

Міністерство освіти і науки України
Національний університет «Львівська політехніка»

LITTERIS ET ARTIBVS



**Знайомство з IDE Eclipse.
Створення та виконання простої Java-програми.
Програмування алгоритмів лінійної структури.
Консольний ввід числових даних та виконання простих
арифметичних обчислень.**

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних робіт № 1-2
з дисципліни «Мови програмування»
для студентів базового напрямку
6.040204 – "Прикладна фізика".

Затверджено
на засіданні кафедри
обчислювальної математики
та програмування
Протокол № 10 від 21.04.2016р.

Львів - 2016

Знайомство з IDE Eclipse. Створення та виконання простої Java-програми. Програмування алгоритмів лінійної структури. Консольний ввід числових даних та виконання простих арифметичних обчислень. Методичні вказівки до виконання лабораторних робіт № 1-2 для студентів базового напрямку 6.040204 «Прикладна фізика»/ Укл.: Гнатів Л.Б., Ментинський С.М., Угрин С.З., 2016. - 12с.

Укладачі:

Гнатів Л.Б., к. ф-м. н, доцент,
Ментинський С.М., ст. викл,
Угрин С.З., асистент.

Відповідальний за випуск: Угрин С.З..

Рецензент: Пукач П.Я. д.т.н., професор кафедри ВМ.

Передмова

У методичних вказівках розглянуто процес створення простої лінійної програми у вигляді консольного проекту на Java засобами інтегрованого середовища розробки програмного забезпечення Eclipse.

Методичні вказівки містять коротку довідкову інформацію, зразки виконання типових навчальних завдань, завдання до індивідуальної роботи та детальний опис їх виконання.

Методичні вказівки призначені для студентів базового напрямку 6.040204 «Прикладна фізика» і укладені відповідно до робочої навчальної програми з дисципліни «Мови програмування».

Лабораторна робота №1 - 2

Знайомство з IDE Eclipse. Створення та виконання простої Java-програми. Програмування алгоритмів лінійної структури. Консольний ввід числових даних та виконання простих арифметичних обчислень.

Мета роботи: набуття навичок практичної роботи в інструментальному середовищі Eclipse; ознайомлення з структурою програми на мові Java, елементами синтаксису мови та деякими типами даних на прикладі простої програми.

Короткі теоретичні відомості

Найпростіша програма на Java.

Мова програмування Java повністю ґрунтується на об'єктно-орієнтованому підході (об'єктно-орієнтованій парадигмі програмування). В об'єктно-орієнтованому програмуванні (ООП) програми утворюються з сукупності об'єктів, що взаємодіють між собою. Об'єкт є іменованою моделлю реальної сутності, що володіє конкретними значеннями своїх властивостей та визначеною поведінкою. Набір властивостей та поведінка об'єкта описуються в класі, на основі якого створюється об'єкт. З іншого боку, клас об'єднує у собі набір даних та методи їх обробки, а об'єкт є представником класу, який володіє конкретними значеннями описаних у класі даних. Навіть найпростіша програма на Java повинна складатися принаймні з одного класу. Наприклад, програма, що виводить на екран текст «Hello, world!», може мати вигляд:

```
public class HelloWorld {  
  
    public static void main(String[] args) {  
        System.out.println("Hello, world!");  
    }  
  
}
```

В першому рядку коду оголошується клас HelloWorld, з якого складається програма. Далі в фігурних дужках записана тіло класу, в якому є лише головний метод main, що містить єдину команду для виводу на консоль тексту «Hello, world!».

