
| [**initialization**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\initialization\package-summary.html) | Містить класс , що реалізує запуск,ініциалізацію та завершення програми |
| [**model**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\package-summary.html) | Модель даних прикладної області |
| [**model.event**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\event\package-summary.html) | Події моделі |
| [**view**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\view\package-summary.html) | Графічний інтерфейс користувача |

## 

Пакет configuration

Конфигурация приложения

See: [Опис](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\package-summary.html#package_description)

|  |  |
| --- | --- |
| **Основні інтерфейси** | |
| **Interface** | **Опис** |
| [**IConfiguration**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html) | Интерфейс конфигурации |

|  |  |
| --- | --- |
| **Основні класи** | |
| **Class** | **Опис** |
| [**Configuration**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\Configuration.html) | Конфигурация |

|  |  |
| --- | --- |
| **Enum Summary** | |
| **Enum** | **Опис** |
| [**ApplicationConfiguration**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\ApplicationConfiguration.html) | Конфигурация приложения |

**Interface IConfiguration**

**All Known Implementing Classes:**

[ApplicationConfiguration](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\ApplicationConfiguration.html), [Configuration](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\Configuration.html)

public interface **IConfiguration**

Интерфейс конфигурации

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| java.util.List<java.lang.String> | [**getAllKey**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html#getAllKey())()  Возращает все ключи |
| java.lang.String | [**getValue**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html#getValue(java.lang.String))(java.lang.String key)  Возращает значение по ключу |
| void | [**setValue**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html#setValue(java.lang.String, java.lang.String))(java.lang.String key, java.lang.String Value)  Возвращает зачение по ключу |

Method Detail

#### setValue

* + - void setValue(java.lang.String key, java.lang.String Value)

Возвращает зачение по ключу

**Parameters:**

key -ключ

Value -значение

#### getValue

java.lang.String getValue(java.lang.String key)

Возращает значение по ключу

**Parameters:**

key -ключ

**Returns:** значение

#### getAllKey

java.util.List<java.lang.String> getAllKey()

Возращает все ключи

**Returns:**

- все ключи

**Class Configuration**

* java.lang.Object
  + configuration.Configuration
* **All Implemented Interfaces:**

[IConfiguration](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html)

public class **Configuration**

extends java.lang.Object

implements [IConfiguration](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html)

Конфигурация

Constructor Summary

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**Configuration**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\Configuration.html#Configuration())() |

Method Summary

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| java.util.List<java.lang.String> | [**getAllKey**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\Configuration.html#getAllKey())()  Возращает все ключи |
| java.lang.String | [**getValue**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\Configuration.html#getValue(java.lang.String))(java.lang.String key)  Возращает значение по ключу |
| void | [**setValue**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\Configuration.html#setValue(java.lang.String, java.lang.String))(java.lang.String key, java.lang.String value)  Возвращает зачение по ключу |

Constructor Detail

#### Configuration

public Configuration()

**Enum ApplicationConfiguration**

* java.lang.Object
  + java.lang.Enum<[ApplicationConfiguration](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\ApplicationConfiguration.html)>
    - configuration.ApplicationConfiguration
* **All Implemented Interfaces:**

[IConfiguration](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html), java.io.Serializable, java.lang.Comparable<[ApplicationConfiguration](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\ApplicationConfiguration.html)>

public enum **ApplicationConfiguration**

extends java.lang.Enum<[ApplicationConfiguration](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\ApplicationConfiguration.html)>

implements [IConfiguration](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html)

Конфигурация приложения

Enum Constant Summary

|  |  |
| --- | --- |
| **Enum Constants** | |
| **Enum Constant and Опис** | |
| [**INSTANCE**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\ApplicationConfiguration.html#INSTANCE) | |
| **Методи** | |
| **Тип** | **Опис методів** |
| java.util.List<java.lang.String> | [**getAllKey**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\ApplicationConfiguration.html#getAllKey())()  Возращает все ключи |
| java.lang.String | [**getValue**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\ApplicationConfiguration.html#getValue(java.lang.String))(java.lang.String key)  Возращает значение по ключу |
| void | [**setValue**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\ApplicationConfiguration.html#setValue(java.lang.String, java.lang.String))(java.lang.String key, java.lang.String value)  Возвращает зачение по ключу |

**Пакет configuration.utils**

Интерфейс для работы с конфигурацией

|  |  |
| --- | --- |
| **Основні класи** | |
| **Класи** | **Опис** |
| [**Reader**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\Reader.html) | Інтерфейс для считуванння конфігурації |
| [**Writer**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\Writer.html) | Інтерфейс для запису конфінупації |

**Class Reader**

* java.lang.Object
  + configuration.utils.Reader

public abstract class **Reader**

extends java.lang.Object

Интерфейс для заполнения конфигурации

Constructor Summary

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**Reader**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\Reader.html#Reader())() |

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| abstract void | [**load**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\Reader.html#load(java.lang.String))(java.lang.String path)  читає конфігурацію |
| void | [**setConfiguration**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\Reader.html#setConfiguration(configuration.IConfiguration))([**IConfiguration**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html) configuration)  встановлює обьект конфігурації в який буде відбуватися читання |

Constructor Detail

Reader

public Reader()

Method Detail

#### load

public abstract void load(java.lang.String path)

читає конфігурацію

**Parameters:**

path - адреса файлу з конфігурацією

#### setConfiguration

public void setConfiguration([IConfiguration](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html) configuration)

встановлює обьект конфігурації в який буде відбуватися читання

**Parameters:**

configuration -

**Class Writer**

* java.lang.Object
  + configuration.utils.Writer
* **Direct Known Subclasses:**

[XMLWriter](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\xml\XMLWriter.html)

public abstract class **Writer**

extends java.lang.Object

Интерфейс для записи конфигурации

Constructor Summary

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**Writer**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\Writer.html#Writer())() |

Method Summary

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| abstract void | [**save**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\Writer.html#save(java.lang.String))(java.lang.String path)  зберігає конфігурацію у файл |
| void | [**setConfiguration**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\Writer.html#setConfiguration(configuration.IConfiguration))([**IConfiguration**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html) configuration)  встановлює конфігурацію яку необходно зберегти |

Constructor Detail

#### Writer

public Writer()

Method Detail

#### save

* + - public abstract void save(java.lang.String path)
    - throws java.io.IOException,

javax.xml.parsers.ParserConfigurationException

зберігає конфігурацію у файл

**Parameters:**

path - адреса файлу

**Throws:**

java.io.IOException

javax.xml.parsers.ParserConfigurationException

#### setConfiguration

public void setConfiguration([IConfiguration](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html) configuration)

встановлює конфігурацію яку необходно зберегти

**Parameters:**

configuration - конфігурація

**Пакет configuration.utils.xml**

Утилиты для работы с конфигурацией в формате XML

|  |  |
| --- | --- |
| **Основні класи** | |
| **Class** | **Опис** |
| [**XMLReader**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\xml\XMLReader.html) | читає XML файли |
| [**XMLWriter**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\xml\XMLWriter.html) | Записує конфігурацію у XML |

Class XMLReader

* java.lang.Object
  + [configuration.utils.Reader](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\Reader.html)
    - configuration.utils.xml.XMLReader

public class **XMLReader**

extends [Reader](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\Reader.html)

читає XML файли

Constructor Summary

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**XMLReader**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\xml\XMLReader.html#XMLReader(configuration.IConfiguration))([**IConfiguration**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html) configuration) |

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| void | [**load**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\xml\XMLReader.html#load(java.lang.String))(java.lang.String path)  читає конфігурацію |

XMLReader

public XMLReader([IConfiguration](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html) configuration)

**Class XMLWriter**

* java.lang.Object
  + [configuration.utils.Writer](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\Writer.html)
    - configuration.utils.xml.XMLWriter

public class **XMLWriter**

extends [Writer](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\Writer.html)

Записує конфігурацію у XML

Constructor Summary

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**XMLWriter**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\xml\XMLWriter.html#XMLWriter(configuration.IConfiguration))([**IConfiguration**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html) configuration) |

Method Summary

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| void | [**save**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\utils\xml\XMLWriter.html#save(java.lang.String))(java.lang.String path)  зберігає конфігурацію у файл |

#### XMLWriter

public XMLWriter([IConfiguration](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html) configuration)

**Пакет core.controller**

Контроллер паттерна MVC

|  |  |
| --- | --- |
| **Enum Summary** | |
| **Enum** | **Опис** |
| [**Controller**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Controller.html) | Принимает и выполянет команды переданые от Viev |

Enum Controller

* java.lang.Object
  + java.lang.Enum<[Controller](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Controller.html)>
    - core.controller.Controller
* **All Implemented Interfaces:**

java.io.Serializable, java.lang.Comparable<[Controller](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Controller.html)>, java.lang.Runnable

public enum **Controller**

extends java.lang.Enum<[Controller](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Controller.html)>

implements java.lang.Runnable

Принимает и выполянет команды переданые от Viev

Enum Constant Summary

|  |
| --- |
| **Enum Constants** |
| **Enum Constant and Опис** |
| [**INSTANCE**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Controller.html#INSTANCE)  Обєкт контроллера |

Method Summary

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| void | [**addCommand**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Controller.html#addCommand(core.controller.Commands.Command))([**Command**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Command.html) newCommand)  Добавляет новую команду |
| void | [**run**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Controller.html#run())() |

#### INSTANCE

public static final [Controller](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Controller.html) INSTANCE

Обєкт контроллера

addCommand

public void addCommand([Command](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Command.html) newCommand)

Добавляет новую команду

**Parameters:**

newCommand - новая команда

#### run

public void run()

**Specified by:**

run in interface java.lang.Runnable

**Пакет core.controller.annotation**

Интерфейс аннотаций описания команды

|  |  |
| --- | --- |
| **Annotation Types Summary** | |
| **Annotation Type** | **Опис** |
| [**COMMAND**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\COMMAND.html) |  |
| [**CONTEXT**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\CONTEXT.html) |  |
| [**PARAMETER**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\PARAMETER.html) |  |

Annotation Type COMMAND

@Retention(value=RUNTIME)

@Target(value=TYPE)

public @interface **COMMAND**

**Required** Element Summary

|  |  |
| --- | --- |
| **Required Elements** | |
| **Тип** | **Required Element and Опис** |
| java.lang.String | [**key**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\COMMAND.html#key()) |

**Element** Detail

#### key

public abstract java.lang.String key

Annotation Type CONTEXT

@Retention(value=RUNTIME)

@Target(value=TYPE)

public @interface **CONTEXT**

**Required** Element Summary

|  |  |
| --- | --- |
| **Required Elements** | |
| **Тип** | **Required Element and Опис** |
| [**PARAMETER**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\PARAMETER.html)[] | [**list**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\CONTEXT.html#list()) |

**Element** Detail

#### list

public abstract [PARAMETER](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\PARAMETER.html)[] list

Annotation Type PARAMETER

@Retention(value=RUNTIME)

@Target(value=TYPE)

public @interface **PARAMETER**

**Required** Element Summary

|  |  |
| --- | --- |
| **Required Elements** | |
| **Тип** | **Required Element and Опис** |
| java.lang.String | [**key**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\PARAMETER.html#key()) |

Optional **Element** Summary

|  |  |
| --- | --- |
| **Optional Elements** | |
| **Тип** | **Optional Element and Опис** |
| boolean | [**optional**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\PARAMETER.html#optional()) |
| java.lang.Class<?> | [**type**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\PARAMETER.html#type()) |

**Element** Detail

#### key

public abstract java.lang.String key

#### type

public abstract java.lang.Class<?> type

**Default:**

java.lang.Object.class

optional

public abstract boolean optional

**Default:**

false

**Пакет core.controller.annotation.utils**

Утилиты для разбора аннотаций в командах контроллера

|  |  |
| --- | --- |
| **Основні класи** | |
| **Class** | **Опис** |
| [**ControllerAnnotationUtils**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\utils\ControllerAnnotationUtils.html) | Клас для роботи з аннотацією Команди |
| [**Validator**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\utils\Validator.html) | Перевіряє правильність виклику команди |

**Class ControllerAnnotationUtils**

* java.lang.Object
  + core.controller.annotation.utils.ControllerAnnotationUtils

public class **ControllerAnnotationUtils**

extends java.lang.Object

Клас для роботи з аннотацією Команди

Constructor Summary

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**ControllerAnnotationUtils**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\utils\ControllerAnnotationUtils.html#ControllerAnnotationUtils())() |

