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

Методичне керівництво для виконання практичної роботи

Об'єктно-орієнтоване програмування.Мова Java.

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

Київ - 2014

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. **Матеріал для вивчення:**

Java— об'єктно-орієнтована мова програмування, розроблений компанією Sun Microsystems. Програми Java зазвичай компілюються в спеціальний байт-код, тому вони можуть працювати на будь віртуальній Java-машині (JVM) незалежно від комп'ютерної архітектури. Дата офіційного випуску - 23 травня 1995.

Переваги мови Java:

1. Одна з основних переваг мови Java - незалежність від платформи, на якій виконуються програми: один і той же код можна запускати під управлінням операційних систем Windows, Solaris, Linux, Machintosh та ін

Це дійсно необхідно, коли програми завантажуються через Інтернет для подальшого виконання під управлінням різних операційних систем.

2. Синтаксис мови Java схожий на синтаксис мови C + +, і програмістам, що знають мови C і C + +, його вивчення не складає труднощів.

3. Крім того, Java - повністю об'єктно-орієнтована мова, навіть більшою мірою, ніж C + +. Всі сутності в мові Java є об'єктами, за винятком небагатьох основних типів (primitive types), наприклад чисел. (За допомогою Об'єктно-орієнтованого програмування легко розробляти складні проекти.)

4. Висока надійність. Творці забезпечили мову Java засобами, що дозволяють виключити саму можливість створювати програми, в яких були б приховані найбільш поширені помилки. Для цього в мові Java зроблено наступне.

- *Виключена можливість явного виділення і звільнення пам'яті.* Пам'ять у мові Java звільняється автоматично за допомогою механізму збірки сміття. Програміст гарантований від помилок, пов'язаних з неправильним використанням пам'яті.

- *Введені істинні масиви і заборонена арифметика покажчиків.*

Тепер програмісти в принципі не можуть стерти дані з пам'яті внаслідок неправильного використання покажчиків.

- *Виключена можливість переплутати оператор присвоювання з оператором порівняння на рівність.* Тепер не можна навіть скомпілювати вираз if (ntries = 3). . . (програмісти мовою Visual Basic можуть взагалі не помітити тут жодної проблеми, оскільки ця помилка - джерело більшості непорозумінь в мовах C і C + +). - Исключено множественное наследование. Оно заменено новым понятием — интерфейсом, позаимствованным из языка Objective C.

Інтерфейс дає програмісту майже все, що той може отримати від множинного спадкоємства, уникаючи при цьому складнощів, що виникають при управлінні ієрархіями класів.

Лексика та синтаксис мови Java.

Програми на Java - це набір прогалин, коментарів, ключових слів, ідентифікаторів, літеральних констант, операторів і роздільників.

Пробіли

Java - мова, яка допускає довільне форматування тексту програм. Для того, щоб програма працювала нормально, немає ніякої необхідності вирівнювати її текст спеціальним чином. Наприклад, клас HelloWorld можна було записати в двох рядках або будь-яким іншим способом, який буде вам до душі. І він працюватиме точно так само за умови, що між окремими лексемами (між якими немає операторів або роздільників) є принаймні по одному пробілу, символу табуляції або символу перекладу рядка.

В даний час зарезервовано 59 ключових слів. Ці слова, у поєднанні з синтаксисом операторів і роздільниками, утворюють основу мови Java. Їх не можна використовувати ні в якості ідентифікаторів, ні для імен змінних, класів або методів.

abstract	continue	for	new	switch
assert	default	goto *	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws

case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const *	float	native	super	while

Слова byvalue, cast, const, future, generic, goto, inner, operator, outer, rest, var зарезервовані у Java, но пока не используются Кроме этого, в Java есть зарезервированные имена методов (эти методы наследуются каждым классом, их нельзя использовать, за исключением случаев явного переопределения методов класса Object).

Зарезервовані імена методів Java

clone	equals	finalize	getClass	hashCode
notify	notifyAll	toString	wait	

Роздільники

У мові Java є символи, які використовуються як роздільники. Найчастіше використовується символ крапка з комою. Однак він не єдиний.

