

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ

до самостійної роботи студентів та
виконання контрольної роботи з дисципліни

“Алгоритмізація та програмування”

для студентів I року заочної форми навчання

Спеціальність – 122 «Комп’ютерні науки»

Одеса 2016

Методичні вказівки до самостійної роботи студентів та виконання контрольної роботи з дисципліни “Алгоритмізація та програмування” для студентів I року заочної форми навчання. Спеціальність – 122 «Комп’ютерні науки» / Укладачі: Коваленко Л.Б., к.г.н., доц., Кузніченко С.Д., к.г.н., доц. – Одеса, ОДЕКУ, 2016. – 50 с.

1 Загальна частина

1.1 Передмова

Методичні вказівки призначені для студентів I курсу заочної форми навчання. Мета виконання контрольної роботи – закріплення теоретичного лекційного матеріалу та придбання практичних навичок програмування мовою Java, яка є сучасною та найбільш розвиненою у цей час.

Дисципліна «Алгоритмізація та програмування» є нормативною дисципліною у напрямі бакалаврської підготовки «Комп'ютерні науки».

Методичні вказівки містять рекомендації по вивченню розділів дисципліни, контрольні запитання та завдання. Контрольна робота підкріплена прикладами розв'язання типових задач на ПЕОМ.

Загальний обсяг навчального процесу в годинах, рівень знань та вмінь при вивченні дисципліни визначаються освітньо-професійними програмами.

Дисципліна " Алгоритмізація та програмування " базується на базових знаннях, отриманих в шкільних та вузівських загальноосвітніх курсах основ інформатики та математики.

Внаслідок вивчення дисципліни " Алгоритмізація та програмування " студенти **повинні знати** основні поняття інформатики і програмування (типові алгоритмічні конструкції, етапи обробки програм на ПЕОМ, примитивні типи даних та оператори мови Java). Вони **повинні вміти** застосовувати алгоритми подання, зберігання й обробки інформації, складати лінійні та розгалужені програми мовою Java та здобути практичні навички створення і модифікації невеликих програмних проектів.

Отримані студентами знання та вміння використовуються при виконанні розрахунково-графічних робіт, курсовому та дипломному проектуванні, у низці магістерських дисциплін.

Методичні вказівки призначені допомогти студентам заочної форми навчання опанувати основні положення мови Java, а також ознайомити їх з практичними аспектами розробки програм.

У методичних вказівках розглядаються питання, які відповідають навчальній програмі дисципліни.

1.2 Зміст дисципліни

Тема 1. Основи процедурного програмування

Поняття алгоритму й основні алгоритмічні структури.

Введення в мову програмування Java. Виконання Java-програми.

Коментар. Ідентифікатори.

Тема 2. Примітивні типи даних і операції

Прості типи даних: логічний, цілі, дійсні, символи та рядки.

Оголошення змінних. Арифметичні і логічні операції. Операції порівняння. Приведення типів. Операції присвоювання.

Методи класу Math пакета java.lang.

Побітові операції. Умовна операція. Пріоритет операцій. Вираження.

Тема 3. Оператори мови Java

Організація розгалужень у програмі. Умовний оператор if.

Оператор варіанту Switch.

Мітки. Організація переривань за допомогою оператора break.

Оператор присвоювання.

Цикли. Циклічні конструкції while і do- while.

Цикл із параметром for. Ітераційні алгоритми. Оператор кома.

Оператор continue і return.

Тема 4. Посилальні типи даних. Масиви.

Масиви. Опис масивів. Операції над масивами. Методи сортувань елементів масивів.

Алгоритми обробки символьних масивів.

Багатомірні масиви. Приклади рішення завдань із використанням багатомірних масивів.

При вивченні даного курсу використовується наступна **навчальна та методична література**:

1.3 Література

Основна

1. *Верлань А.Ф., Чмырь І.А., Кузніченко С.Д., Коваленко Л.Б.* Императивное программирование и объектно-ориентированное моделирование: Java, UML, OCL : Учебное пособие для студентов высших учебных заведений. Одесса, Издательство Экология. – 2014., 326 с.

2. *Кузніченко С.Д., Коваленко Л.Б.* Алгоритмізація та програмування: Конспект лекцій – Одеса: (ел.вар), 2015. – 340 с.