Method Summary

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| static java.lang.String | [**getCommandKey**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\utils\ControllerAnnotationUtils.html#getCommandKey(java.lang.Class))(java.lang.Class c)  Повертає назву команди |
| static [**PARAMETER**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\PARAMETER.html)[] | [**getParameterList**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\utils\ControllerAnnotationUtils.html#getParameterList(java.lang.Class))(java.lang.Class c)  Повертає параметри классу Команди |
| static boolean | [**paramIsOptional**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\utils\ControllerAnnotationUtils.html#paramIsOptional(java.lang.String, java.lang.Class))(java.lang.String paramKey, java.lang.Class command)  Повертає параметр обовязковості параметру |

#### ControllerAnnotationUtils

public ControllerAnnotationUtils()

Method Detail

getCommandKey

public static java.lang.String getCommandKey(java.lang.Class c)

Повертає назву команди

**Parameters:**

c - класс команди

**Returns:**

- назва команди

#### getParameterList

public static [PARAMETER](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\PARAMETER.html)[] getParameterList(java.lang.Class c)

Повертає параметри классу Команди

**Parameters:**

c - класс Команди для якого потрібно визначити параметри

**Returns:**

- параметри з якими потрібно виконувати команду

#### paramIsOptional

* + - public static boolean paramIsOptional(java.lang.String paramKey,

java.lang.Class command)

Повертає параметр обовязковості параметру

**Parameters:**

paramKey - ім'я параметру

command - класс команди

**Returns:**

- true якщо параметр необовязковий

**Class Validator**

* java.lang.Object
  + core.controller.annotation.utils.Validator

public class **Validator**

extends java.lang.Object

Перевіряє правильність виклику команди

Constructor Summary

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**Validator**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\utils\Validator.html#Validator())() |

Method Summary

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| boolean | [**checkCommand**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\annotation\utils\Validator.html#checkCommand(core.controller.Commands.Command))([**Command**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Command.html) comand)  Перевірити наявність параметрів команди |

Constructor Detail

#### Validator

public Validator()

Method Detail

checkCommand

public boolean checkCommand([Command](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Command.html) comand)

Перевірити наявність параметрів команди

**Parameters:**

comand - команда, що перевіряється

**Returns:**

- true - якщо правильні параметри

**Пакет core.controller.Commands**

Інтерфейс опису команди та парамтрів

|  |  |
| --- | --- |
| **Основні класи** | |
| **Класи** | **Опис** |
| [**Command**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Command.html) | Интерфейс команды |
| [**Context**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Context.html) | Описывет сигнатуру вызова метода |

**Пакет core.controller.Commands Опис**

Інтерфейс опису команди та парамтрів

**Class Command**

* java.lang.Object
  + core.controller.Commands.Command

public abstract class **Command**

extends java.lang.Object

Интерфейс команды

Constructor Summary

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**Command**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Command.html#Command())() |

Method Summary

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| void | [**addParameter**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Command.html#addParameter(java.lang.String, java.lang.Object))(java.lang.String key, java.lang.Object value)  Добавляет параметр к методу |
| void | [**deleteParametr**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Command.html#deleteParametr(java.lang.String))(java.lang.String key)  удаляет параметр по имени |
| abstract void | [**execute**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Command.html#execute())()  выполнить действие |
| [**Context**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Context.html) | [**getParameters**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Command.html#getParameters())()  Возращает параметры метода |
| void | [**setParameters**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Command.html#setParameters(core.controller.Commands.Context))([**Context**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Context.html) parameters)  задает параметры метода |

Command

public Command()

Method Detail

#### execute

public abstract void execute()

выполнить действие

#### setParameters

public void setParameters([Context](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Context.html) parameters)

задает параметры метода

**Parameters:**

parameters - параметры

#### getParameters

public [Context](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Context.html) getParameters()

Возращает параметры метода

**Returns:**

- параметры

#### addParameter

* + - public void addParameter(java.lang.String key, java.lang.Object value)

Добавляет параметр к методу

**Parameters:**

key - название параметра

value - значение

#### deleteParametr

public void deleteParametr(java.lang.String key)

удаляет параметр по имени

**Parameters:**

key - название параметра

**Class Context**

* java.lang.Object
  + java.util.AbstractMap<K,V>
    - java.util.HashMap<java.lang.String,java.lang.Object>
      * core.controller.Commands.Context

Описывет сигнатуру вызова метода

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**Context**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Context.html#Context())() |

Constructor Detail

#### Context

public Context()

**Пакет core.controller.Commands.AllCommands**

Команды выполняемые контроллером

|  |  |
| --- | --- |
| **Основні класи** | |
| **Класи** | **Опис** |
| [**Command1**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\AllCommands\Command1.html) |  |

**Пакет core.controller.exception**

Виключні ситуації при роботі з контроллером

|  |  |
| --- | --- |
| **Exception Summary** | |
| **Exception** | **Опис** |
| [**IncorrectParameterListException**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\exception\IncorrectParameterListException.html) | Неверный список параметров |

Пакет core.controller.exception Опис

Виключні ситуації при роботі з контроллером

Class IncorrectParameterListException

* java.lang.Object
  + java.lang.Throwable
    - java.lang.Exception
      * core.controller.exception.IncorrectParameterListException
* **All Implemented Interfaces:**

java.io.Serializable

public class **IncorrectParameterListException**

extends java.lang.Exception

Неверный список параметров

Field Summary

|  |  |
| --- | --- |
| **Fields** | |
| **Тип** | **Опис** |
| [**Command**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Command.html) | [**command**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\exception\IncorrectParameterListException.html#command) |

Constructor Summary

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**IncorrectParameterListException**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\exception\IncorrectParameterListException.html#IncorrectParameterListException(core.controller.Commands.Command))([**Command**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Command.html) command) |

Method Summary

Field Detail

Command

public [Command](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Command.html) command

Constructor Detail

IncorrectParameterListException

public IncorrectParameterListException([Command](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\core\controller\Commands\Command.html) command)

**Пакет dao**

Шаблон Data Access Object

|  |  |
| --- | --- |
| **Основні інтерфейси** | |
| **Інтерфейси** | **Опис** |
| [**CRUDInterface**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html) | Описує інтерфейс доступу до сховища даних |
| [**DAOFilter**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\DAOFilter.html) | Інтерфейс фільтру обєктів сховища |

|  |  |
| --- | --- |
| **Основні класи** | |
| **Класи** | **Опис** |
| [**TransparentCRUD**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\TransparentCRUD.html) |  |

|  |  |
| --- | --- |
| **Enum Summary** | |
| **Enum** | **Опис** |
| [**Storage**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html) | Хранилище данных |

Interface CRUDInterface

* **All Known Implementing Classes:**

[Storage](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html), [TransparentCRUD](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\TransparentCRUD.html)

public interface **CRUDInterface**

Описує інтерфейс доступу до сховища даних

Method Summary

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| void | [**close**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html#close())()  Закриває сховище |
| <T> void | [**delete**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html#delete(T))(T instance)  видаляє обєкт зі сховища |
| <T> T | [**insert**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html#insert(T))(T instance)  додає обьект у сховище |
| void | [**open**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html#open(configuration.IConfiguration))([**IConfiguration**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html) configuration)  відкриває сховище |
| <T> T | [**read**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html#read(java.lang.Class, java.util.UUID))(java.lang.Class entityClass, java.util.UUID id)  повертає об'єкт з сховища за Id |
| <T> java.util.List<T> | [**select**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html#select(java.lang.Class, dao.DAOFilter))(java.lang.Class entityClass, [**DAOFilter**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\DAOFilter.html) filter)  Повертає список обєктів, що задовільняють умовам фільтру |
| <T> java.util.List<T> | [**select**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html#select(java.lang.String))(java.lang.String SQLString) |
| <T> void | [**update**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html#update(T))(T instance)  Оновлює об'єкт у сховищі |

open

* + - void open([IConfiguration](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html) configuration) throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

відкриває сховище

**Parameters:**

configuration - конфігурація з налаштуваннями для вікриття

**Throws:**

[DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

close

* + - void close() throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

Закриває сховище

**Throws:** [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

insert

* + - <T> T insert(T instance) throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

додає обьект у сховище

**Parameters:**

instance - об'єкт, який необхідно додати

**Returns:**

- додавємий об'єкт

**Throws:** [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

read

* + - <T> T read(java.lang.Class entityClass, java.util.UUID id) throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

повертає об'єкт з сховища за Id

**Parameters:**

entityClass - тип об'єкта

id - обєкта який потрібно

**Returns:**

- об'єкт зі сховища

**Throws:** [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

update

* + - <T> void update(T instance) throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

Оновлює об'єкт у сховищі

**Parameters:**

instance - об'єкт,що потрібно оновити

**Throws:** [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

delete

* + - <T> void delete(T instance) throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

видаляє обєкт зі сховища

**Parameters:**

instance - обєкт,що необхідно видалити

**Throws:** [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

select

* + - <T> java.util.List<T> select(java.lang.Class entityClass, [DAOFilter](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\DAOFilter.html) filter) throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

Повертає список обєктів, що задовільняють умовам фільтру

**Parameters:**

entityClass - тип об'єктів

filter -фільтр

**Returns:**

- спсикок об'єктів

**Throws:** [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

select

* + - <T> java.util.List<T> select(java.lang.String SQLString) throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

**Throws:** [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

**Interface DAOFilter**

* **All Known Implementing Classes:**

[AllObjectsOfClass](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\filters\AllObjectsOfClass.html), [GetChildrenTask](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\filters\GetChildrenTask.html)

public interface **DAOFilter**

Інтерфейс фільтру обєктів сховища

Method Summary

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| <T> boolean | [**accept**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\DAOFilter.html#accept(T))(T entity)  Задає умови фільтрування об'єктів |

accept

<T> boolean accept(T entity)

Задає умови фільтрування об'єктів

**Parameters:**

entity - об'єкт, що перевіряється

**Returns:**

-результат перевірки, true - якщо об'єкт відповідає умовам перевірки

**Class TransparentCRUD**

* java.lang.Object
  + dao.TransparentCRUD
* **All Implemented Interfaces:**

[CRUDInterface](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html)

public class **TransparentCRUD**

extends java.lang.Object

implements [CRUDInterface](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html)

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**TransparentCRUD**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\TransparentCRUD.html#TransparentCRUD())() |
| [**TransparentCRUD**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\TransparentCRUD.html#TransparentCRUD(dao.CRUDInterface))([**CRUDInterface**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html) nextCRUD) |

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| void | [**close**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\TransparentCRUD.html#close())()  Закриває сховище |
| <T> void | [**delete**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\TransparentCRUD.html#delete(T))(T instance)  видаляє обєкт зі сховища |
| [**CRUDInterface**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html) | [**getNextCRUD**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\TransparentCRUD.html#getNextCRUD())() |
| <T> T | [**insert**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\TransparentCRUD.html#insert(T))(T instance)  додає обьект у сховище |
| void | [**open**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\TransparentCRUD.html#open(configuration.IConfiguration))([**IConfiguration**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html) configuration)  відкриває сховище |
| <T> T | [**read**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\TransparentCRUD.html#read(java.lang.Class, java.util.UUID))(java.lang.Class entityClass, java.util.UUID id)  повертає об'єкт з сховища за Id |
| <T> java.util.List<T> | [**select**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\TransparentCRUD.html#select(java.lang.Class, dao.DAOFilter))(java.lang.Class entityClass, [**DAOFilter**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\DAOFilter.html) filter)  Повертає список обєктів, що задовільняють умовам фільтру |
| <T> java.util.List<T> | [**select**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\TransparentCRUD.html#select(java.lang.String))(java.lang.String SQLString) |
| void | [**setNextCRUD**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\TransparentCRUD.html#setNextCRUD(dao.TransparentCRUD))([**TransparentCRUD**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\TransparentCRUD.html) nextCRUD) |
| <T> void | [**update**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\TransparentCRUD.html#update(T))(T instance)  Оновлює об'єкт у сховищі |

TransparentCRUD

public TransparentCRUD()

TransparentCRUD

public TransparentCRUD([CRUDInterface](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html) nextCRUD)

Method Detail

getNextCRUD

public [CRUDInterface](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html) getNextCRUD()

setNextCRUD

public void setNextCRUD([TransparentCRUD](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\TransparentCRUD.html) nextCRUD)

insert

* + - public <T> T insert(T instance) throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

