Державний університет телекомунікацій
Навчально-науковий інститут телекомунікацій та інформатизації
Кафедра комутаційних систем

Методичне керівництво для виконання лабораторної роботи №1

Знайомство з мовою Java. Основи об'єктно-орієнтованого програмування.

з дисципліни «Програмне забезпечення телекомунікаційних систем».
освітньо-кваліфікаційного рівня магістр

Київ - 2013

Укладачі: проф. каф. КС Кунах Н.І.
асистент. каф. КС Невдачина О.В.

Методичні вказівки обговорені і схвалені
на засіданні кафедри КС

Протокол № _____
від «___» _____ 2013р.

Тема: Знайомство з мовою Java. Основи об'єктно-орієнтованого програмування.

Мета заняття: Вивчення базових основ мови програмування Java. Отримання основних навичок написання найпростіших програм мовою Java.

Час заняття: 90 хвилин.

Список літератури.

1. Гребешков А.Ю. «Микропроцессорные системы и программное обеспечение в средствах связи», ПГУТИ Самара, 2009. (надається в електронному вигляді).
2. Голицина О.Л., Партыка Т.Л. «Програмное обеспечение», 2-ое издание, Москва, 2008. (надається в електронному вигляді).

Зміст заняття.

1. **Ознайомча частина.**

Перевірка присутності студентів, оголошення теми заняття та порядок його проведення.

2. Завдання для виконання.

Інтегроване середовище розробки Java програм Eclipse

Програма Eclipse є інтегрованою середовищем розробки (Integrated Development Environment, скорочено IDE). Основним розробником Eclipse є фірма IBM. Програма вільно доступна за адресою http://www.eclipse.org

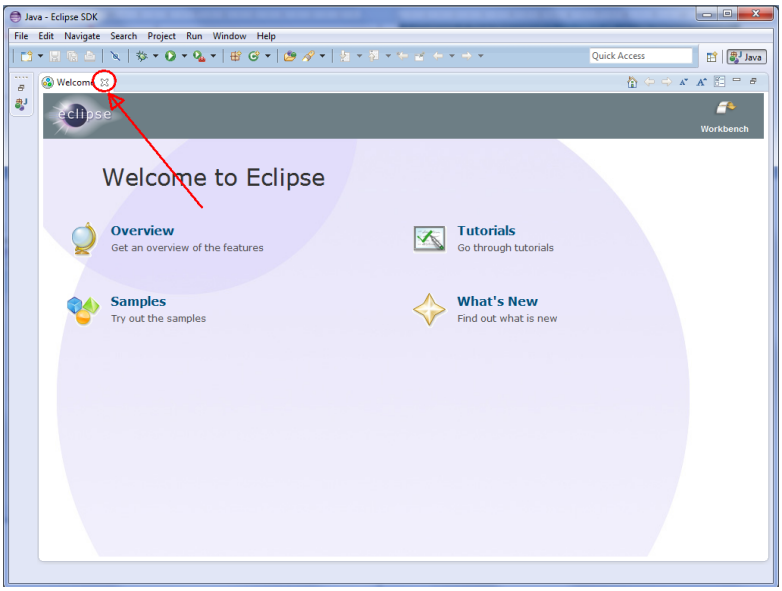
Установка Eclipse не вимагає інсталяції додаткових компонентів. У вибраному при інсталяції каталозі буде побудована структура підкаталогів необхідних для роботи середовища Eclipse.

Завдання для лабораторної роботи:

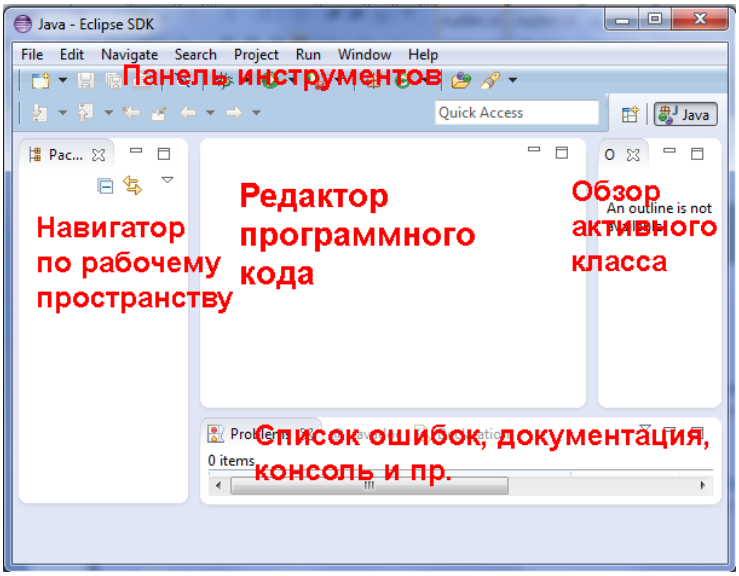
1. Установка і запуск середовища Eclipse

Для запуску Eclipse потрібна наявність встановленої середовища для виконання Java програм (JRE). Запускається файл називається eclipse.exe і знаходиться в установчому каталозі Eclipse.