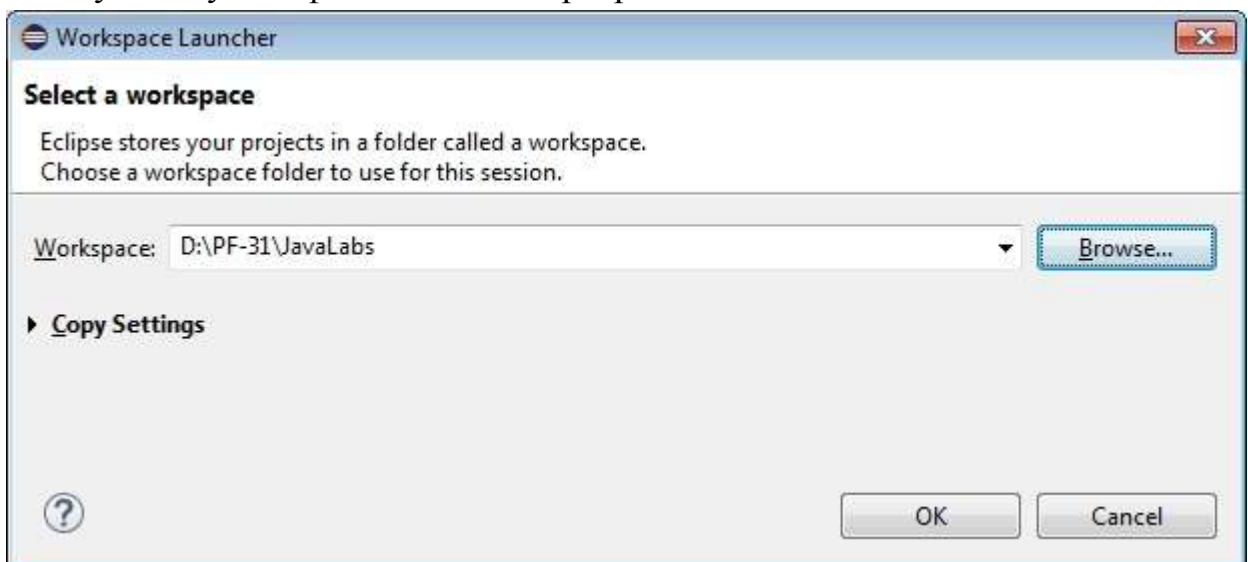
Інструментальні засоби.

Для того, щоб мати можливість створювати та виконувати Java-програми, на комп'ютері слід встановити пакет JDK (Java Development Kit). До його складу входять компілятор та бібліотеки стандартних класів, а також середовище JRE (Java Runtime Environment) для виконання байт-кодів, в які перетворює програми компілятор. JDK розповсюджується вільно, його можна завантажити, наприклад, за адресою .

JDK цілком достатньо для створення програмного забезпечення будь-якої складності, оскільки текст програми можна набрати навіть за допомогою звичайного текстового редактора. Проте набагато зручніше використовувати для цього спеціальні інтегровані середовища розробки програмного забезпечення – IDE (Integrated Development Environment). IDE окрім текстового редактора мають багато корисних інструментів, що дозволяють значно підвищити ефективність праці програміста. Про одне з поширених IDE Eclipse йтиме мова далі. Завантажити цю програму можна за адресою eclipse.org/download.

Створення програми в середовищі Eclipse

Запустити IDE Eclipse можна відкривши файл eclipse.exe з кореневої папки цієї програми, для швидкого запуску, зазвичай, створюють ярлик до цього файлу на робочому столі чи панелі задач. При першому запуску Eclipse вимагає встановити робочий простір (Workspace) – місце на диску куди записуватимуться файли з кодом програм.