**Опис copied from interface:**[**CRUDInterface**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html#insert(T))

додає обьект у сховище

**Specified by:**

[insert](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html#insert(T)) in interface [CRUDInterface](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html)

**Parameters:**

instance - об'єкт, який необхідно додати

**Returns:**

- додавємий об'єкт

**Throws:** [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

read

* + - public <T> T read(java.lang.Class entityClass, java.util.UUID id) throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

**Опис copied from interface:**[**CRUDInterface**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html#read(java.lang.Class, java.util.UUID))

повертає об'єкт з сховища за Id

**Specified by:**

[read](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html#read(java.lang.Class, java.util.UUID)) in interface [CRUDInterface](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html)

**Parameters:**

entityClass - тип об'єкта

id - обєкта який потрібно

**Returns:**

- об'єкт зі сховища

**Throws:**

[DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

update

* + - public <T> void update(T instance) throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

**Опис copied from interface:**[**CRUDInterface**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html#update(T))

Оновлює об'єкт у сховищі

**Specified by:**

[update](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html#update(T)) in interface [CRUDInterface](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html)

**Parameters:**

instance - об'єкт,що потрібно оновити

**Throws:** [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

delete

* + - public <T> void delete(T instance) throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

**Опис copied from interface:**[**CRUDInterface**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html#delete(T))

видаляє обєкт зі сховища

**Specified by:**

[delete](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html#delete(T)) in interface [CRUDInterface](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html)

**Parameters:**

instance - обєкт,що необхідно видалити

**Throws:**[DAOException](file:///E:\\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\\%D0%9F%D0%9F%D0%97\\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\\program\\doc\\dao\\exception\\DAOException.html" \o "class in dao.exception)

select

* + - public <T> java.util.List<T> select(java.lang.Class entityClass, [DAOFilter](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\DAOFilter.html) filter) throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

Повертає список обєктів, що задовільняють умовам фільтру

**Specified by:**

[select](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html#select(java.lang.Class, dao.DAOFilter)) in interface [CRUDInterface](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html)

**Parameters:**

entityClass - тип об'єктів

filter -фільтр

**Returns:**  спсикок об'єктів

**Throws:**

[DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

select

* + - public <T> java.util.List<T> select(java.lang.String SQLString) throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

**Specified by:**[select](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html#select(java.lang.String)) in interface [CRUDInterface](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html)

**Throws:**[DAOException](file:///E:\\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\\%D0%9F%D0%9F%D0%97\\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\\program\\doc\\dao\\exception\\DAOException.html" \o "class in dao.exception)

open

* + - public void open([IConfiguration](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html) configuration) throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

відкриває сховище

**Specified by:**[open](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html#open(configuration.IConfiguration)) in interface [CRUDInterface](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html)

**Parameters:** configuration - конфігурація з налаштуваннями для вікриття

**Throws:** [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

close

* + - public void close() throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

Закриває сховище

**Throws:** [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

**Enum Storage**

* java.lang.Object
  + java.lang.Enum<[Storage](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html)>
    - dao.Storage
* **All Implemented Interfaces:**

[CRUDInterface](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html), java.io.Serializable, java.lang.Comparable<[Storage](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html)>

public enum **Storage**

extends java.lang.Enum<[Storage](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html)>

implements [CRUDInterface](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\CRUDInterface.html)

Хранилище данных

Enum Constant Summary

|  |
| --- |
| **Enum Constants** |
| **Enum Constant and Опис** |
| [**INSTANCE**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html#INSTANCE) |

|  |  |
| --- | --- |
| **Fields** | |
| **Тип** | **Field and Опис** |
| java.util.HashMap<java.lang.String,java.lang.Class> | [**classStorage**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html#classStorage) |

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| void | [**close**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html#close())()  Закриває сховище |
| <T> void | [**delete**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html#delete(T))(T instance)  видаляє обєкт зі сховища |
| java.util.List<java.lang.Class> | [**getClassStorage**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html#getClassStorage())()  Возращает список хранимых склассов |
| <T> T | [**insert**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html#insert(T))(T instance)  додає обьект у сховище |
| void | [**open**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html#open(configuration.IConfiguration))([**IConfiguration**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\configuration\IConfiguration.html) configuration)  відкриває сховище |
| <T> T | [**read**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html#read(java.lang.Class, java.util.UUID))(java.lang.Class entityClass, java.util.UUID id)  повертає об'єкт з сховища за Id |
| <T> java.util.List<T> | [**select**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html#select(java.lang.Class, dao.DAOFilter))(java.lang.Class entityClass, [**DAOFilter**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\DAOFilter.html) filter)  Повертає список обєктів, що задовільняють умовам фільтру |
| <T> java.util.List<T> | [**select**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html#select(java.lang.String))(java.lang.String SQLString) |
| void | [**setClassStorage**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html#setClassStorage(java.util.List))(java.util.List<java.lang.Class> classes)  Устанавливает какие классы необходимо хранить |
| <T> void | [**update**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html#update(T))(T instance)  Оновлює об'єкт у сховищі |
| static [**Storage**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html) | [**valueOf**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html#valueOf(java.lang.String))(java.lang.String name)  Returns the enum constant of this type with the specified name. |
| static [**Storage**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html)[] | [**values**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\Storage.html#values())()  Returns an array containing the constants of this enum type, in the order they are declared. |

setClassStorage

public void setClassStorage(java.util.List<java.lang.Class> classes)

Устанавливает какие классы необходимо хранить

**Parameters:** classes - необходимые классы

getClassStorage

public java.util.List<java.lang.Class> getClassStorage()

Возращает список хранимых склассов

**Returns:** классы

**Пакет dao.annotation**

Аннотации для разметки классов к которым будет применятся DAO

|  |  |
| --- | --- |
| **Annotation Types Summary** | |
| **Annotation Type** | **Опис** |
| [**Primary**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\annotation\Primary.html) |  |
| [**Stored**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\annotation\Stored.html) |  |

**Annotation Type Primary**

@Retention(value=RUNTIME)

@Target(value=FIELD)

public @interface **Primary**

**Annotation Type Stored**

@Retention(value=RUNTIME)

public @interface **Stored**

|  |  |
| --- | --- |
| **Основні класи** | |
| **Class** | **Опис** |
| [**AllObjectsOfClass**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\filters\AllObjectsOfClass.html) | Возвращает все обьекты класса |
| [**GetChildrenTask**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\filters\GetChildrenTask.html) | Возращает подзадачи |

**Пакет initialization**

Запуск, инициализация и завершение программы

|  |  |
| --- | --- |
| **Enum Summary** | |
| **Enum** | **Опис** |
| [**Application**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\initialization\Application.html) | Точка входа в приложения |

Пакет initialization Опис

Запуск, инициализация и завершение программы

Enum Application

* java.lang.Object
  + java.lang.Enum<[Application](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\initialization\Application.html)>
    - initialization.Application
* **All Implemented Interfaces:**

java.io.Serializable, java.lang.Comparable<[Application](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\initialization\Application.html)>

public enum **Application**

extends java.lang.Enum<[Application](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\initialization\Application.html)>

Точка входа в приложения

|  |
| --- |
| **Enum Constants** |
| **Enum Constant and Опис** |
| [**INSTANCE**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\initialization\Application.html#INSTANCE) |

|  |  |
| --- | --- |
| **Fields** | |
| **Тип** | **Field and Опис** |
| java.util.Map<java.lang.String,[**AllObjectClass**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html)> | [**DATA**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\initialization\Application.html#DATA) |
| java.util.Map<javax.swing.JFrame,java.lang.Boolean> | [**openFrame**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\initialization\Application.html#openFrame) |

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| static void | [**main**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\initialization\Application.html#main(java.lang.String[]))(java.lang.String[] args) |

**Пакет model**

Модель данных прикладной области

|  |  |
| --- | --- |
| **Основні класи** | |
| **Class** | **Опис** |
| [**AllCategoryUsers**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllCategoryUsers.html) | Зберігає всі категорії користівачів |
| [**AllConsists**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllConsists.html) | Зберігає всі прилажності користувачів до груп |
| [**AllGroups**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllGroups.html) | Зберігає всі групи |
| [**AllObjectClass**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html)<T> | Хранит все обьекты одного класса |
| [**AllTasks**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllTasks.html) | Зберігає всі задачі |
| [**AllUsers**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllUsers.html) | Зберігаж всіх користувачів |
| [**CategoryUsers**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\CategoryUsers.html) | Категорія користувача |
| [**Consist**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Consist.html) | Відповідність між користувачем та группою доя кої він належить |
| [**Groups**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Groups.html) | Група користувачів |
| [**Model**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Model.html) | Інтерфейс моделі |
| [**Task**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Task.html) | Описує задачу |
| [**User**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\User.html) | Описує користувача |

Class AllCategoryUsers

* java.lang.Object
  + javax.swing.event.EventListenerList
    - [model.Model](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Model.html)
      * [model.AllObjectClass](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html)
        + model.AllCategoryUsers
* **All Implemented Interfaces:**

java.io.Serializable

public class **AllCategoryUsers**

extends [AllObjectClass](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html)

Зберігає всі категорії користівачів

Constructor Summary

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**AllCategoryUsers**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllCategoryUsers.html#AllCategoryUsers())() |

**Class AllConsists**

* java.lang.Object
  + javax.swing.event.EventListenerList
    - [model.Model](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Model.html)
      * [model.AllObjectClass](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html)
        + model.AllConsists
* **All Implemented Interfaces:**

java.io.Serializable

public class **AllConsists**

extends [AllObjectClass](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html)

Зберігає всі прилажності користувачів до груп

**See Also:**

[Serialized Form](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\serialized-form.html#model.AllConsists)

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**AllConsists**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllConsists.html#AllConsists())() |

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| java.util.List<java.lang.String> | [**getGroups**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllConsists.html#getGroups(java.lang.String))(java.lang.String idUser)  Поверає список груп,до яких належить користувач |
| java.util.List<java.lang.String> | [**getUsersGroup**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllConsists.html#getUsersGroup(java.lang.String))(java.lang.String idGroup)  Повертає склад групи |

#### AllConsists

* + - public AllConsists() throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

**Throws:** [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

#### getUsersGroup

public java.util.List<java.lang.String> getUsersGroup(java.lang.String idGroup)

Повертає склад групи

**Parameters:** idGroup - id групи, для якої визначається склад

**Returns:** склад групи

#### getGroups

public java.util.List<java.lang.String> getGroups(java.lang.String idUser)

Поверає список груп,доя ких належить користувач

**Parameters:**idUser - id користувача

**Returns:** список груп

**Class AllGroups**

* java.lang.Object
  + - [model.Model](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Model.html)
      * [model.AllObjectClass](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html)
        + model.AllGroups

public class **AllGroups**

extends [AllObjectClass](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html)

Зберігає всі групи

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**AllGroups**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllGroups.html#AllGroups())() |

#### AllGroups

* + - public AllGroups() throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

**Throws:** [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

**Class AllObjectClass<T>**

* java.lang.Object
  + - [model.Model](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Model.html)
      * model.AllObjectClass<T>
* **Type Parameters:**

T - тип обьектов

Хранит все обьекты одного класса

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**AllObjectClass**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html#AllObjectClass())() |

Method Summary

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| boolean | [**add**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html#add(T))([**T**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html) object)  Добавляет обьект в список и в хранилище |
| boolean | [**delete**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html#delete(T))([**T**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html) object)  удаляет обьект из списка и из хранилища |
| java.util.Set<[**T**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html)> | [**getAll**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html#getAll())()  возращает все обьекты |
| void | [**update**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html#update(T))([**T**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html) object)  Обновляет состояние обьекта в хранилище |

#### AllObjectClass

public AllObjectClass()

Method Detail

#### update

* + - public void update([T](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html) object) throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

Обновляет состояние обьекта в хранилище

**Parameters:**object - обьект

**Throws:**[DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

#### add

* + - public boolean add([T](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html) object) throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

Добавляет обьект в список и в хранилище

**Parameters:** object - говый обьект

**Returns:** результат добавления

**Throws:**[DAOException](file:///E:\\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\\%D0%9F%D0%9F%D0%97\\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\\program\\doc\\dao\\exception\\DAOException.html" \o "class in dao.exception)

#### delete

* + - public boolean delete([T](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html) object) throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