Після установки програми, Ви побачите вікно привітання, переглянувши яке закрийте вікно хрестиком.



На екрані з'являється безпосередньо середу розробки Eclipse, в якій ми будемо працювати.



Налаштування робочого місця середовища Eclipse доступні в пункті меню Window > Preferences > Workbench.

Меню користувача Eclipse

Меню користувача містить команди, згруповані по функціональності.

File – команди для роботи з файлами: створення, запис, друк файлу і т.д.

Edit – команди текстового редактора: копіювання, видалення, вставка тексту і т.д.

Source – команди форматування, документування і генерації вихідного коду.

Refactor – команди для перетворення вихідного коду.

Navigate – команди для навігації в середовищі розробки.

Search – команди для пошуку в системі.

Project – команди для складання проекту і установки його властивостей.

Run – команди для запуску програм.

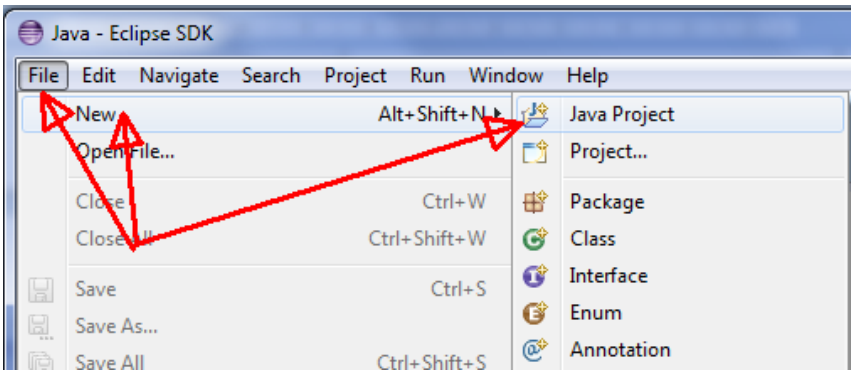
Window – містить пункти для налаштування різних властивостей середовища розробки.

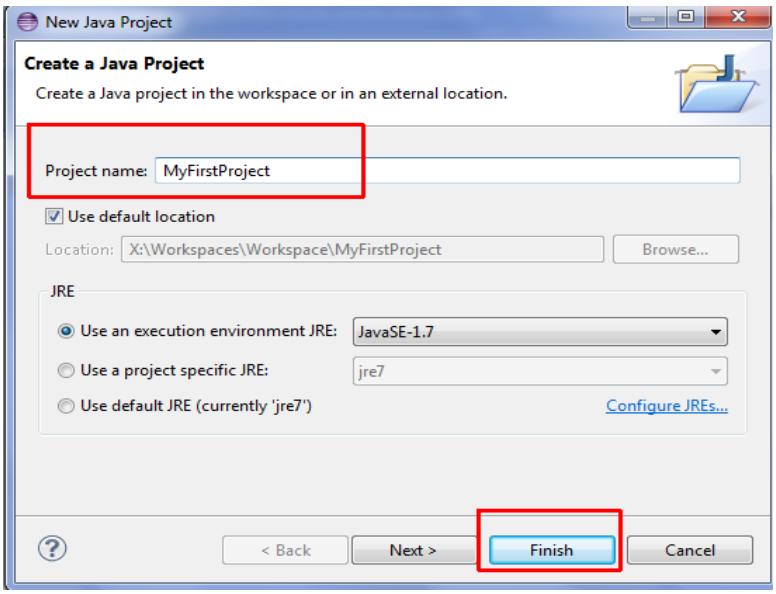
Help – команди виклику допомоги.

2. Створення першого Java проекту.

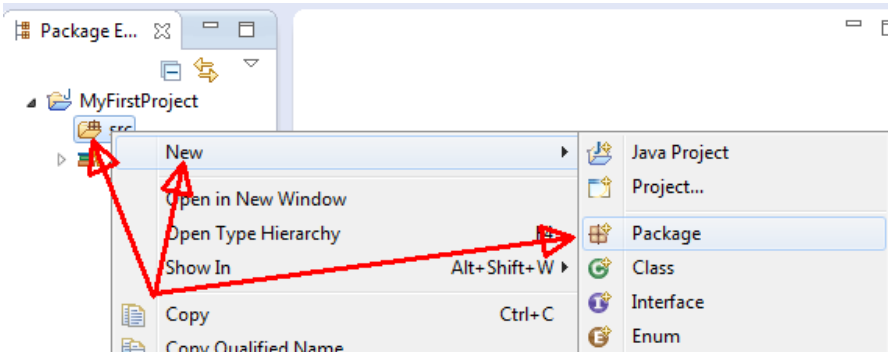
Перш ніж приступити до програмування, необхідно створити проект, в якому Eclipse буде зберігати всі ресурси, пов'язані з вашій програмі.