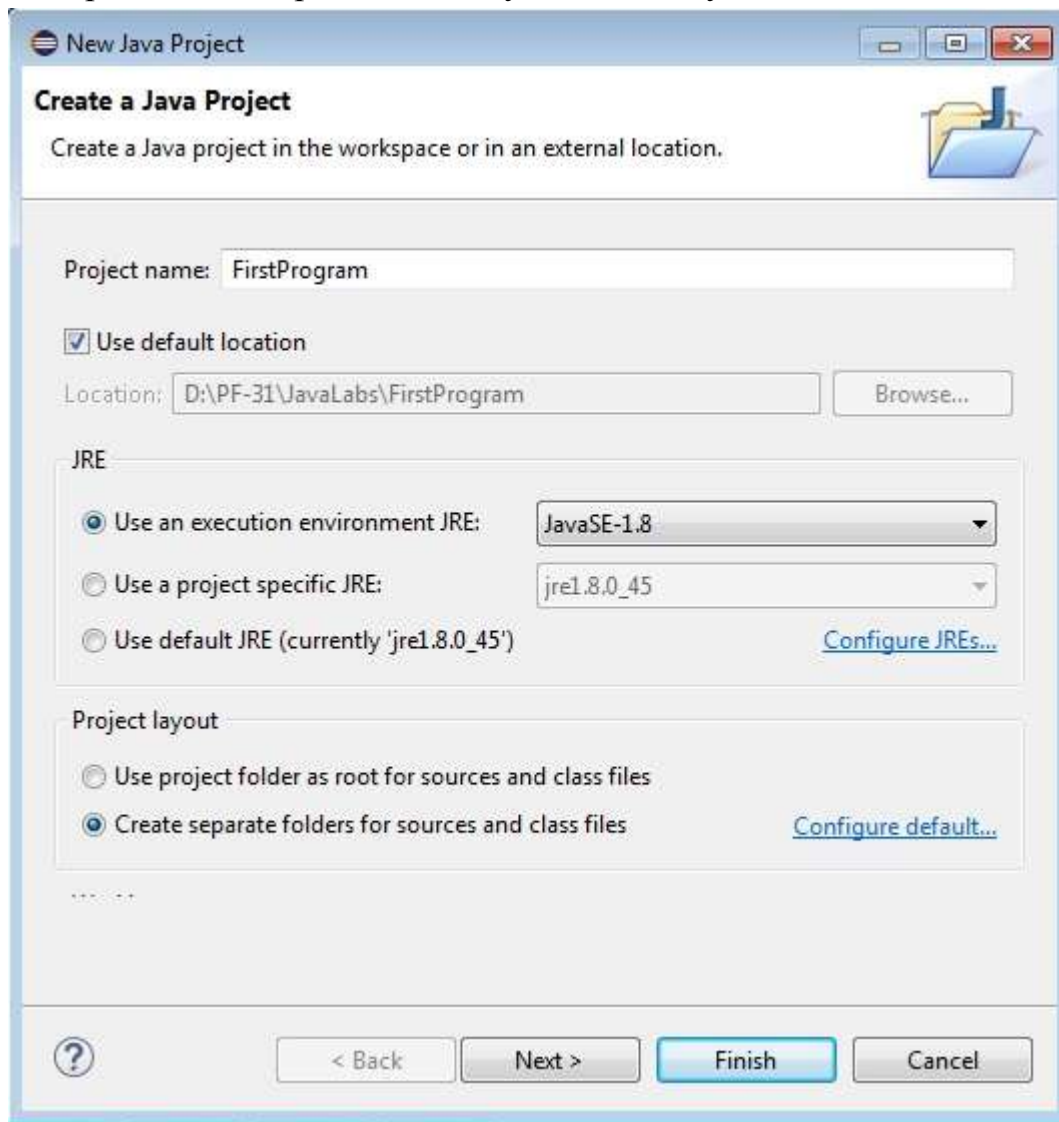


В якості Workspace можна вказати будь-яку папку на диску комп'ютера (при потребі її можна змінити пізніше). При повторних запусках Eclipse використовує вказану папку для робочого простору і таке діалогове вікно не з'являється.