Додаткова

1. *Ковалюк Т.В.* Алгоритмізація та програмування: Підручник – Львів: Видавництво «Магнолія 2006», 2013. – 400 с.
2. *Эккель Б.* Философия Java.: ВНУ., СПб – 2001. 850с.
3. *Морган М.* Java 2. Руководство разработчика. – М.: Изд. дом «Вильямс», 2000.
4. *Нотон П., Шилдт Г.* Полный справочник по Java. – Киев: Диалектика, 1997.
5. *Флэнэген Д.* Java in a Nutshell. Полное руководство. – Киев: ВНУ, 1998.
6. *Роганов Е.А.* Основы информатики и программирования: Учебное пособие – М.: МГИУ, 2001. – 315 с.
7. *Java 2.* Библиотека профессионала, том 1. Основы. 7-е издание / Кей С. Хорстманн, Гари Корнелл. – Издательский дом «Вильямс», 2006, 896 с.: ил.
8. *Быстрая разработка программ на Java и C++:* принципы, примеры, практика Роберт С. Мартин - Издательский дом «Вильямс», 2003, 752 с.: ил.
9. *Ноутон О., Шилдт Г.,* Java 2 в подлиннике.: ВНУ., СПб – 2000., 1050с.
10. *Шень А.* Программирование: теоремы и задачи. – М.: МЦНМО, 1995.
11. *Кормен Т., Лейзерсон Ч., Ривест Р.* Алгоритмы: построение и анализ. – М.: МЦНМО, 2000.
12. *Вирт Н.* Алгоритмы+структуры данных=программы. – М.: Мир, 1985.
13. *Ахо А., Хопкрофт Д., Ульман Д.* Структуры данных и алгоритмы. – М.: Изд. дом «Вильямс», 2000.

1.4 Перелік знань та вмінь

Після освоєння цієї дисципліни студент **повинен знати**: основні типові алгоритмічні конструкції, етапи обробки програм на ПЕОМ (редагування, трансляція, компонування), основні оператори мови Java, типи даних. Крім цього, необхідно знати рекурсивні алгоритми та алгоритми сортування масивів.

За результатами навчання студент **повинен вміти**: використовувати основні поняття алгоритмічних структур для побудови алгоритмів рішення задач за допомогою ЕОМ, розробляти алгоритми методом покрокового уточнення, складати лінійні та розгалужені програми мовою Java, працювати з різними типами даних, у тому числі масивами і класами, відлагоджувати та виконувати Java-програму.

2. ОРГАНІЗАЦІЯ САМОСТІЙНОЇ РОБОТИ СТУДЕНТА

2.1 Рекомендації студенту по роботі над курсом "Основи програмування та алгоритмічні мови"

Основною формою навчання студента є самостійна робота (СРС) над навчальним матеріалом. СРС повинна сприяти активізації творчого мислення студентів, формуванню високої культури розумової праці, підвищенню самостійності студентів та індивідуалізації процесу навчання. СРС складається з наступних елементів: вивчення матеріалу по підручниках, конспектах лекцій, виконання лабораторних та контрольних робіт. Студент може звертатися до викладача з питаннями для одержання письмової чи усної консультації. Однак студент повинен пам'ятати, що тільки при систематичній і завнятій самостійній роботі допомога викладача виявиться досить ефективною.

2.1.1 Загальні поради:

– зміст кожної теми курсу вивчається за допомогою наведеного у розділі I переліку навчальної та методичної літератури, в першу чергу використовується основна та методична література.

– після засвоєння змісту кожної теми курсу треба відповісти на „Запитання для самоперевірки”, які наведені у цих методичних вказівках.

– завдання з контрольної роботи виконувати згідно з наведених вимог (п.2.2)

– якщо виникли питання при вивченні теоретичного матеріалу або при виконанні контрольної роботи, можна звернутися до викладача, який читав установчі лекції, письмово на адресу університету звичайною або електронною поштою: kaf-inform@odeku.edu.ua або dean-comp@odeku.edu.ua

2.1.2 Повчання по послідовному вивченню теоретичного матеріалу

Java є суворо типізованою мовою. Це означає, що будь-яка змінна і будь-який вираз мають відомий тип ще на момент компіляції. Таке суворе правило дозволяє виявляти багато помилок вже під час компіляції. Компілятор, знайшовши помилку, вказує точне місце (рядок) і причину її виникнення, а динамічні "баги" (від англійського bugs) необхідно спочатку виявити за допомогою тестування (що може зажадати значних зусиль), а потім знайти місце в коді, яке їх породило. Тому чітке розуміння моделі типів даних в Java дуже допомагає в написанні якісних програм.