1. Для створення проекту виконайте команду File --> New --> Project. У вікні виберіть Java Project та натисніть «Далі». Вкажіть ім'я свого проекту. Зверніть увагу, що в директорії, вказаної вами як робочий простір, буде створена папка з ім'ям вашого проекту.

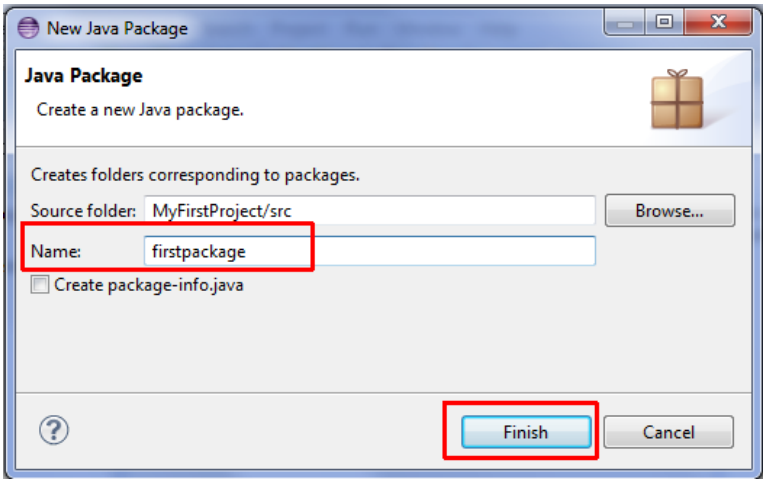




2. Створіть java-пакет (package) і надайте йому ім'я. Згідно з прийнятими угодами ім'я пакета має складатися з англійських малих літер. Для цього клацніть правою клавішею миші по папці «src», а потім виберіть «New / Package».



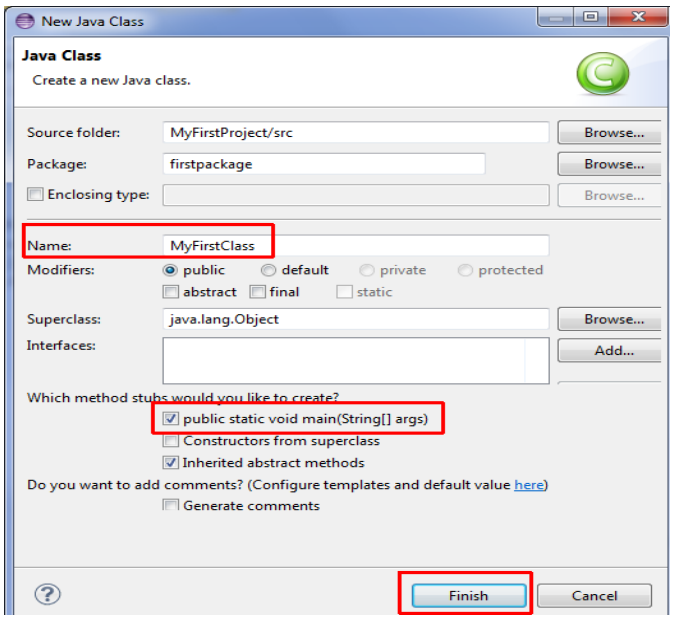
Після відкриття діалогового вікна введіть ім'я свого першого java-пакета та натисніть Finish.



3. Створіть Class.

Програма на Java завжди складається з одного або декількох класів. Створити клас можна командою New -> Class в контекстному меню уявлення Navigator (або Package Explorer, не має значення). При створенні класу необхідно вибрати пакет, до якого він буде ставитися (виберіть щойно створений вами пакет) і придумати йому ім'я. Імена класів прийнято починати з великої літери. Якщо не дотриматися це правило хорошого тону, Eclipse видасть попередження, але нічого страшного не станеться.

Для наших цілей корисно поставити галочку в розділі «Які методи ви хочете со ¬ здать в своєму класі?» Навпроти опції public static void main (String [] args). У результаті в тілі класу згенерує метод (функція) main (). Java вимагає, щоб хоча б в одному з класів програми існував метод з таким заголовком. Саме він і буде виконаний при старті програми.