удаляет обьект из списка и из хранилища

**Parameters:**object - удаляемый обьект

**Returns:** результат удаления

**Throws:**[DAOException](file:///E:\\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\\%D0%9F%D0%9F%D0%97\\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\\program\\doc\\dao\\exception\\DAOException.html" \o "class in dao.exception)

#### getAll

public java.util.Set<[T](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html)> getAll()

возращает все обьекты

**Returns:** множество обьектов

**Class AllTasks**

* java.lang.Object
  + - [model.Model](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Model.html)
      * [model.AllObjectClass](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html)
        + model.AllTasks

|  |  |
| --- | --- |
| **Конструктори** | |
| **Опис конструктора** | |
| [**AllTasks**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllTasks.html#AllTasks())() | |
| **Методи** | | |
| **Тип** | **Опис методів** | |
| java.util.List<[**Task**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Task.html)> | [**getChildrensTask**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllTasks.html#getChildrensTask(java.util.UUID))(java.util.UUID parentId)  повертає підзадачі по id | |

#### AllTasks

* + - public AllTasks() throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

**Throws:** [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

Method Detail

#### getChildrensTask

* + - public java.util.List<[Task](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Task.html)> getChildrensTask(java.util.UUID parentId) throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

повертає підзадачі по id

**Parameters:** parentId -id батьківської здачі

**Returns:** підзадачі

**Throws:** [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

**Class AllUsers**

* java.lang.Object
  + - [model.Model](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Model.html)
      * [model.AllObjectClass](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllObjectClass.html)
        + model.AllUsers

Зберігаж всіх користувачів

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**AllUsers**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllUsers.html#AllUsers())() |

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| [**User**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\User.html) | [**getUserFromUUID**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\AllUsers.html#getUserFromUUID(java.util.UUID))(java.util.UUID id)  Повертає користувача за id |

Constructor Detail

#### AllUsers

* + - public AllUsers() throws [DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

**Throws:**[DAOException](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\dao\exception\DAOException.html)

Method Detail

#### getUserFromUUID

public [User](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\User.html) getUserFromUUID(java.util.UUID id)

Повертає користувача за id

**Parameters:**id - id пользователя которого нужно найти

**Returns:**-користувач,null - якщо такого користувача немає

**Class CategoryUsers**

* java.lang.Object
  + model.CategoryUsers

Категорія користувача

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**CategoryUsers**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\CategoryUsers.html#CategoryUsers())() |
| [**CategoryUsers**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\CategoryUsers.html#CategoryUsers(java.lang.String))(java.lang.String name) |

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| java.util.UUID | [**getId**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\CategoryUsers.html#getId())() |
| java.lang.String | [**getName**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\CategoryUsers.html#getName())() |
| void | [**setId**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\CategoryUsers.html#setId(java.util.UUID))(java.util.UUID id) |
| void | [**setName**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\CategoryUsers.html#setName(java.lang.String))(java.lang.String name) |

**Class Consist**

* java.lang.Object
  + model.Consist

Відповідність між користувачем та группою доя кої він належить

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**Consist**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Consist.html#Consist())() |
| [**Consist**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Consist.html#Consist(java.util.UUID, java.util.UUID))(java.util.UUID idGroup, java.util.UUID idUser) |

Method Summary

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| java.util.UUID | [**getId**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Consist.html#getId())()  - повертає Id |
| java.lang.String | [**getIdGroup**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Consist.html#getIdGroup())() - повертає Id групи |
| java.lang.String | [**getIdUser**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Consist.html#getIdUser())() - повертає Id користувача |
| void | [**setId**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Consist.html#setId(java.util.UUID))(java.util.UUID id) - встановлює Id |
| void | [**setIdGroup**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Consist.html#setIdGroup(java.lang.String))(java.lang.String idGroup) - встановлює групи |
| void | [**setIdUser**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Consist.html#setIdUser(java.lang.String))(java.lang.String idUser) - встановлює Id користувача |

**Class Model**

* java.lang.Object
  + - model.Model

Інтерфейс моделі

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**Model**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Model.html#Model())() |

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| void | [**addListener**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Model.html#addListener(java.lang.Class, view.ModelListener))(java.lang.Class modelEvent, [**ModelListener**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\view\ModelListener.html) listener)  Добавить слушателя |
| void | [**removeListener**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Model.html#removeListener(java.lang.Class, view.ModelListener))(java.lang.Class modelEvent, [**ModelListener**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\view\ModelListener.html) listener)  Видаляє слухача |

Constructor Detail

#### Model

public Model()

Method Detail

#### addListener

* + - public void addListener(java.lang.Class modelEvent, [ModelListener](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\view\ModelListener.html) listener)

Добавить слушателя

**Parameters:** listener - новый слушатель

#### removeListener

* + - public void removeListener(java.lang.Class modelEvent, [ModelListener](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\view\ModelListener.html) listener)

Видаляє слухача

**Parameters:**modelEvent -

**Пакет model.event**

События модели данных

|  |  |
| --- | --- |
| **Основні класи** | |
| **Class** | **Опис** |
| [**ModelEvent**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\event\ModelEvent.html) |  |

**Пакет view**

Графический интерфейс пользователя

|  |  |
| --- | --- |
| **Основні інтерфейси** | |
| **Interface** | **Опис** |
| [**ModelListener**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\view\ModelListener.html) |  |

|  |  |
| --- | --- |
| **Основні класи** | |
| **Class** | **Опис** |
| [**ExpertGroups**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\view\ExpertGroups.html) | Вікно для роботи з групами |
| [**ManagingTasksForm**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\view\ManagingTasksForm.html) | Вікно для роботи з задачами |

**Class ExpertGroups**

* java.lang.Object
  + java.awt.Component
    - java.awt.Container
      * java.awt.Window
        + java.awt.Frame

javax.swing.JFrame

view.ExpertGroups

Вікно для роботи з групами

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**ExpertGroups**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\view\ExpertGroups.html#ExpertGroups())() |

Method Summary

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| void | [**showGroup**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\view\ExpertGroups.html#showGroup(int))(int indexGroup) |

Constructor Detail

#### ExpertGroups

public ExpertGroups()

Method Detail

#### showGroup

public void showGroup(int indexGroup)

**Class ManagingTasksForm**

* java.lang.Object
  + java.awt.Component
    - java.awt.Container
      * java.awt.Window
        + java.awt.Frame

javax.swing.JFrame

view.ManagingTasksForm

Вікно для роботи з задачами

Constructor Summary

|  |
| --- |
| **Конструктори** |
| **Опис конструктора** |
| [**ManagingTasksForm**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\view\ManagingTasksForm.html#ManagingTasksForm())() |

|  |  |
| --- | --- |
| **Методи** | |
| **Тип** | **Опис методів** |
| void | [**showTask**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\view\ManagingTasksForm.html#showTask(model.Task))([**Task**](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Task.html) task) |

Constructor Detail

#### ManagingTasksForm

public ManagingTasksForm()

Create the frame.

Method Detail

#### showTask

public void showTask([Task](file:///E:\%D0%98%D0%BD%D1%81%D1%82%D0%B8%D1%82%D1%83%D1%82\4%20%D1%81%D0%B5%D0%BC%D0%B5%D1%81%D1%82%D1%80\%D0%9F%D0%9F%D0%97\%D0%BA%D1%83%D1%80%D1%81%D0%BE%D0%B2%D0%B0%D1%8F\program\doc\model\Task.html) task)

## 3.2 Діаграма класів

На діаграмі показані основні класи та основні методи

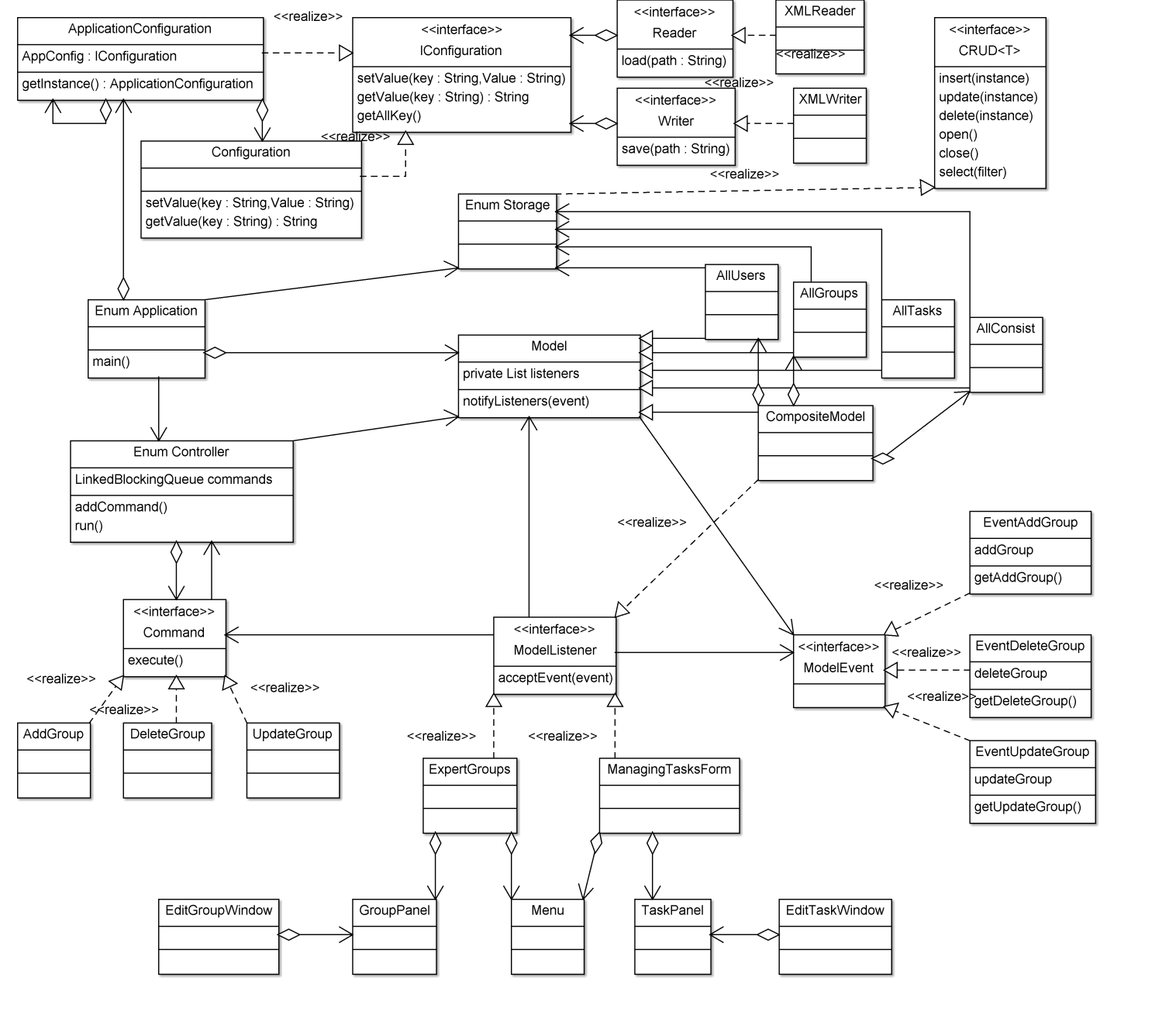
****

Рис. 3.1 Діаграма основних класів

## 3.3 Інстукція користування програмним додатком

**Додавання задачі або під задачі**

1) Виділити задачу вищого рівня (при доданні задачі потрібно виділити корінь дерева задач) клацнути правою кнопкою миші та вибрати з контекстного меню пункт «Додати задачу». Відкриється вікно додавання нової задачі.

2) Задати всі необхідні поля та натиснути кнопку «Додати».

**Редагування задачі**

1) Виділити необхідну задачу.

2) З контекстного меню вибрати пункт «Редагувати». Відкриється вікно редагування.

3) Внести зміни натиснути кнопку «Зберегти».

**Видалення задачі**

1) Виділити необхідну задачу.

2) З контекстного меню вибрати пункт «Видалити».

3) Підтвердити видалення.

**Зміна ієрархії задач**

1) Виділити необхідну задачу.

2) З контекстного меню вибрати пункт «Змінити ієрархію».

3) Виділити вузол до якого необхідно перемістити вибрану задачу.