Всі типи даних поділяються на дві групи. Першу складають 8 простих, або примітивних (від англійського primitive), типів даних. Вони поділяються на три підгрупи:

цілочисельні

- byte
- short
- int
- long
- char (також є цілочисловим типом)

дробові

- float
- double

булеві

- boolean

Другу групу складають об'єктні, або вказівні (від англійського reference), типи даних. Це всі класи, інтерфейси та масиви. У стандартних бібліотеках перших версій Java перебувало кілька сот класів і інтерфейсів, зараз їх вже тисячі. Крім стандартних, написано багато класів та інтерфейсів, складаючи будь-яку Java-програму.

Цілочисельні типи

Цілочисельні типи - це *byte*, *short*, *int*, *long*, також до них відносять і *char*. Перші чотири типи мають довжину 1, 2, 4 і 8 байт відповідно, довжина *char* - 2 байти, це безпосередньо впливає з того, що всі символи Java описуються стандартом Unicode. Довжини типів наведені тільки для оцінки областей значення. Як вже говорилося, пам'ять в Java представляється віртуальною і обчислити, скільки фізичних ресурсів займе та чи інша змінна, так прямолінійно не вийде.

4 основних типи є знаковими. *char* доданий до цілочисловим типам даних, так як з точки зору JVM символ і його код - поняття взаємооднозначної. Звичайно, код символу завжди позитивний, тому *char* - єдиний беззнаковий тип. Ініціювати його можна як символічним, так і цілочисловим літералом. У всьому іншому *char* - повноцінний числовий тип даних, який може брати участь, наприклад, в арифметичних діях, операціях порівняння і т.п. У таблиці 4.1 зведені дані по всім розібраним типами:

Таблиця 4.1

Назва типу	Довжина (байти)	Область значень
Byte	1	-128 .. 127
Short	2	-32.768 .. 32.767
Int	4	-2.147.483.648 .. 2.147.483.647
Long	8	-9.223.372.036.854.775.808 .. 9.223.372.036.854.775.807 (приблизно 10^{19})
Char	2	'\u0000' .. '\uffff', или 0 .. 65.535

Над цілочисельними аргументами можна проводити наступні операції:

- операції порівняння (повертають булево значення)
 - $<$, $<=$, $>$, $>=$
 - $==$, $!=$
- числові операції (повертають числове значення)
 - унарні операції $+$ і $-$
 - арифметичні операції $+$, $-$, $*$, $/$, $\%$

- операції інкремента і декремента (в префіксній і постфіксній формі): ++ і --
- операції бітового зсуву <<, >>, >>>
- бітові операції ~, &, |, ^
- оператор з умовою? :
- оператор приведення типів
- оператор конкатенації з рядком +

Оператори порівняння цілком очевидні і окремо ми їх розглядати не будемо. Їх результат завжди булевого типу (*true* або *false*).

Дробові типи

Дробові типи - це *float* і *double*. Їх довжина - 4 і 8 байт, відповідно. Обидва типи знакові. Нижче в таблиці зведені їх характеристики:

Таблиця 4.2. Дробові типи даних.

Назва типу	Довжина (байт)	Область значень
Float	4	3.40282347e+38f ; 1.40239846e-45f
Double	8	1.79769313486231570e+308 ; 4.94065645841246544e-324

Для цілочисельних типів область значень задавалася верхньою і нижньою межами, вельми близькими по модулю. Для дробових типів додається ще одне обмеження - наскільки можна наблизитися до нуля, іншими словами - яке найменше позитивне ненульове значення. Таким чином, не можна задати літерал свідомо більший, ніж дозволяє відповідний тип даних, це призведе до помилки *overflow*. І не можна задати літерал, значення якого за модулем занадто мало для такого типу, компілятор згенерує помилку *underflow*.

Нагадаємо, що коли наприкінці літерала стоїть буква *F* або *f*, то літерал розглядається як значення типу *float*. За замовчуванням дробовий літерал має тип *double*, при бажанні це можна підкреслити буквою *D* або *d*.