()	круглі дужки	Використовуються для передачі списків параметрів у визначеннях і викликах методів. Їх застосовують також для визначення пріоритету у виразах, виразів в керуючих операторах і перетворенні типів
{}	фігурні дужки	Використовуються для вказівки значень автоматично ініціалізіруемих масивів. Їх застосовують також для визначення блоків коду, класів, методів і локальних областей видимості
[]	квадратні	Використовуються для оголошення типів масивів, а також при зверненні

	дужки	до елементів масивів
;	Крапка з комою	завершує оператори
,	Кома	Розділяє послідовні ідентифікатори в оголошеннях змінних. Цей символ також використовується для створення ланцюжків операторів усередині оператора for
.	Крапка	Використовується для відділення імен пактів від імен вкладених пакетів і класів, а також для відділення змінної або методу від посилальної змінної

Ідентифікатори

Ідентифікатори використовуються для іменування класів, методів і змінних. Ідентифікатором може служити будь-яка послідовність малих і великих літер, цифр або символів підкреслення (_) або долара ($). Для загального використання символ долара не призначається. Ідентифікатори не повинні починатися з цифри. Мова Java чутливий до регістру символів, тому ідентифікатори FIRST і First - є різними. Варто також відзначити, що оскільки в Java повною мірою застосовується Unicode, то в якості «букви» може використовуватися також символ з вірменського, корейського, грузинського, індійського та практично будь-якого алфавіту.

Коментарі.

Коментарі - це пояснюють написи, які використовують програмісти для поліпшення зрозумілості коду. При компіляції програми коментарі ігноруються, тому в них може бути написано все, що завгодно. Головне показати, що даний напис є коммен ¬ таріем і її не треба намагатися трактувати як команди програми. У Java це робиться одним з наступних образів:

1. 1. Ставляться дві косі риси / /. З цього моменту і до кінця рядка можна писати все, що завгодно - Java буде вважати це коментарем.

2. 2. На початку коментаря ставляться символи / *, а в кінці - * /. У цьому випадку коментар може займати яке завгодно кількість рядків.

3. Особливо виділяються коментарі для документування, які поміщаються між маркерами / ** і * /. Про їх використання буде розказано пізніше.

Відступи.

Java - мова вільної форми. При написанні програми не потрібно слідувати ніяким спеціальним правилам щодо відступів. Єдина обов'язкова вимога - наявність, щонайменше, одного пробілу між лексемами, які не розмежовано оператором або роздільником. У Java відступами є символи пробілу, табуляції або символу нового рядка.

Літерал — константа деякого типу, яка не має імені, а зазначена власне записом, наприклад: 0, 145, 'a', "Hello, world!".

Оператор — найменша автономна частина мови програмування (також звана інструкцією або командою) здатна виконуватися самостійно і несуча самостійний сенс. Прикладом може служити оператор виводу на екран або оператор циклу. Програма звичайно являє собою послідовність операторів (команд, інструкцій).

Операція — дія над однією або більше (частіше всього двома) величинами - операндами, що приводить до появи нової величини - результату операції. Операція, як правило, не має сенсу поза контекстом-якого оператора. Наприклад, безглуздо писати 3 +4 як самостійну команду, оскільки обчислений результат операції тут же буде втрачено. Але команда x = 3 +4; (використання результату операції в операторі присвоювання) або System.out.println (3 +4); (передача результату в метод як параметр) цілком виправдана.

Операнд — величина, що бере участь в операції.

Метод (функція) - Частина програми, що має власне ім'я. Це ім'я можна використовувати в програмі як команду (така команда називається

викликом методу). При виклику методу виконуються команди, з яких він складається. Метод аналогічно операції може повертати значення-результат.

Вираз – це послідовність операцій і викликів методів, які виконуються в певному порядку (по пріоритету операцій, з урахуванням дужок), що дає при обчисленні деяке значення.

Змінна – іменована область пам'яті ЕОМ, в якій програма може зберігати дані певного типу (звані значенням змінної) і звертатися до цих даних, використовуючи ім'я змінної.

Основи основ синтаксису Java

1. ***Мова Java розрізняє прописні і малі літери.*** Це означає, що імена всіх функцій і ключові слова слід записувати в точності так, як вони значаться в прикладах і довідниках.
2. ***Кожна команда (оператор) в мові Java повинна закінчуватися крапкою з комою..***
3. Програма на Java складається з одного або декількох класів. Абсолютно вся функціональна частина програми (тобто те, що вона робить) повинна бути поміщена в методи тих чи інших класів. (Клас і метод, як поняття об'єктно-орієнтованого програмування, будуть розглядатися в третьому занятті. Там же буде розглянуто синтаксис класів. У перших вправах використовуйте класи, які за умовчанням генерує Eclipse.)
4. ***Класи групуються в пакети***.
5. Хоча б в одному з класів повинен існувати метод main (), в точності такий, як у розглянутому нами прикладі. (На перших порах розбиратися або намагатися запам'ятати правильне написання цього

методу необов'язково - Eclipse все згенерує сам, якщо поста ¬ вить потрібну галочку.) Саме цей метод і буде виконуватися першим.

У найпростішому випадку програма може складатися з одного (або навіть жодного) пакета, одного класу всередині пакету і єдиного методу main () усередині класу. Команди програми будуть записуватися між рядком

public static void main (String [] args) {

і закриває фігурною дужкою}, позначає закінчення тіла методу. Цього підходу і слід дотримуватися при виконанні простих вправ.