**Створення нової групи експертів**

1) Клацнути правою кнопкою миші та вибрати з контекстного меню пункт «Додати групу». Відкриється вікно додавання нової групи.

2) Задати всі необхідні поля та натиснути кнопку «Ок».

**Редагування групи**

1) Виділити необхідну групу.

2) З контекстного меню вибрати пункт «Змінити групу». Відкриється вікно редагування.

3) Внести зміни натиснути кнопку «Ок».

**Видалення групи**

1) Виділити необхідну групу.

2) З контекстного меню вибрати пункт «Видалити групу».

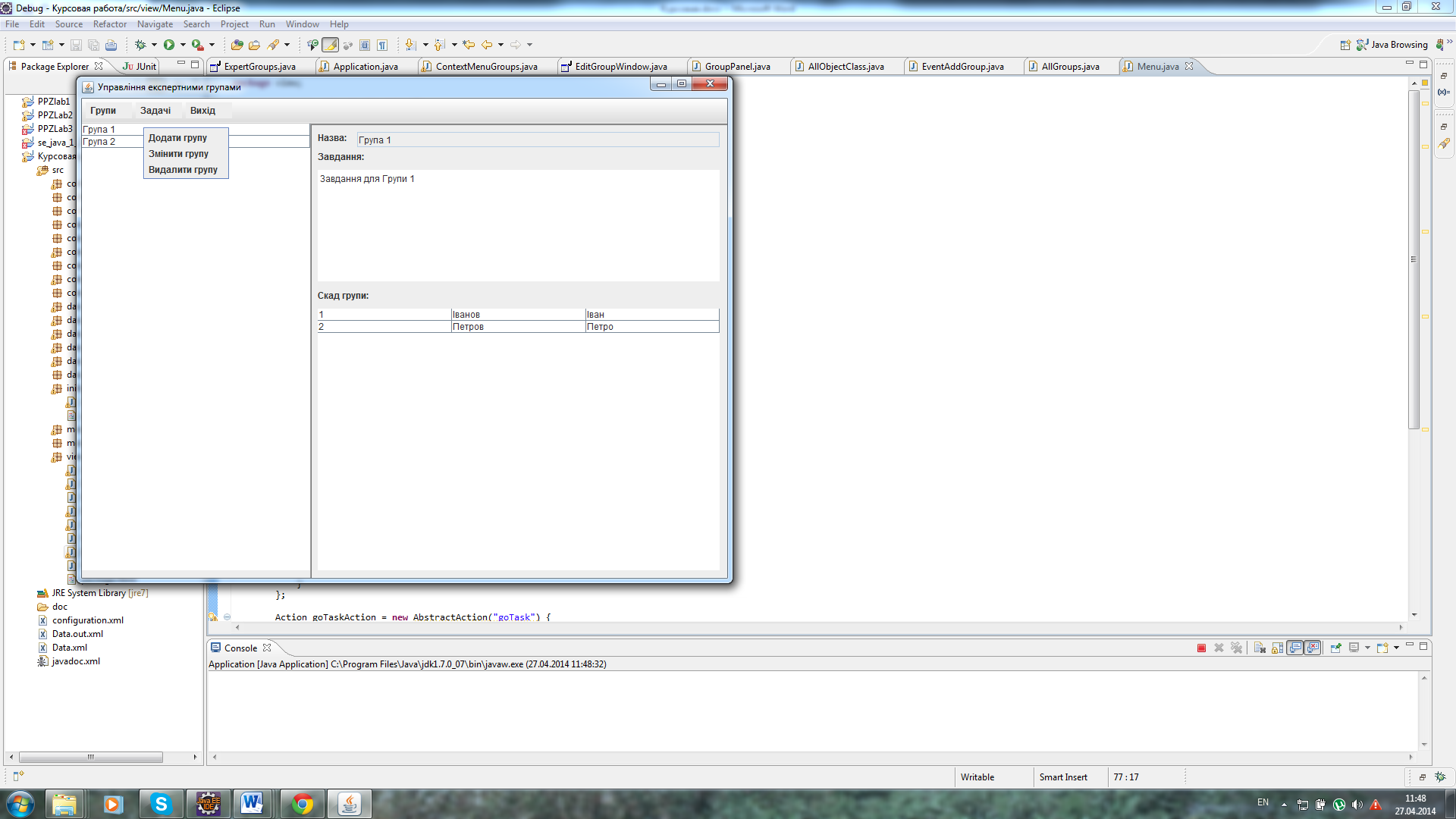
3) Підтвердити видалення.

# 4 ТЕСТУВАННЯ ПРОГРАМНОГО ДОДАТКУ

Тестування програмного забезпечення є дуже важливим етапом у розробці програми. У зв’язку з цим було розроблено дуже багато методик та засобів для тестування. Найпоширенішим методом тестування є Unit Test, він дозволяє виконувати тестування на самих ранніх версіях програми, але він неможе використуваватися при тестуванні графічного інтерфейсу користувача.

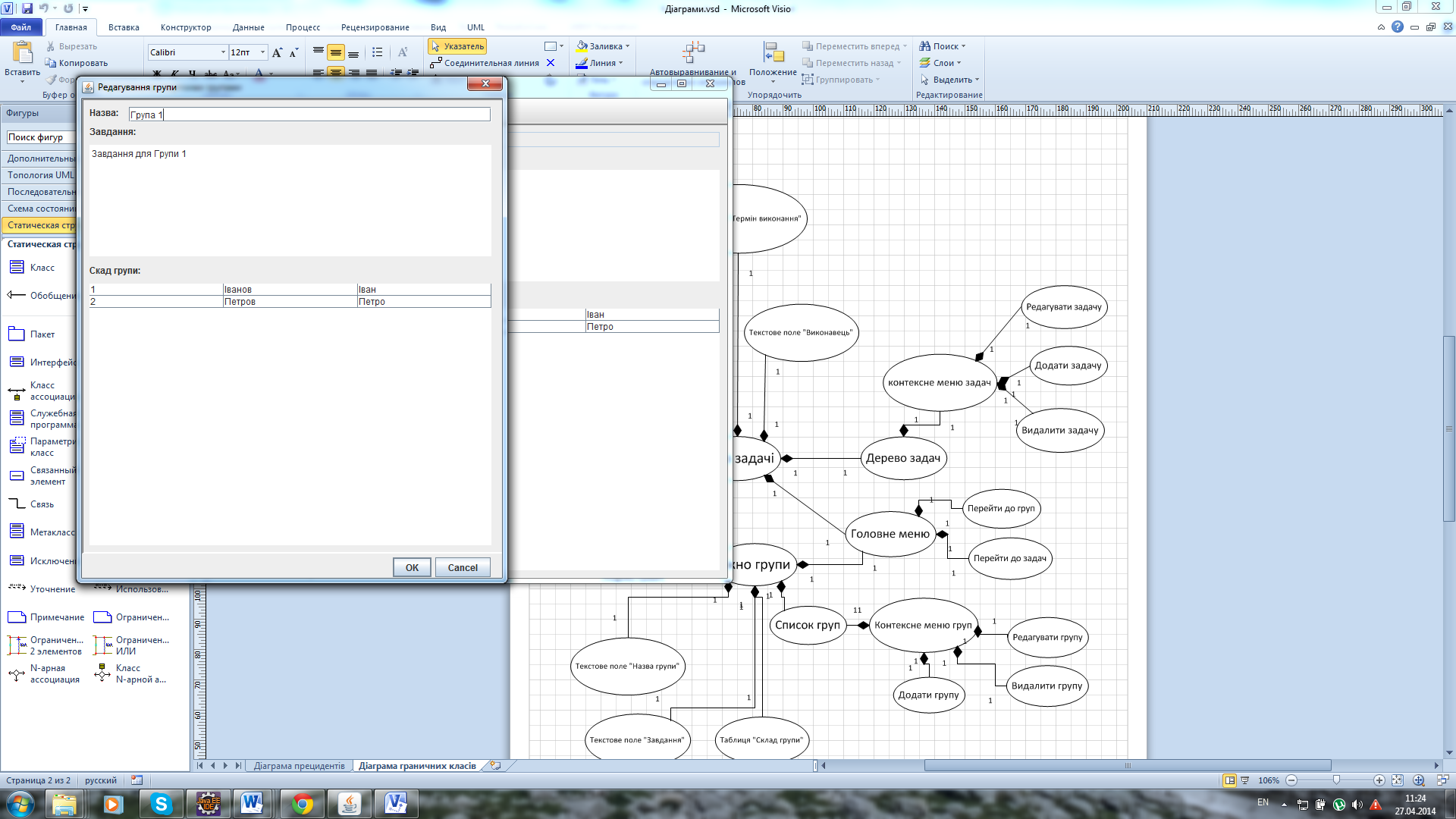
При розробці великих програмних продуктів, для тестування графічного інтерфейсу використовують засоби автоматичного порівняння скріншотів вікна програми з еталоном. Але це складний механізм і він невиправдовує себе для малих програм.

На рисунку 4.1 представлено вікно управління групами.



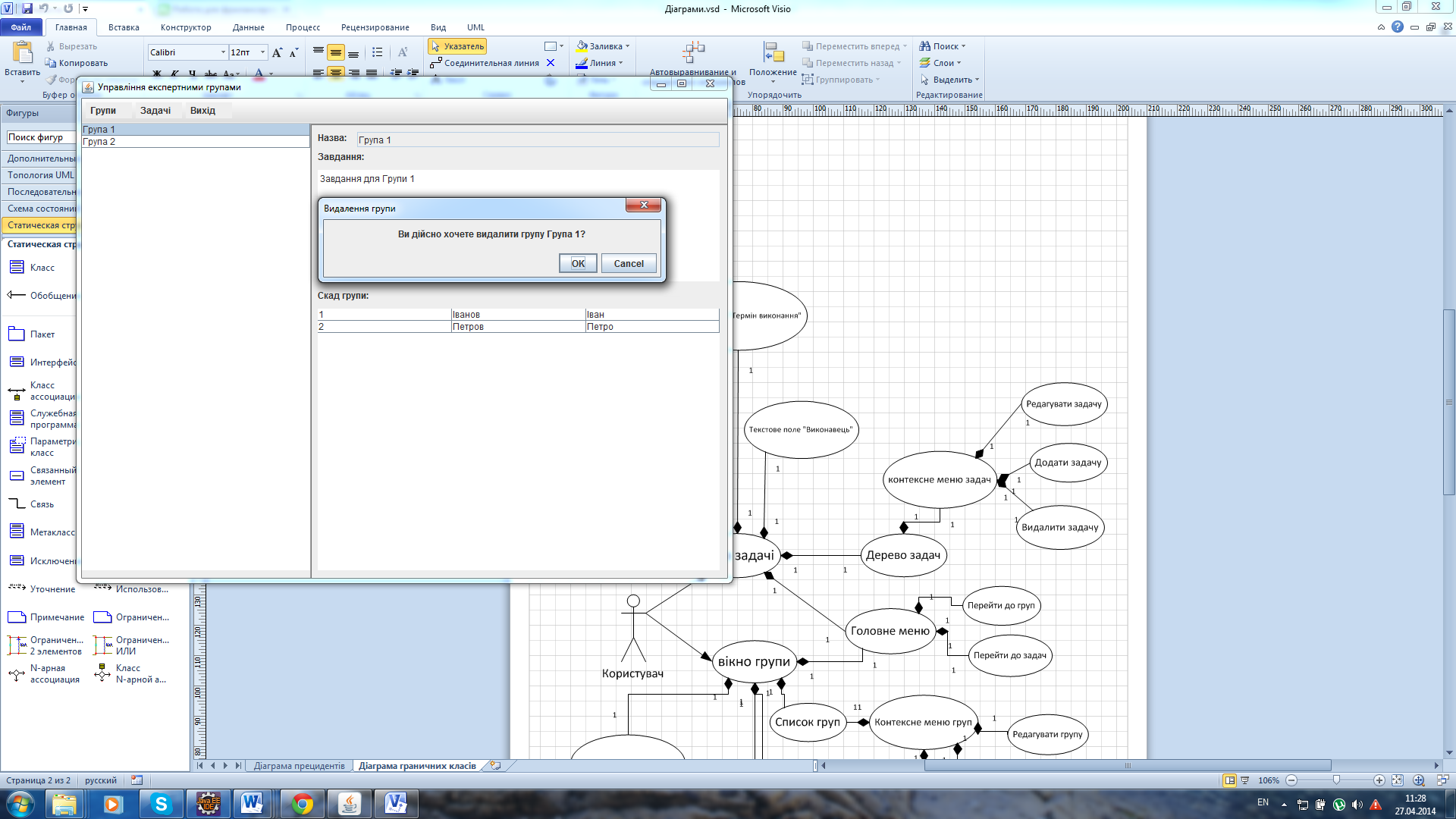
4.1 Вікно для управління групами

На рисунку 4.2 представлено вікно редагування групи.