Над дробовими аргументами можна проводити наступні операції:

- операції порівняння (повертають булево значення)
 - <, <=, >, > =

- ==, !=
- числові операції (повертають числове значення)
 - унарні операції + і -
 - арифметичні операції +, -, *, /,%
 - операції інкремента і декремента (в префіксній і постфіксній формі): ++ і --
- оператор з умовою « ? : »
- оператор приведення типів
- оператор конкатенації з рядком +

Практично всі оператори діють за тими ж принципами, що передбачені для цілочисельних операторів (оператор ділення із залишком % розглядалося в попередній лекції, а оператори ++ і -- також збільшують або зменшують значення змінної на одиницю). Уточнимо лише, що оператори порівняння коректно працюють і у разі порівняння цілочисельних значень з дробовими. Таким чином, в основному необхідно розглянути питання переповнення і перетворення типів при обчисленнях.

Логічний тип

Логічний тип представлений лише одним типом boolean, який може зберігати всього два можливі значення - true і false. Величини саме цього типу виходять в результаті операцій порівняння.

Над булевими аргументами можна проводити наступні операції:

- операції порівняння (повертають булево значення):
 - ==, !=
- логічні операції (повертають булево значення):
 - !
 - &, |, ^
 - &&, ||
- оператор з умовою « ? : »
- оператор конкатенації з рядком +

У оператора з умовою « ? : » Першим аргументом може бути тільки значення типу boolean. Також допускається, щоб другий і третій аргументи одночасно мали логічний тип.

Операція конкатенації з рядком перетворює булеву величину в текст "true" або "false" залежно від значення.

Тільки булеві вирази допускаються для управління потоком обчислень, наприклад, в якості критерію умовного переходу `if`.

Ніяке число не може бути інтерпретовано як логічне вираження. Якщо передбачається, що ненульове значення еквівалентно істині (за правилами мови C), то необхідно записати `x != 0`. Вказівні величини можна перетворювати в `boolean` виразом `ref! = Null`.

Оператор `if/else`

Оператор `if/else` використовується для ухвалення рішення, тобто він вибирає необхідну послідовність дій. Синтаксична структура:

```
if(boolean expression)
{
    // do this if the expression is true
}
else
{
    // do this if the expression is false
}
}
```

Гілка `else` в операторі `if/else` є необов'язковим. Фігурні дужки необов'язкові, якщо основна частина складається з одного вираження. Наприклад:

```
if (x == 5) {}           // compiles, executes body if x
                          // is equal to 5
  if (x = 0) {}         // does not compile, because
                          // expression is non-boolean

  if (x = true) {}     // compiles, but the body is
                          // always executed
```

У випадку вкладених операторів `if/else`, кожне ключове слово `else` ставиться до найближчого попереднього йому ключовому слову `if`, що не має `else`.

Оператор `switch`

Оператор `switch` використовується для прийняття рішення, ґрунтуючись на цілочисельному вираженні. Аргумент, переданий

операторам `switch` і `case` повинен мати тип `int`, `short`, `char`, або `byte`. Аргумент для оператора `case` повинен бути літералом або `final` (остаточною) змінною. Якщо збіг не знайдений, виконується гілка `default`, що є необов'язковою.

Коли виявлений оператор `break`, контроль передається за межі оператора `switch`. Якщо оператор `break` не зазначений, всі оператори `case` виконуються доти, поки не виявиться `break`, або оператор `switch` закінчиться. Наприклад:

```
int x = 1;
switch(x)
{
    case 1:      System.out.println("one");
    case 2:      System.out.println("two");
}

// both one and two are printed
```

Зверніть увагу на положення оператора `default`. Хоча оператор `default` звичайно перебуває наприкінці всіх опцій вибору, це не обов'язково, як видно з нижченаведеного приклада:

```
int i = 1;
switch(i)
{
    default:
        System.out.println("Default");
        break;
    case 0:
        System.out.println("Zero");
        break;
    case 1:
        System.out.println("One");
        break;
}
```

Оператори циклу

Існує три типи конструкцій мови Java для створення циклів: `while`, `do-while` і `for`.

Цикл `while`

Синтаксична структура циклу `while`:

```
while(boolean expression) {}
```

Тіло циклу `while` виконується, тільки якщо вираження істинно:

```
while(false) {} // body never executed
    while(1) {} // code will not compile, not a
boolean
```

Циклу `do-while`

Синтаксична структура циклу `do-while`:

```
do { } while(boolean expression);
```

Тіло циклу `do-while` виконується, принаймні, один раз, тому що тестове вираження оцінюється тільки після виконання тіла циклу. Не забудьте поставити крапку з коми після вираження `while`.

