МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

Одеський національний політехнічний університет

Інститут Комп’ютерних Систем

Кафедра Інформаційних Систем та Технологій

Протокол практичної роботи №1

з дисципліни об'єктно-орієнтоване програмування

на тему: «Початок роботи с Java SE»

Виконав студент групи

КН/ІПЗ-2хх

Іванов І.І.

Прийняв

Рудніченко М.Д.

Одеса, 2022

ЗМІСТ

[ВСТУП](#_ki7i591yqske) 3

[ТЕОРЕТИЧНА ЧАСТИНА](#_335513nxkbjl) 4

[ПРАКТИЧНА ЧАСТИНА](#_9cm35ikas1ko) 7

[ВИСНОВОК](#_xlxs31hqwy0n) 9

[ЛІТЕРАТУРА](#_5m79jhxx931e) 10

# 

# ВСТУП

Мета роботи - отримати загальні відомості про платформу JavaSE і про принципи розробки додатків на цій платформі. Навчитися користуватися документацією.

# ТЕОРЕТИЧНА ЧАСТИНА

Мова програмування Java була створена в рамках проекту корпорації Sun Microsystems по створенню комп'ютерних програмно-апаратних комплексів нового покоління. Перша версія мови була офіційно опублікована в 1995 році.

Програми, написані на мові Java, вдають із себе набори класів і зберігаються в текстових файлах з розширенням .java. При компіляції текст програми перекладається (транслюються) в двійкові файли з розширенням .class. Такі файли містять байт-код - інструкції для абстрактного Java-процесора у вигляді байтових послідовностей команд цього процесора і даних до них. Для того, щоб байт-код був виконаний на будь-якому комп'ютері, він повинен бути переведений в інструкції для відповідного процесора. Саме цим і займається Java-машина. Спочатку байт-код завжди інтерпретувався: кожен раз, як зустрічалася якась інструкція Java-процесора, вона переводилася в послідовність інструкцій процесора комп'ютера. Природно, це значно уповільнювало роботу додатків Java.

Завдяки компіляції програм Java в платформонезалежний байт-код забезпечується переносимість цих програм не тільки на рівні вихідного коду, а й на рівні скомпільованих додатків. Звичайно, при цьому на комп'ютері, де виконується додаток, повинна бути встановлена ​​програма віртуальної Java-машини (Java Virtual Machine - JVM), скомпільована в коди відповідного процесора (native code - "рідний" код). На одному і тому ж комп'ютері може бути встановлено кілька Java-машин різних версій або від різних виробників.

SDK (від англ. Software development kit) - комплект засобів розробки, який дозволяє фахівцям з програмного забезпечення створювати додатки для певного пакету програм, програмного забезпечення базових засобів розробки, апаратної платформи, комп'ютерної системи, ігрових консолей, операційних систем і інших платформ.

Java Development Kit (скорочено JDK) - безкоштовно поширюваний компанією Oracle Corporation (раніше Sun Microsystems) комплект розробника додатків на мові Java, що включає в себе компілятор Java (javac), стандартні бібліотеки класів Java, приклади, документацію, різні утиліти і виконавчу систему Java (JRE). До складу JDK не входить інтегроване середовище розробки на Java, тому розробник, що використовує тільки JDK, змушений використовувати зовнішній текстовий редактор і компілювати свої програми, використовуючи утиліти командного рядка.

Все современные интегрированные среды разработки приложений на Java, такие, как JDeveloper, NetBeans IDE, Sun Java Studio Creator, IntelliJ IDEA, Borland JBuilder, Eclipse, опираются на сервисы, предоставляемые JDK. Большинство из них для компиляции Java-программ используют компилятор из комплекта JDK.

Всі сучасні інтегровані середовища розробки додатків на Java, такі, як JDeveloper, NetBeans IDE, Sun Java Studio Creator, IntelliJ IDEA, Borland JBuilder, Eclipse, спираються на послуги, що надаються JDK. Більшість з них для компіляції Java-програм використовують компілятор з комплекту JDK.