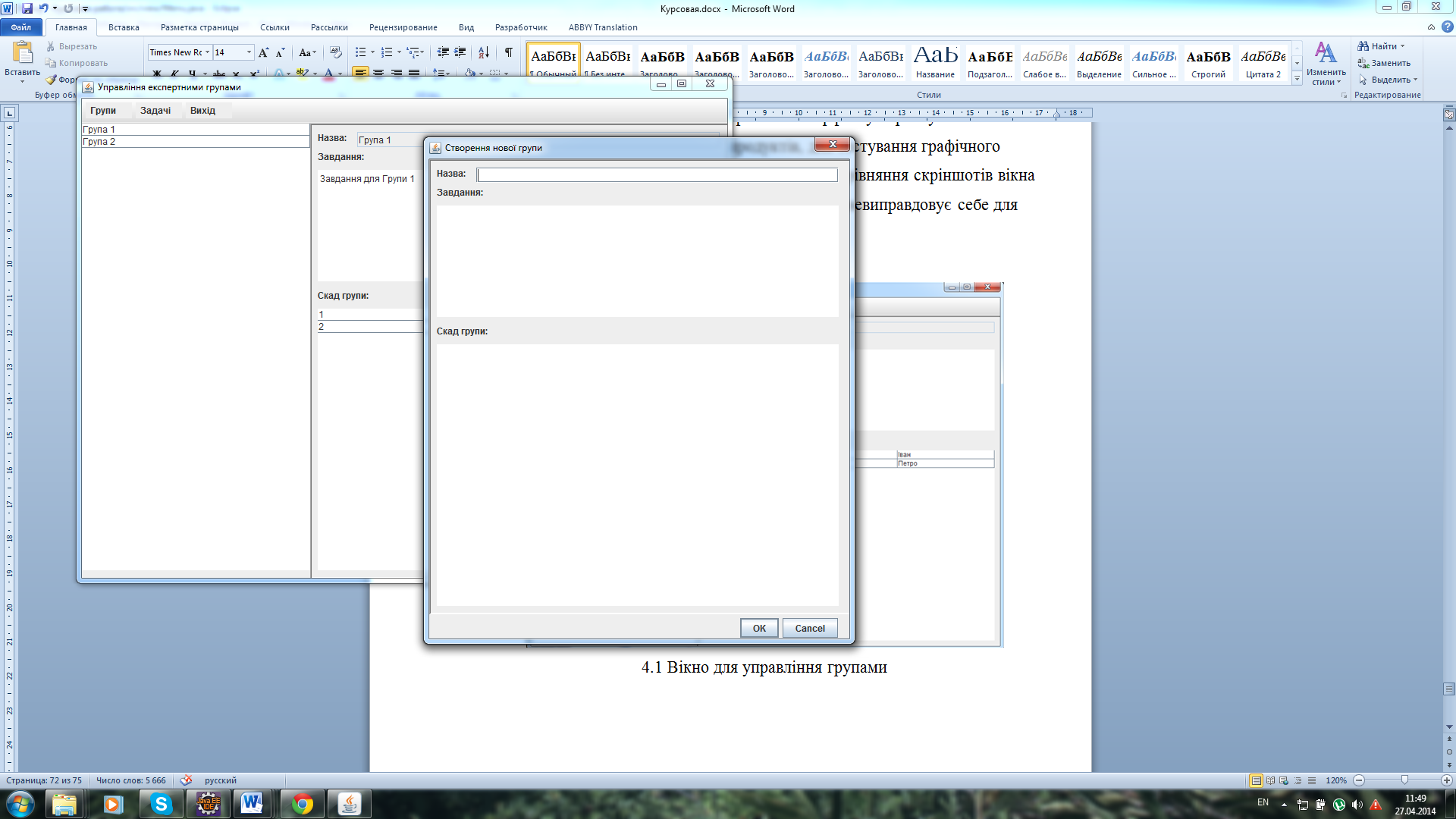
4.2 Вікно для редагування групи

На рисунку 4.3 представлено вікно видалення групи.



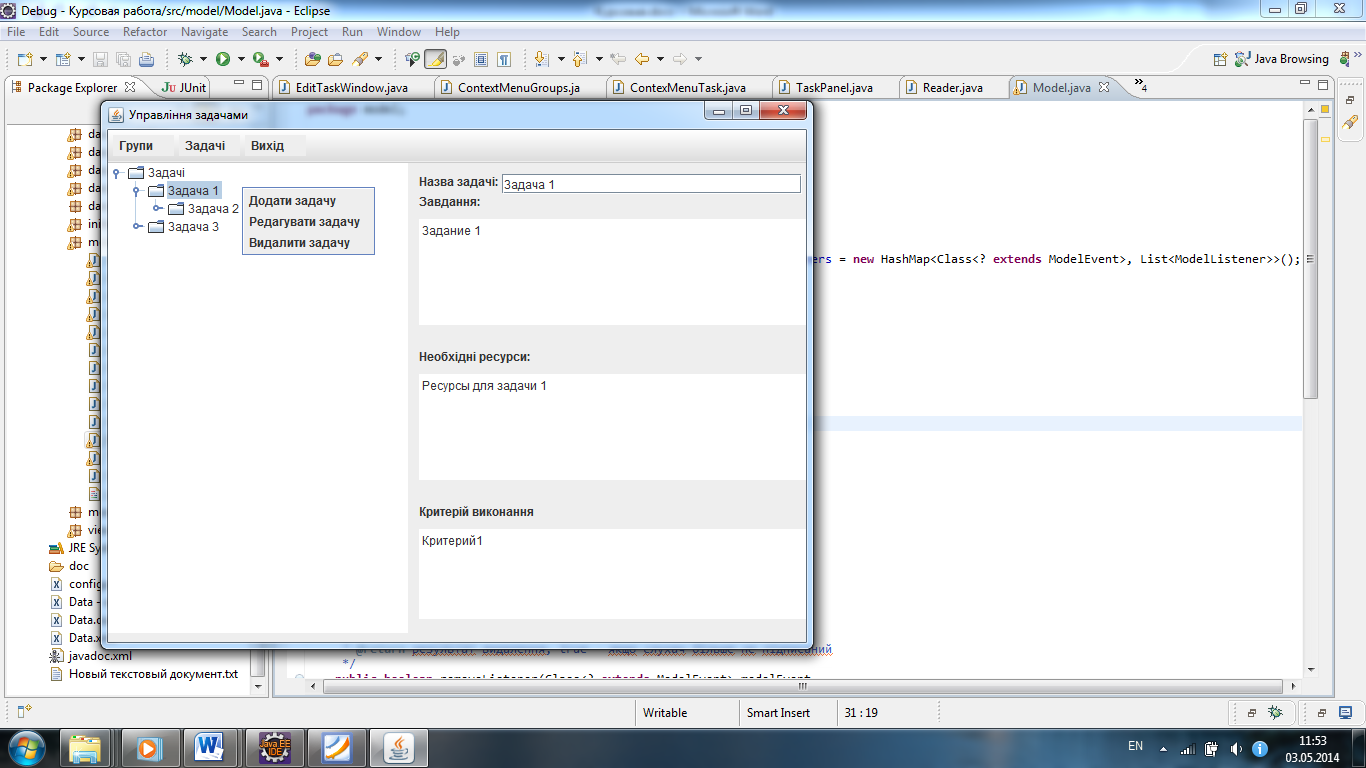
4.3 Вікно видалення групи

На рисунку 4.3 представлено вікно створення нової групи.



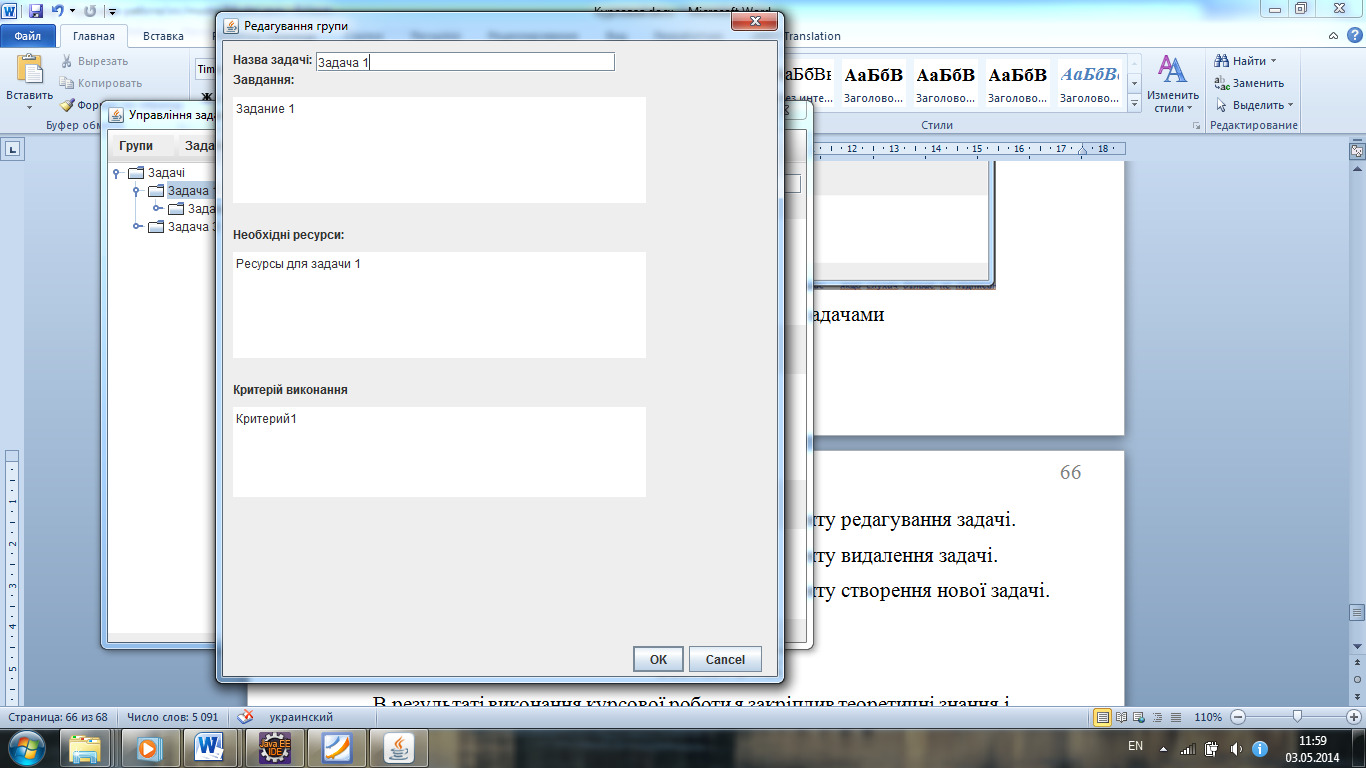
4.4 Вікно для створення нової групи

На рисунку 4.5 представлено вікно управлінням задачами.