Щоб набрати і виконати запропонований вище код простої Java-програми в Eclipse потрібно:

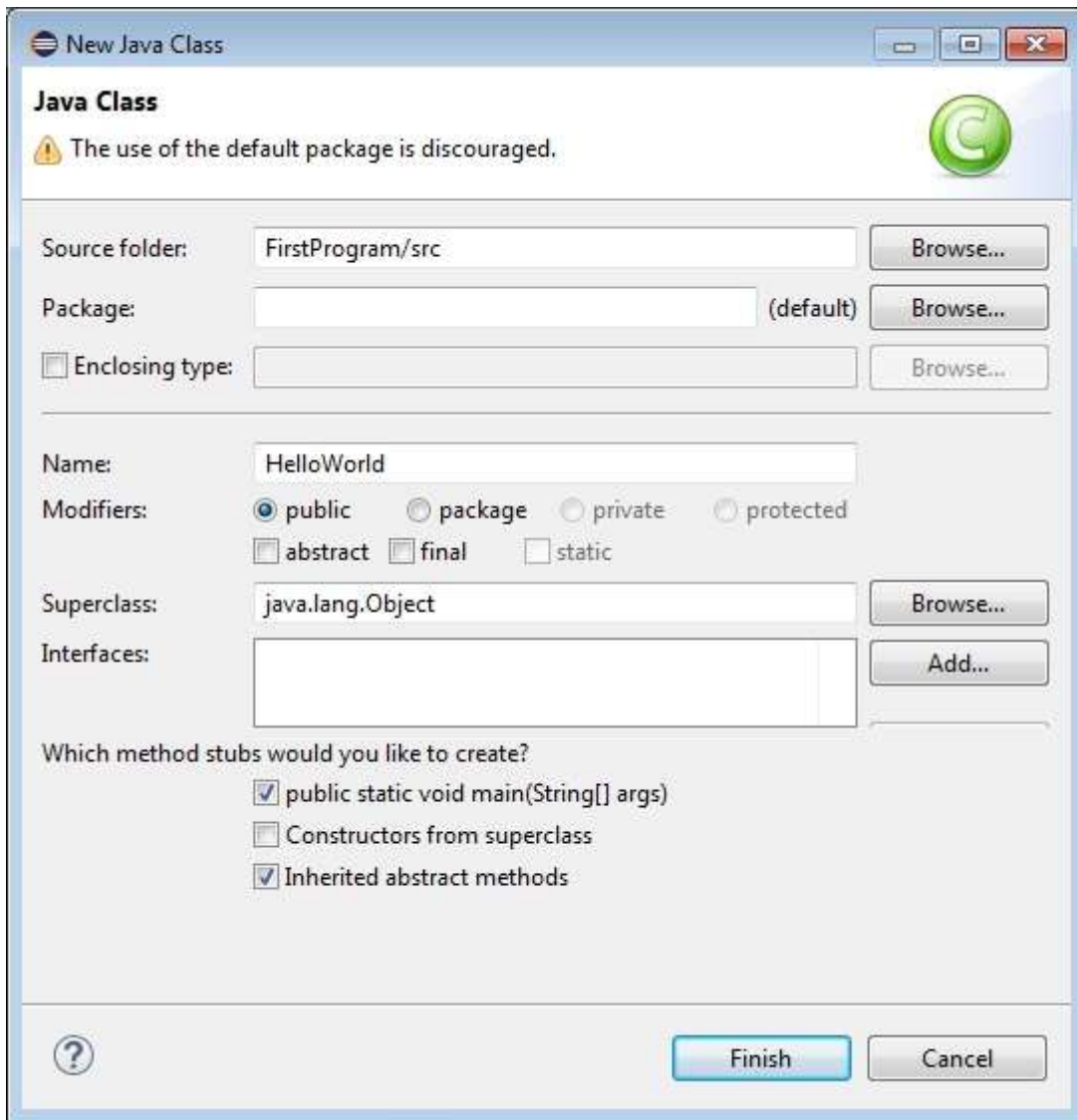
Запустити Eclipse;

Створити новий проект. Виконуємо команду File => New => Java Project.