Цикл `for`

Синтаксична структура циклу `for`:

```
for(expr1; expr2; expr3)
{
    // body    }
```

`expr1` для ініціалізації, `expr2` перевірка умови, і `expr3` вираження для ітерації. Кожна із цих трьох секцій може бути опущена й синтаксична структура залишиться достовірною:

```
for( ; ; ) {} // an endless loop
```

Може бути більше одного вираження для ініціалізації й більше одного вираження для ітерації, але тільки одне тестове вираження.

Оператори break i continue

Оператор `break` використовується для виходу з операторів `loop` або `switch`, оператор `continue` використовується для того, щоб пропустити поточну ітерацію й перейти до наступній.

У випадку вкладених циклів оператор `break` виходить тільки із самого внутрішнього циклу. Якщо ключове слово `break` використовується разом з міткою, оператор `break` вийде з позначеного циклу:

```
outer:    for (int i = 0; i < 10; i++)
    {
        while(y > 0)
        {
            break outer;
        }
    }

// breaks from the for loop if y > 0
```

2.2 Перелік завдань до контрольної роботи

2.2.1 Загальні поради до виконання контрольної роботи

1. За допомогою навчальної та методичної літератури, список якої наведено у попередній частині Методичних вказівок та повчань, які сформульовано у п.2.1 Методичних вказівок (дивись вище), необхідно вивчити зміст теоретичної частини тем курсу. Самоперевірка засвоєння знань здійснюється за допомогою „Контрольних питань”, які наводяться у кожному із завдань.

2. Після засвоєння теоретичного матеріалу необхідно виконати контрольну роботу 1, яка містить в собі завдання по всіх темах курсу.

Виконання контрольної роботи

Номер варіанта контрольних завдань визначається останньою цифрою номера залікової книжки.

Звіт по контрольному завданню повинен містити:

- номер і варіант контрольного завдання
- відповіді на контрольні запитання
- програму на мові JAVA
- результати розрахунку і друк (при наявності засобів обчислювальної техніки)

КОНТРОЛЬНА РОБОТА № 1

Завдання № 1

Тема: «Примітивні типи даних. Арифметичні і логічні операції»

Мета роботи

Метою завдання є ознайомлення студентів з прийомами програмування найпростіших програм мовою Java з використанням примітивних типів даних та арифметичних і логічних операцій.

Завдання до роботи

Використовуючи алгоритми, розглянуті на лекційних заняттях, скласти програму розрахунку заданих величин.

Методичні вказівки

■ Завдання спирається на знання й уміння, отримані при вивченні наступних тем лекційного курсу:

- Виконання Java-програми. Коментар. Ідентифікатори.
- Прості типи даних: логічний, цілі, дійсні, символи та рядки.
- Оголошення змінних.
- Арифметичні і логічні операції.
- Операції порівняння.
- Правила приведення типів.
- Операції присвоювання.

Тому під час підготовки до виконання завдання рекомендується повторити зазначені розділи дисципліни.

■ Перед виконанням завдання необхідно уважно ознайомитися з особливостями виконання Java - програми.

Написати програму на Java можна в будь-якому текстовому редакторі, наприклад, Notepad, WordPad в MS Windows і ін. Запустити компілятор, а потім інтерпретатор байт-кодів можна з командного рядка. Однак, зручніше для цих цілей використовувати одну із численних інтегрованих середовищ розробки програм для Java. Найпростіші з них є просто інтегрованими оболонками над JDK, що викликають із одного вікна текстовий редактор, компілятор і інтерпретатор. Ці інтегровані середовища

вимагають попередньої установки JDK.

Для виконання завдань студенти можуть використовувати середовище розробки програм BlueJ, яке спеціально створене для навчання на початковому рівні. У додатку наведений докладний опис етапів виконання Java-програми та особливості роботи із середовищем розробки BlueJ.

■ В кожному варіанті завдання є два невеликих завдання, тому студент за бажанням може створити у власному проєкті два окремих класа, або передбачити виконання усіх завдань в одному класі проєкту.

■ При рішенні завдань можуть бути корисними наступні математичні формули з планіметрії:

- площа круга $S = \pi R^2$;
- довжина кола $l = 2\pi R$;
- об'єм конуса $V = \frac{\pi R^2 h}{3}$;
- об'єм прямого кругового циліндра $V = \pi R^2 h$;
- площа бічної поверхні циліндра $S = 2\pi R h$;
- об'єм кулі $V = \frac{4}{3} \pi R^3$;
- площа поверхні кулі $S = 4\pi R^2$.