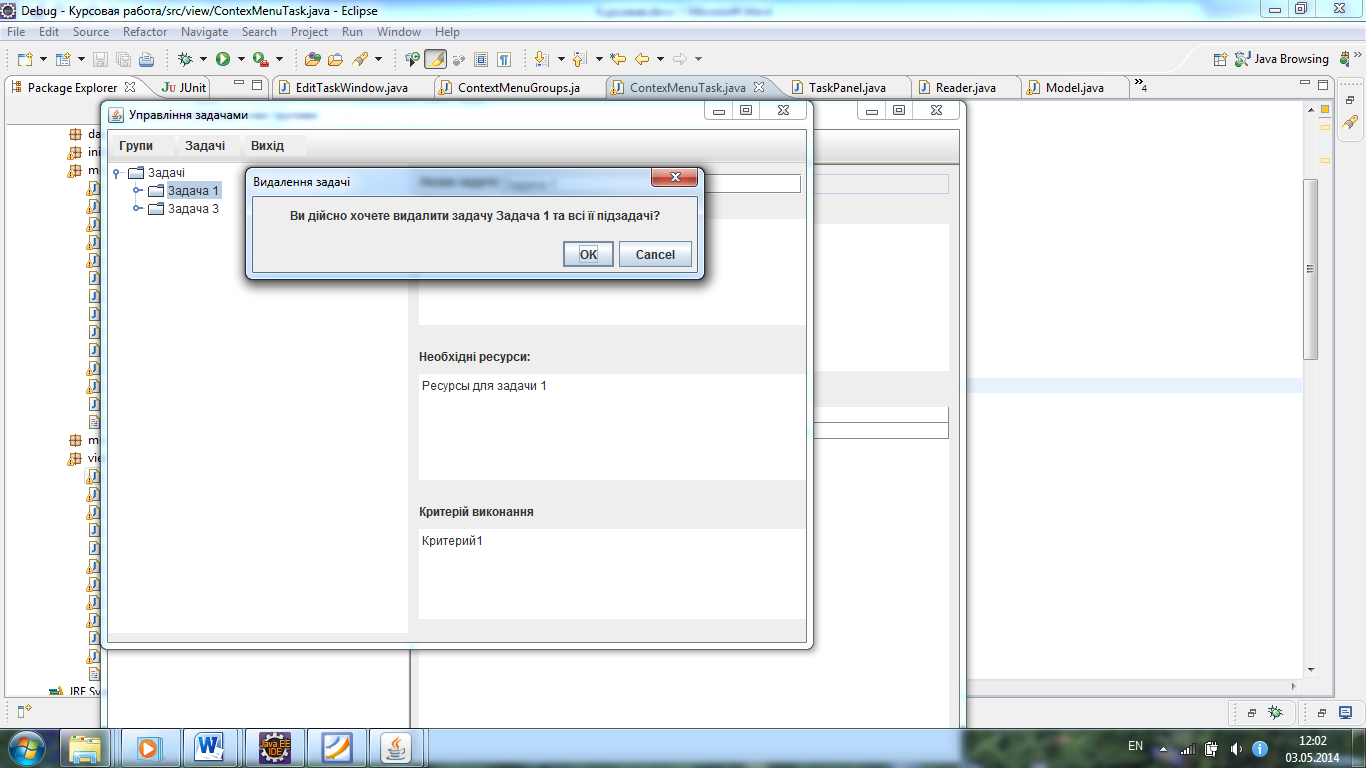
4.5 Вікно для управління задачами

На рисунку 4.6 представлено вікно редагування задачі.



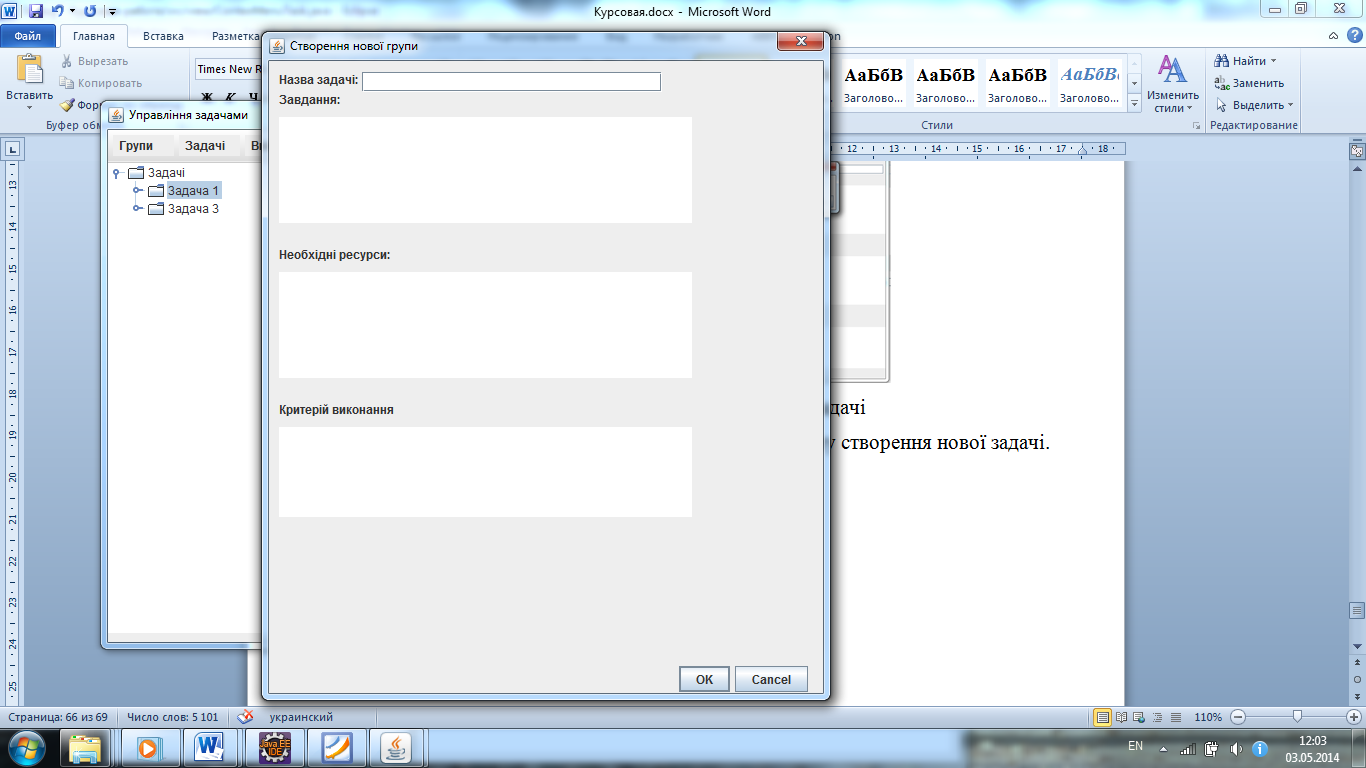
4.6 Вікно для редагування задачі

На рисунку 4.7 представлено вікно видалення задачі.



4.7 Вікно для видалення задачі

На рисунку 4.8 представлено вікно створення нової задачі.



4.7 Вікно для створення нової задачі

# ВИСНОВОК

В результаті виконання курсової роботи були закріплені теоретичні знання та отримані практичні навички з проектування, моделювання, розробки та тестування програмного забезпечення.

Закріплено використання шаблону проектування MVC, багатопотоковість у Java, бібліотеку для створення графічного інтерфейсу Swing на прикладі програми, що призначена для управління експертними групами.

# ЛІТЕРАТУРА

1. Огляд патерну MVC - [Електронне джерело] – режим доступу: <http://uk.wikipedia.org/wiki/MVC> - Заголовок з вікна.
2. Обобщенный Model-View-Controller [Електронне джерело] – режим доступу: <http://www.rsdn.ru/article/patterns/generic-mvc.xml#EZB> - Заголовок з вікна.
3. Г. Шилдт. Полный справочник по Java SE6: 7-е издание. Москва: «Вильямс», 2007. - 1035 с.( ISBN 0-07-226385-7/ ISBN 978-5-8459-1168-1).
4. Г.Шилдт. Swing руководство для начинающих: .Москва: «Вильямс», 2007. - 704 с. ( ISBN 0-07-226314-8/ISBN 978-5-8459-1162-9).
5. Б. Эккель. Философия Java. Библиотека программиста: 2-е издание. «Питер», 2009. - 637 c. (ISBN 5-88782-105-1).
6. Berthold Daum. Professional Eclipse 3 for Java™ Developers. «Wrox», 2004. - 575 c. (ISBN 0-470-02005-9).
7. Kathy Sierra/Bert Bates. SCJP Sun®Certifed Programmer for Java™ 6 Study Guide. «McGraw-Hill», 2008. - 890 c. (ISBN 0-07-159107-9).
8. И.Н. Блинов. Java .ПРОМЫШЛЕННОЕ ПРОГРАММИРОВАНИЕ. Минск, «УниверсалПресс», 2007. -704 c. (ISBN 978-985-6699-63-7).
9. Гамма Э., Хелм Р., Джонсон Р.Приемы объектно-ориентированного проектирования. «Питер», 2008. -336 c. (ISBN 5-469-01136-4 ).
10. Д. Блох. Java. Эффективное программирование. Москва: «Лори», 2002. – 224 с.
11. И.Н. Блинов, В.С. Романчик. Java 2. Практическое руководство. Мн.: «УниверсалПресс», 2005. – 400 с.
12. Д. Блох. Java. Эффективное программирование. М.: «Лори», 2002. – 224 с.

# ДОДАТКИ

## Додаток А. Код програми

package initialization;

import dao.Storage;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import java.util.Set;

import javax.swing.JFrame;

import view.ExpertGroups;

import view.ManagingTasksForm;

import model.AllCategoryUsers;

import model.AllConsists;

import model.AllGroups;

import model.AllObjectClass;

import model.AllTasks;

import model.AllUsers;

import model.CategoryUsers;

import model.CompositeModel;

import model.Consist;

import model.Groups;

import model.Task;

import model.User;

import model.event.EventAddGroup;

import model.event.EventDeleteGroup;

import model.event.EventUpdateGroup;

import configuration.ApplicationConfiguration;

import configuration.utils.xml.XMLReader;

import core.controller.Controller;

import dao.exception.DAOException;

/\*\*

\* Точка входа в приложения

\*

\*/

public enum Application {

INSTANCE;

public Map<String, AllObjectClass> DATA;

public Map<JFrame, Boolean> openFrame;

private final String CONFIGURATION\_FILE\_NAME = "configuration.xml";

private Thread threadController;

private void loadData() {

List<Class> classStorage = new ArrayList<Class>();

classStorage.add(CategoryUsers.class);

classStorage.add(Task.class);

classStorage.add(User.class);

classStorage.add(Groups.class);

classStorage.add(Consist.class);

Storage storage = Storage.INSTANCE;

try {

storage.setClassStorage(classStorage);

storage.open(ApplicationConfiguration.INSTANCE);

DATA.put(Task.class.getName(), new AllTasks());

DATA.put(User.class.getName(), new AllUsers());

DATA.put(CategoryUsers.class.getName(), new AllCategoryUsers());

DATA.put(Groups.class.getName(), new AllGroups());

DATA.put(Consist.class.getName(), new AllConsists());

} catch (DAOException e) {

e.printStackTrace();

}

}

private void loadConfiguration() {

XMLReader xmlReader = new XMLReader(ApplicationConfiguration.INSTANCE);

xmlReader.load(CONFIGURATION\_FILE\_NAME);

}

private void startController() {

threadController = new Thread(Controller.INSTANCE);

threadController.start();

}

private void stopController() {

while (!Controller.INSTANCE.isEmpty) {

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

try {

threadController.join(10);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

private void showView() {

ManagingTasksForm frameManagingTasks = new ManagingTasksForm();

openFrame.put(frameManagingTasks, true);

ExpertGroups frameExpertGroups = new ExpertGroups();

openFrame.put(frameExpertGroups, true);

CompositeModel model = new CompositeModel();

model.addModel(DATA.get(Groups.class.getName()));

model.addListener(EventUpdateGroup.class, frameExpertGroups);

model.addListener(EventDeleteGroup.class, frameExpertGroups);

model.addListener(EventAddGroup.class, frameExpertGroups);

frameExpertGroups.setVisible(true);

frameManagingTasks.setVisible(true);

}

private void saveData() {

try {

Storage.INSTANCE.close();

} catch (DAOException e) {

e.printStackTrace();

}

System.out.println("saveData");

}

/\*\*

\* Проверяет есть ли открытые окна

\*

\* @throws InterruptedException

\*/

private void checkVisibleView() throws InterruptedException {

while (true) {

boolean result = false;

Set<JFrame> frames = openFrame.keySet();

for (JFrame frame : frames) {

if (openFrame.get(frame)) {

result = true;

break;

}

}

if (result) {

Thread.sleep(1000);

} else {

break;

}

}

}

private void closeApplication() {

System.exit(0);

}

public static void main(String[] args) throws InterruptedException {

INSTANCE.loadConfiguration();

INSTANCE.loadData();

INSTANCE.startController();

INSTANCE.showView();

INSTANCE.checkVisibleView();

INSTANCE.stopController();

INSTANCE.saveData();

INSTANCE.closeApplication();

}

private Application() {

DATA = new HashMap<String, AllObjectClass>();

openFrame = new HashMap<JFrame, Boolean>();