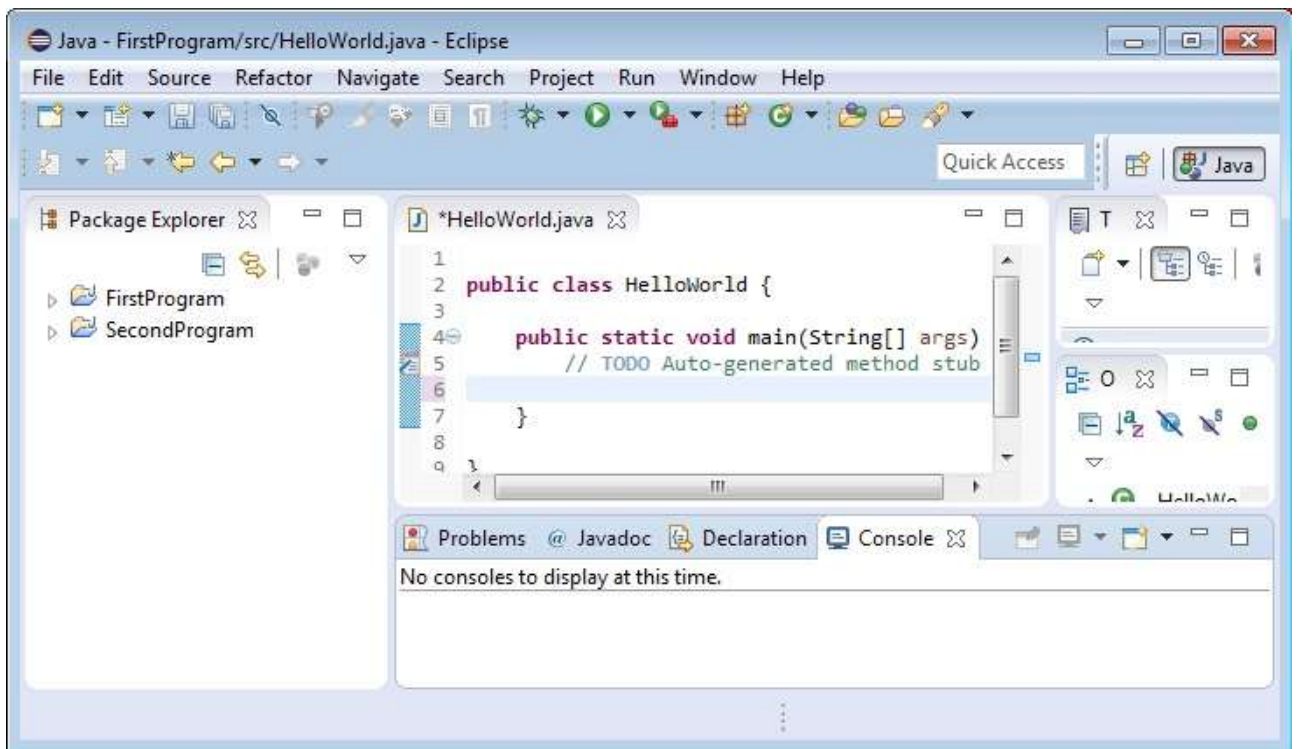
У цьому вікні достатньо задати ім'я проекту, а решту параметрів залишити без зміни і натиснути кнопку Finish.

Створити новий клас. Виконуємо команду File=>New=>Class



У діалоговому вікні створення нового класу задаємо його назву, ставимо позначку у прапорці «public static void main(String[] args)» та натискаємо кнопку Finish.

В результаті отримуємо проект з заготовкою Java-коду, в який залишається додати лише команду для виводу тексту на консоль.



Вікна Eclipse. Виконання програми.

Середовище розробки Eclipse має багатовіконний інтерфейс. Головне вікно у верхній частині містить рядок меню та панелі інструментів. Нижче, в робочій області, є декілька вікон, керувати відображенням яких можна через меню Window. Кожне внутрішнє вікно має кнопки керування, з допомогою яких його можна закрити чи розгорнути до розмірів головного вікна. За замовчуванням в робочій області Eclipse видимі такі вікна: справа Package Explorer, в якому у вигляді деревовидної структури відображаються пакети і класи відкритих в робочій області проектів; посередині – вікно з Java-кодом відкритого класу, зліва – вікна деяких додаткових інструментів Eclipse; внизу - вікно з вкладками, в яких відображається список помилок, консоль, тощо. Метод main будь-якого класу можна виконати як окрему програму безпосередньо із середовища Eclipse командою Run, або відповідною кнопкою на панелі інструментів. Команда Run запускає на виконання метод main того класу, код якого відкритий на екрані.