Доступні повні вихідні тексти JDK, включаючи вихідні тексти самого Java-компілятора javac. В обраному при інсталяції каталозі буде побудована структура підкаталогів пакета SDK. Коротко розглянемо цю структуру і з'ясуємо вміст і призначення всіх підкаталогів.

* bin \ - каталог інструментарію розробника
* demo \ - каталог з прикладами
* include \ - для взаємодії з програмами на C, C ++
* include-old \ - аналогічно, але попередня версія
* jre \ - каталог інструментарію користувача (то, що поставляється кінцевому користувачеві при установці (deployment) готового додатка)
* jre \ bin \ - Java-машина (и) (JVM)
* jre \ lib \ - бібліотеки Java для кінцевих користувачів + ряд настроювальних файлів
* lib \ - бібліотеки Java для розробників

Програми Java можно розділити на декілька основних категорій:

1. Додаток (application) - аналог "звичайної" прикладної програми.

2. Аплет (applet) - спеціалізована програма з обмеженими можливостями, що працює у вікні WWW-документа під управлінням браузера.

3. Сервлет (servlet) - спеціалізована програма з обмеженими можливостями, що працює в WWW на стороні сервера. Використовується переважно в рамках технології JSP (Java Server Pages - серверних Сторінок Java) для програмування WWW-документів з боку сервера.

4. Серверний додаток (Enterprise application) - призначено для багаторазового використання на стороні сервера.

5. Бібліотека (Java Class Library - бібліотека класів, або NetBeans Module - модуль платформи NetBeans) - призначена для багаторазового використання програмами Java

Версії Java 1.0 і 1.1 прийнято називати Java 1. Всі версії Java починаючи c 1.2 називають Java 2. Однак більш надійно класифікувати за номером SDK, так як мова Java для версії SDK 1.5 дуже помітно відрізняється за можливостями від мови Java для більш ранніх версій SDK - в нього додано велику кількість нових синтаксичних конструкцій, а також змінений ряд правил. Тому код, правильний в Java для версії SDK 1.5, може виявитися неправильним в Java для версії SDK 1.4. Не кажучи вже про Java для версії SDK 1.3 або 1.2. Крім того, нещодавно компанія Sun перестала використовувати в назвах комплектів програмного забезпечення термін Java 2 і походять від нього скорочення виду j2.

# 

# ПРАКТИЧНА ЧАСТИНА

(Вар. 19) 6. Из небольшого текста удалить все символы, кроме пробелов, не являющиеся буквами. Между последовательностями подряд идущих букв оставить хотя бы один пробел.

Код програми:

import java.util.Scanner;

public class HelloWorld {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String str = sc.nextLine();

String newstr = "";

boolean lastCharIsLetter = false;

for (int i = 0; i < str.length(); i++) {

char ch = str.charAt(i);

if (('A' <= ch && ch <= 'Z') || ('a' <= ch && ch <= 'z')) {

lastCharIsLetter = true;

newstr += ch;

} else {

if (ch == ' ' || lastCharIsLetter) {

newstr += ' ';

}

lastCharIsLetter = false;

}

}

System.out.println(newstr);

}

}

Результат роботи програми (приклад 1) наведено на рис.1.

Результат роботи програми (приклад 2) наведено на рис.2.



Риcунок 1 – Результат роботи програми (приклад 1)



Риcунок 2 – Результат роботи програми (приклад 2)

# 

# ВИСНОВОК

В ході роботи я отримав загальні відомості про платформу JavaSE і про принципи розробки додатків на цій платформі, а також навчився користуватися документацією.

# 

# ЛІТЕРАТУРА

1. <http://www.javable.com/tutorials/fesunov/lesson1/>
2. <https://ru.wikibooks.org/wiki/Java/Hello_World>!
3. <http://jarlad.blogspot.com/2011/04/java.html>
4. <http://www.intuit.ru/studies/courses/569/425/lecture/9667?page=4>
5. <http://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-142931.html>

Посилання на github: <https://github.com/acarrizo228/Java8>