}

}

**package** model;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** java.util.List;

**import** java.util.Map;

**import** model.event.ModelEvent;

**import** view.ModelListener;

/\*\*

\* Інтерфейс моделі

\*

\*/

**public** **abstract** **class** Model {

**private** Map<Class<? **extends** ModelEvent>, List<ModelListener>> listeners = **new** HashMap<Class<? **extends** ModelEvent>, List<ModelListener>>();

/\*\*

\* Добавить слушателя

\*

\* **@param** listener

\* новый слушатель

\* **@return** результат добаления, true если добавлен

\*/

**public** **boolean** addListener(Class<? **extends** ModelEvent> modelEvent,

ModelListener listener) {

List<ModelListener> list = listeners.get(modelEvent);

**if** (list == **null**) {

list = **new** ArrayList<ModelListener>();

listeners.put(modelEvent, list);

}

**return** list.add(listener);

}

/\*\*

\* Видаляє слухача

\*

\* **@param** modelEvent

\* тип подій

\* **@param** listener

\* слухач

\* **@return** результат видалення, true - якщо слухач більше не підписаний

\*/

**public** **boolean** removeListener(Class<? **extends** ModelEvent> modelEvent,

ModelListener listener) {

List<ModelListener> list = listeners.get(modelEvent);

**if** (list == **null**) {

**return** **true**;

}

**return** list.remove(listener);

}

/\*\*

\* Оповіщує слухачів підписаних на подію event,що модель змінилася

\*

\* **@param** event

\* подія

\*/

**protected** **void** notifyListeners(ModelEvent event) {

// підписники на тип подій event

List<ModelListener> list = listeners.get(event.getClass());

**if** (list != **null**)

**for** (ModelListener listener : list) {

listener.acceptEvent(event);

}

// підписники на всі події

List<ModelListener> list2 = listeners.get(ModelEvent.**class**);

**if** (list2 != **null**)

**for** (ModelListener modelListener : list2) {

**if** ((list == **null**) || (!list.contains(modelListener))) {

modelListener.acceptEvent(event);

}

}

}

}

package model;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

import java.util.UUID;

import java.util.concurrent.locks.Lock;

import java.util.concurrent.locks.ReentrantReadWriteLock;

import dao.Storage;

import dao.exception.DAOException;

import dao.filters.AllObjectsOfClass;

/\*\*

\* Хранит все обьекты одного класса

\*

\* @param <T>

\* - тип обьектов

\*/

public abstract class AllObjectClass<T> extends Model {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

protected Set<T> allObject = new HashSet<T>();

private ReentrantReadWriteLock rwl = new ReentrantReadWriteLock();// обьект

// блокировки

protected Lock readLock = rwl.readLock(); // обьект блокировки чтения

protected Lock writeLock = rwl.writeLock(); // обьект блокировки записи

/\*\*

\* загружает данные из хранилища

\*

\* @param currentClass

\* - тип загружаемых обьектов

\* @throws DAOException

\*/

protected void load(Class currentClass) throws DAOException {

List<T> list = Storage.INSTANCE.select(currentClass,

new AllObjectsOfClass());

for (T t : list) {

allObject.add(t);

}

}

/\*\*

\* Обновляет состояние обьекта в хранилище

\*

\* @param object

\* - обьект

\* @throws DAOException

\*/

public void update(T object) {

try {

Storage.INSTANCE.update(object);

} catch (DAOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

/\*\*

\* Добавляет обьект в список и в хранилище

\*

\* @param object

\* - говый обьект

\* @return - результат добавления

\* @throws DAOException

\*/

public boolean add(T object) throws DAOException {

boolean result = false;

writeLock.lock();

try {

if (!allObject.contains(object))

result = allObject.add(object);

if (result) {

Storage.INSTANCE.insert(object);

}

} finally {

writeLock.unlock();

}

if(result){

}

return result;

};

/\*\*

\* удаляет обьект из списка и из хранилища

\*

\* @param object

\* - удаляемый обьект

\* @return - результат удаления

\* @throws DAOException

\*/

public boolean delete(T object) throws DAOException {

writeLock.lock();

try {

boolean result = allObject.remove(object);

if (result) {

Storage.INSTANCE.delete(object);

return true;

}

} finally {

writeLock.unlock();

}

return false;

}

/\*\*

\* возращает все обьекты

\*

\* @return - множество обьектов

\*/

public Set<T> getAll() {

return allObject;

}

}

package core.controller;

import java.util.concurrent.LinkedBlockingQueue;

import core.controller.Commands.Command;

import core.controller.annotation.utils.Validator;

import core.controller.exception.IncorrectParameterListException;

/\*\*

\* Принимает и выполянет команды переданые от Viev

\*

\*/

public enum Controller implements Runnable {

/\*\*

\* Обєкт контроллера

\*/

INSTANCE;

private LinkedBlockingQueue<Command> commands;

private Validator validator;

/\*\*

\* Показує,чи є команди для виконання true - якщо немає команд для виконання

\*/

public boolean isEmpty;

private Controller() {

commands = new LinkedBlockingQueue<Command>();

validator = new Validator();

isEmpty = true;

}

/\*\*

\* Добавляет новую команду

\*

\* @param newCommand

\* - новая команда

\*/

public void addCommand(Command newCommand) {

if (newCommand != null) {

commands.add(newCommand);

isEmpty = false;

}

}

/\*\*

\* Выполяет команду

\*

\* @throws IncorrectParameterListException

\* @throws InterruptedException

\*/

private void executeCommand() throws IncorrectParameterListException,

InterruptedException {

Command currentCommand = commands.remove();

if (validator.checkCommand(currentCommand)) {

currentCommand.execute();

} else {

throw new IncorrectParameterListException(currentCommand);

}

}

/\*

\* Выполяет все команды в очереди, если очередь пуста - ожидает

\*/

@Override

public void run() {

while (!Thread.currentThread().isInterrupted()) {

try {

while (commands.size() > 0)

try {

executeCommand();

} catch (IncorrectParameterListException e) {

e.printStackTrace();

}

isEmpty = true;

Thread.sleep((long) 100);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

**package** view;

**import** initialization.Application;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.border.EmptyBorder;

**import** javax.swing.JTable;

**import** javax.swing.ListSelectionModel;

**import** javax.swing.table.AbstractTableModel;

**import** model.AllGroups;

**import** model.Groups;

**import** model.User;

**import** model.event.EventAddGroup;

**import** model.event.EventDeleteGroup;

**import** model.event.EventUpdateGroup;

**import** model.event.ModelEvent;

**import** java.awt.event.WindowAdapter;

**import** java.awt.event.WindowEvent;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.Comparator;

**import** java.util.List;

**import** java.awt.event.MouseAdapter;

**import** java.awt.event.MouseEvent;

/\*\*

\* Сортує назви груп за алфавітом

\*

\*/

**class** AlphabeticSortGroups **implements** Comparator<Groups> {

@Override

**public** **int** compare(Groups o1, Groups o2) {

**return** o1.getName().compareTo(o2.getName());

}

}

/\*\*

\* Сортує користувачів по прізвищу за алфавітом

\*

\*/

**class** AlphabeticSortUsers **implements** Comparator<User> {

@Override

**public** **int** compare(User o1, User o2) {

**return** o1.getSurname().compareTo(o2.getSurname());

}

}

/\*\*

\* Вікно для роботи з групами

\*

\*/

**public** **class** ExpertGroups **extends** JFrame **implements** ModelListener {

**private** JFrame thisFrame;

**private** **static** **final** **long** *serialVersionUID* = 1L;

**private** JPanel contentPane;

**private** GroupPanel groupPanel;

**private** JTable tableGroups;

**private** Menu menu;

**private** List<Groups> groups;

**private** **int** indexShowGroup;// індекс групи,що відображається

**private** AllGroups allGroups = (AllGroups) Application.*INSTANCE*.DATA

.get(Groups.**class**.getName());

/\*\*

\* Create the frame.

\*/

**public** ExpertGroups() {

thisFrame = **this**;

addWindowListener(**new** WindowAdapter() {

@Override

**public** **void** windowClosing(WindowEvent e) {

Application.*INSTANCE*.openFrame.put(thisFrame, **false**);

thisFrame.setVisible(**false**);

}

});

setTitle("Управління експертними групами");

setDefaultCloseOperation(JFrame.*DO\_NOTHING\_ON\_CLOSE*);

setBounds(100, 100, 867, 670);

contentPane = **new** JPanel();

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

menu = **new** Menu();

**this**.setJMenuBar(menu);

groupPanel = **new** GroupPanel();

groupPanel.setEditable(**false**);

groupPanel.setBounds(301, 0, 562, 600);

getContentPane().add(groupPanel);

ModelTableGroups modelTableGroups = **new** ModelTableGroups();

// modelTableGroups.addTableModelListener(l)

tableGroups = **new** JTable(modelTableGroups);

tableGroups.setBounds(0, 0, 300, 589);

tableGroups.setSelectionMode(ListSelectionModel.*SINGLE\_SELECTION*);

tableGroups.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

showGroup(tableGroups.getSelectedRow());

}

});

tableGroups.getSelectionModel().setSelectionMode(

ListSelectionModel.*SINGLE\_SELECTION*);

contentPane.add(tableGroups);

ContextMenuGroups contextMenu = **new** ContextMenuGroups(**this**);

contextMenu.addPopup(tableGroups);

showGroup(0);

}

**private** **void** showGroup(**int** indexGroup) {

indexShowGroup = indexGroup;

Groups showGroup = **null**;

**if** ((indexShowGroup > -1) && (indexShowGroup < groups.size()))

showGroup = groups.get(indexShowGroup);

groupPanel.showGroup(showGroup);

}

/\*\*

\* Модель таблицы для отображения груп

\*

\*/

**class** ModelTableGroups **extends** AbstractTableModel {

**private** **static** **final** **long** *serialVersionUID* = 1L;

**public** ModelTableGroups() {

update();

}

@Override

**public** **int** getRowCount() {

**return** groups.size();

}

@Override

**public** **int** getColumnCount() {

**return** 1;

}

@Override

**public** Object getValueAt(**int** rowIndex, **int** columnIndex) {

**return** groups.get(rowIndex).getName();

}

@Override

**public** String getColumnName(**int** column) {

**return** "Групи";

}

**public** **void** update() {

groups = **new** ArrayList<Groups>(allGroups.getAll());

Collections.*sort*(groups, **new** AlphabeticSortGroups());

}

}

/\*\*

\* Повертає список груп

\*

\* **@return** список груп

\*/

**public** List<Groups> getGroups() {

**return** groups;

}

/\*\*

\* Повертає індекс групи,яка детально відображається

\*

\* **@return** индекс групи

\*/

**public** **int** getIndexShowGroup() {

**return** indexShowGroup;

}

/\*\*

\* Повертає головне меню

\*

\* **@return** головне меню

\*/

**public** Menu getMenu() {

**return** menu;

}

/\*\*

\* Оброблює подію зміни групи

\*

\* **@param** updateGroup

\* змінена група

\*/

**private** **void** updateGroup(Groups updateGroup) {

**int** indexUpdateGroup = groups.indexOf(updateGroup);

**if** (indexUpdateGroup > -1) {

**if** (indexShowGroup == indexUpdateGroup) {

showGroup(indexUpdateGroup);

}

tableGroups.updateUI();

}

}

**private** **void** deleteGroup(Groups deleteGroup) {

groups.remove(deleteGroup);

**if** (indexShowGroup >= groups.size()) {

indexShowGroup = groups.size() - 1;

}

tableGroups.updateUI();

showGroup(indexShowGroup);

}

**private** **void** addGroup(Groups addGroups) {

groups.add(addGroups);

Collections.*sort*(groups, **new** AlphabeticSortGroups());

tableGroups.updateUI();

showGroup(groups.indexOf(addGroups));

}

@Override

**public** **void** acceptEvent(ModelEvent event) {

Class<? **extends** ModelEvent> classEvent = event.getClass();

**if** (classEvent == EventUpdateGroup.**class**) {

updateGroup(((EventUpdateGroup) event).getGroup());

}

**if** (classEvent == EventDeleteGroup.**class**) {

deleteGroup(((EventDeleteGroup) event).getDeleteGroup());

}

**if** (classEvent == EventAddGroup.**class**) {

addGroup(((EventAddGroup) event).getGroups());

}

}

}