Використання змінних

Ускладнимо клас HelloWorld змінивши задачу наступним чином:
Написати програму, що запитує в користувача його ім'я, а потім виводить привітання, вставивши в нього введене ім'я.

При такій постановці задачі програма повинна реалізувати ввід та запам'ятовування даних, для цього потрібно використовувати змінні. Для опрацювання текстових даних таких як ім'я в Java використовується тип String. Описувати змінну необхідно перед її використанням, але це можна в будь-якому місці програми. Дозволяється, зокрема, описувати змінну безпосередньо перед присвоєнням їй значення. Наприклад, якщо вважати що нашу програму завжди використовуватиме користувач, якого звать Зеник, код класу HelloWorld, з використанням текстової змінної можна трансформувати так:

```
public class HelloWorld {
    public static void main(String[] args) {
        String userName = "Зеник";
        System.out.println("Привіт, " + userName + "!");
    }
}
```

Знак «+» в команді виводу використовується для з'єднання (конкатенації) частин тексту.

Ввід даних. Використання можливостей синтаксичного аналізатора Eclipse.

Запропонований вище код не розв'язує поставленої задачі, оскільки в ньому не реалізовано ввід даних. Для вводу даних з консолі використовується той же клас System та його статичне поле in. Щоправда, ввід даних через System.in має певну технічну специфіку і щоб реалізувати «людинозрозумілий» ввід даних доводиться використовувати допоміжні об'єкти (буфери) класів BufferedReader та InputStreamReader. В результаті щоб ввести з консолі значення текстової змінної Java-програмістам доводиться писати:

```
BufferedReader input =
    new BufferedReader(new InputStreamReader (System.in));
String userName = input.readLine();
```

Зрозуміло, що перед вводом імені, слід додати вивід пояснення

```
System.out.println("Як вас звати?");
```

щоб користувачеві було зрозуміло що від нього потребує програма.

Якщо ми додамо такий код до свого класу, то зауважимо що редактор коду Eclipse підкреслить назви класів BufferedReader та InputStreamReader, а зліва

від цієї стрічки виведе позначку що вказує на помилку в програмі (див. малюнок)



```
1
2 public class HelloWorld {
3
4     public static void main(String[] args) {
5         System.out.println("Як вас звати?");
6         BufferedReader input = new BufferedReader(new InputStreamReader (System.in));
7         String userName = input.readLine();
8         System.out.println("Привіт, " + userName + "!");
9     }
10 }
```

Помилка полягає в тому, що не всі класи доступні для безпосереднього використання в програмі (так як System чи String). Більшість класів, що використовуються в Java-програмах містяться в окремих пакетах, які потрібно підключати додатково. Цю роботу можна доручити IDE: натискаємо лівою клавішею миші на позначку помилки, і Eclipse дає нам перелік можливих способів вирішення проблеми.



Двічі клацаємо лівою клавішею миші на пункті «Import 'BufferedReader' (java.io)», після чого редактор коду додає стрічку

```
import java.io.BufferedReader;
```

перед заголовком класу HelloWorld. Так само робимо з класом InputStreamReader.

Ще одна помилка виникає в зв'язку з тим, що IDE Eclipse змушує програміста запобігати помилкам вводу даних. Тут слід вибрати пункт «Add throws declaration» зі списку можливих способів вирішення проблеми.