В результаті наших дій в папці пакета буде створений файл з ім'ям нашого класу і розширенням. Java. Eclipse відкриє редактор коду, в якому відобразиться вміст цього файлу. Воно буде приблизно таким:

```
package firstpackage;
      public class MyFirstClass {


    /**
    * @param args
    */
    public static void main(String[] args)
 {
// TODO Auto-generated method stub
    }
}
```

Команди, складові тіло функції, можна написати замість автоматично сгене ¬ рировать коментаря / / TODO Auto-generated method stub. Ми напишемо тільки одну команду, яка буде виводити на екран класичну рядок «Hello, world!»:

System.out.println("Hello, world!");

Залишилося програму запустити. Для цього виконаємо команду Run -> Run і отримаємо діалогове вікно з нетривіальними настройками запуску. У лівій частині цього вікна треба вибрати Java Application (додаток Java). Трохи подумавши, Eclipse знайде наш клас, що містить метод main () і запропонує

почати запуск програми саме з нього (у правій частині вікна на вкладці Main повинні з'явитися назви нашого проекту і нашого класу). Крім цього увазі програміста пропонується ще кілька закладок. Наприклад, на другому з них - Arguments - пропонується ввести параметри командного рядка (якщо програма розрахована на виклик з командного рядка з параметрами). Для нашої простої програми нічого додатково вказувати не потрібно. Просто натисніть кнопку Run.

В результаті роботи програми здійснюється виведення даних у так звану консоль. В операційній системі MS DOS консоллю служив весь екран монітора. Eclipse ж відкриває нам уявлення Console, у якому (якщо все зроблено правильно) і відобразиться рядок "Hello, world!" - Результат виведення нашої програми.

Тепер для повторного запуску програми (наприклад, якщо ми вирішили внести в неї якісь зміни або треба показати викладачеві), можна піти легшим шляхом - виконати команду Run -> Run Last Launched (ще раз запустити попередній додаток) або просто натиснути Ctrl + F11.

Докладний розбір програми.

Строка 2

class MyFirstClass {

У цьому рядку використано зарезервоване слово class. Воно говорить транслятору, що ми збираємося описати новий клас. Повний опис класу розташовується між що відкриває фігурною дужкою в першому рядку і парної їй закриває фігурної дужкою.

Строка 3

public static void main (String args []) {

Така, на перший погляд, надмірно складна рядок прикладу є наслідком важливого вимоги, закладеного при розробці мови Java.

public

Це - модифікатор доступу, який дозволяє програмісту керувати видимістю будь-якого методу і будь змінної. У даному випадку модифікатор доступу public означає, що метод main видно і доступний будь-якому класу. Існують ще 2 покажчика рівня доступу - private і protected

static

Наступне ключове слово - static. За допомогою цього слова оголошуються методи й змінні класу, використовувані для роботи з класом в цілому. Методи, в оголошенні яких використано ключове слово static, можуть безпосередньо працювати тільки з локальними і статичними змінними.

void

У вас нерідко виникатиме потреба в методах, які повертають значення того чи іншого типу: наприклад, int для цілих значень, float - для речових або ім'я класу для типів даних, визначених програмістом. У нашому випадку потрібно просто вивести на екран рядок, а повертати значення з методу main не потрібно. Саме тому і був використаний модифікатор void.

main

Всі існуючі реалізації Java-інтерпретаторів, отримавши команду інтерпретувати клас, починають свою роботу з виклику методу main. Java-транслятор може оттранслировать клас, в якому немає методу main. А ось Java-інтерпретатор запускати класи без методу main не вміє.

Всі параметри, які потрібно передати методу, вказуються всередині пари круглих дужок у вигляді списку елементів, розділених символами ";" (крапка з комою). Кожен елемент списку параметрів складається з розділених пропуском типу та ідентифікатора. Навіть якщо у методу немає параметрів, після його імені все одно потрібно поставити пару круглих дужок. У прикладі, який ми зараз обговорюємо, у методу main тільки один параметр.

Елемент String args[]оголошує параметр з ім'ям args, який є масивом об'єктів - представників класу String. Зверніть увагу на квадратні дужки, що

стоять після ідентифікатора args. Вони говорять про те, що ми маємо справу з масивом, а не з одиночним елементом зазначеного типу.

Строка 4

System. out. prlntln("Hello World!");

У цьому рядку виконується метод println об'єкта out. Об'єкт out оголошений в класі OutputStream і статично ініціалізується в класі System. Закриває фігурною дужкою закінчується оголошення методу main, а така ж наступна дужка завершує оголошення класу MyFirstClass.