Нарешті отримаємо робочий код,

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
```

```

public class HelloWorld {

    public static void main(String[] args) throws IOException
    {
        System.out.println("Як вас звати?");
        BufferedReader input = new BufferedReader(
            new InputStreamReader (System.in));
        String userName = input.readLine();
        System.out.println("Привіт, " + userName + "!");
    }
}

```

перевіряємо його дію командою Run.

Ввід числових даних.

Доповнимо нашу програму роботою з числовими даними, наприклад, запитаємо в користувача його вік та встановимо і повідомимо його рік народження.

Для зберігання числової інформації в Java передбачено 7 стандартних (примітивних) типів. Для поставленої задачі про вік нам достатньо використовувати цілі змінні типу `int`. Наприклад:

```

int now = 2015;
System.out.println("Скільки вам років?");
int age = Integer.parseInt(input.readLine());
System.out.println("Ви народилися в " + (now - age) +
    " році.");

```

Оскільки команда `input.readLine()` зчитує з консолі дані у вигляді тексту (`String`), то доводиться виконувати перетворення введеної послідовності цифр в число типу `int`. Для цього застосовують метод `parseInt` класу `Integer`.

Для опрацювання дробових чисел можна використовувати примітивний тип `float` та перетворення введеного методом `parseFloat` класу `Float`, наприклад

```

System.out.println("Введіть масу тіла:");
float m = Float.parseFloat(input.readLine());

```

Слід зауважити, що слово `input` не є службовим словом Java, а в наведених фрагментах коду позначає екземпляр класу `BufferedReader`, через який

здійснюється ввід даних. З таким же успіхом замість `input` можна було використати, наприклад, `buffer` чи `reader`, просто `b`, чи взагалі `jack`.

Завдання до лабораторної роботи.

1. Створити просту Java-програму, що запитує в користувача ім'я, вітається з ним, питає його вік та виводить рік народження. **Вказівка:** виконати усі пункти створення програми, описані в теоретичних відомостях.
2. Вважаючи усі вхідні і вихідні дані дійсними числами типу **float**, написати програму для розв'язання задачі згідно варіанту.

Варіанти завдань

1. Дано два кола із загальним центром і радіусами R_1 і R_2 ($R_1 > R_2$). Знайти площі цих кіл S_1 та S_2 , а також площу S_3 кільця, зовнішній радіус якого дорівнює R_1 , а внутрішній радіус дорівнює R_2 . Як значення π використовувати 3.14.
2. Дано координати двох протилежних вершин прямокутника: (x_1, y_1) , (x_2, y_2) . Сторони прямокутника паралельні осям координат. Знайти периметр і площу даного прямокутника.
3. Дано сторони прямокутника a і b . Знайти його площу S і периметр P .
4. Дано довжини ребер A , B , C прямокутного паралелепіпеда. Знайти його об'єм та площу поверхні.
5. Дано число A . Обчислити A^{15} , використовуючи дві допоміжні змінні і п'ять операцій множення. Для цього послідовно знаходити A^2 , A^3 , A^5 , A^{10} , A^{15} . Вивести всі знайдені степені числа A .
6. Дано значення температури T в градусах Фаренгейта. Визначити значення цієї ж температури в градусах Цельсія. Температура по Цельсію T_C і температура за T_F Фаренгейтом T_F зв'язані наступними співвідношенням:
$$T_C = (T_F - 32) \cdot 5/9.$$
7. Дано значення температури T в градусах Цельсія. Визначити значення цієї ж температури в градусах Фаренгейта. Температура за Цельсієм T_C і

температура за Фаренгейтом T_F пов'язані наступним співвідношенням: $T_C = (T_F - 32) \cdot 5/9$.

8. Відомо, що X кг шоколадних цукерок коштує A гривень, а Y кг ірисок - B гривень. Визначити, скільки коштує 1 кг шоколадних цукерок, 1 кг ірисок, а також у скільки разів шоколадні цукерки дорожчі ірисок.

9. Дано два ненульових числа. Знайти суму, різницю, добуток і частку їх квадратів.

10. Швидкість першого автомобіля V_1 км / год, другого - V_2 км / год ($V_1 > V_2$), відстань між ними S км. Визначити відстань між ними через T годин, якщо рухаються в одному напрямку.

11. Швидкість першого автомобіля V_1 км / год, другого - V_2 км / год, відстань між ними S км. Визначити відстань між ними через T годин, якщо автомобілі рухаються назустріч один одному.

12. Пішохід пройшов S_1 км за t_1 годин і S_2 км за t_2 годин. Вивести загальний час руху пішохода та його середню швидкість.

13. Змішали m_1 грамів розчину, що містить p_1 % солі з m_2 грамами розчину, що містить p_2 % солі. Вивести загальну масу розчину та відсотковий вміст солі в ньому.

14. Човен проходить відстань S за течією річки і назад. Швидкість човна V км/год, швидкість течії U км/год. Знайти час витрачений човном на дорогу та його середню швидкість.

15. Перша труба наповнює резервуар за t_1 хвилин, друга – за t_2 хвилин. За скільки хвилин обидві труби наповнять резервуар. Яка частина резервуара при цьому буде заповнена першою трубою.

16. В одному магазині придбали m_1 кг деякого товару за ціною p_1 грн за кілограм, в іншому – ще m_2 кг цього ж товару але за ціною p_2 грн за кілограм. Знайти загальну вартість покупки. Якою вийшла середня ціна товару.

Зразок виконання.

Користуючись поясненнями в теоретичних відомостях запускаємо IDE Eclipse, створюємо новий проект, в ньому клас з методом main, де записуємо відповідний код програми. Орієнтовний код класу:

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Lab1 {

    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader(
            new InputStreamReader(System.in));
        System.out.println("Ціна товару в 1-му магазині:");
        float p1 = Float.parseFloat(reader.readLine());
        System.out.println("Маса придбаного товару:");
        float m1 = Float.parseFloat(reader.readLine());
        System.out.println("Ціна товару в 2-му магазині:");
        float p2 = Float.parseFloat(reader.readLine());
        System.out.println("Маса придбаного товару:");
        float m2 = Float.parseFloat(reader.readLine());
        float s = p1*m1 + p2*m2;
        float p = s/(m1+m2);
        System.out.println("Вартість придбаного товару " +
            s + "грн.");
        System.out.println("Середня ціна за кілограм - " +
            p + "грн.");
    }
}
```

Запитання для самоконтролю

1. Що таке IDE?
2. З чого складається Java-програма?
3. Яка синтаксична конструкція Java використовується для консольного виводу тексту?
4. Які класи Java можна використати для вводу текстових даних?
5. Методи яких класів Java можна використати для вводу числових даних?

Використані джерела:

1. Ткаченко О.М. Комп'ютерне програмування на мові Java. – К.: "Аграр Медіа Груп", 2013. – 147 с.
2. Java. Методы программирования : уч.-мет. пособие / И.Н.Блинов, В.С.Романчик. - Минск: издательство "Четыре четверти", 2013.- 896 с.
3. Васильев А.Н. Самоучитель Java с примерами и программами. СПб.: Наука и Техника, 2011. - 352 с.
4. Дэвид Карлсон. Eclipse. - Издательство "Лори", 2013. - 353 с.
5. https://uk.wikibooks.org/wiki/Освоюємо_Java
6. http://wiki.eclipse.org/The_Official_Eclipse_FAQs
7. <https://docs.oracle.com/javase/tutorial/>

НАВЧАЛЬНЕ ВИДАННЯ

Знайомство з IDE Eclipse. Створення та виконання простої Java-програми. Програмування алгоритмів лінійної структури. Консольний ввід числових даних та виконання простих арифметичних обчислень.

МЕТОДИЧНІ ВКАЗІВКИ до виконання лабораторних робіт № 1-2 з дисципліни «Мови програмування»

для студентів базового напрямку 6.040204, «Прикладна фізика».

Укладачі:

Гнатів Л.Б., к. ф-м. н, доцент,
Ментинський С.М., ст. викл,
Угрин С.З., асистент.

Комп'ютерне складання: Ментинський С.М., ст. викл.