Приклад програми

Знайти площу поверхні та об'єм циліндра висотою h та радіусом R . Визначити, чи є цифра 5 серед цифр, що складають десятинний запис додатного тризначного цілого числа M .

```
class lab_1{
public static void main (String[] args){
//-----I-----
double h=5.8, R=3.2,PI=3.1457;
double S=2*PI*R*(R+h);
double V=PI*R*R*h;
System.out.println("Площа циліндра = "+S+
"\n"+"Об'єм циліндра = "+V);

//-----II-----
int M=375;
int k1=M/100; //кількість сотень
int k2=M/10%10;//кількість десятків
int k3=M%10;//кількість одиниць
boolean log=(k1==5) || (k2==5) || (k3==5);
```

```
System.out.println("Серед цифр числа є цифра 5 - "+log);}}
```

Контрольні питання

1. Які ідентифікатори припустимі в Java? Чим відрізняються прості й складені ідентифікатори?
2. Які види коментарів існують в Java?
3. Назвіть особливості виконання Java - програми. Яким чином можна створити вихідний, об'єктний та завантажувальний модуль?
4. Які примітивні типи даних існують в Java? Приведіть діапазони припустимих значень і розрядність для цілих і дійсних типів даних.
5. Який формат має оголошення змінної? Приведіть приклади оголошень змінних примітивних типів.
6. Пояснить дію операції зчеплення.
7. За яким правилом здійснюється приведення типів в Java?
8. Пояснить дію постфіксної і префіксної форми запису операцій інкрементування та декрементування.
9. Пояснить дію простої та складової операцій присвоювання.
10. Які особливості виконання арифметичних операцій ділення та знаходження остачі від ділення стосовно до змінних цілого типу?

Варіанти завдань

1. * Цілій змінній h присвоїти третю з кінця цифру в записі додатного цілого числа k .
* Визначити, чи може окружність радіуса R пройти в прямокутний отвір зі сторонами a, b .
2. * Цілій змінній d присвоїти першу цифру дрібної частини додатного дійсного числа x .
* Визначити, чи є рівнобедреним трикутник зі сторонами a, b, c .
3. * Визначити, чи дорівнює сума двох перших цифр заданого чотиризначного числа сумі двох останніх.
* Для k -ої секунди доби визначити, скільки повних годин (h) і повних хвилин (m) пройшло до цього моменту.

4. * Визначити, чи є цифра 0 серед перших трьох цифр дрібної частини заданого дійсного числа.
* Визначите кут (у градусах) між положенням годинної стрілки на початку доби і її положенням в h годин, m хвилин, s секунд ($0 \leq h \leq 11$, $0 \leq (m,s) \leq 59$).
5. * Визначите, чи є серед цифр заданого тризначного числа однакові.
* Визначите повну кількість годин і хвилин, які пройшли від початку доби до того моменту (у першій половині дня), коли годинна стрілка повернулася на кут φ градусів ($0 \leq \varphi < 360$).
6. * Визначити суму перших трьох цифр дрібної частини додатного дійсного числа x .
* Дано три довільних числа. Визначити, чи можна побудувати трикутник з такими довжинами сторін.
7. * Дано два додатних цілих тризначних числа. Визначите, чи дорівнюють добутки їхніх цифр.
* Поміняти місцями значення цілих змінних x і y , не використовуючи додаткових змінних.
8. * Визначите, чи більше перша цифра додатного цілого тризначного числа за середнє арифметичне всіх його цифр.
* Знайти, скільки цілих діб і годин пройшло за m хвилин.
9. * Надрукувати число, яке виходить при перестановці цифр сотень і одиниць заданого тризначного числа.
* Для довільних дійсних чисел a , b , c визначити, чи має квадратне рівняння хоча б один дійсний корінь.
10. * Визначите, чи є перша цифра додатного цілого чотиризначного числа більше за добуток його інших цифр.
* Обчислити довжину кола, площу круга та об'єм сфери одного й того ж радіусу R .

Завдання № 2

Тема: «Робота з пакетом java.lang. Методи класу Math»

Мета роботи

Отримання навичок програмування алгоритмів лінійної структури та алгоритмічних виразів з використанням методів класу Math пакета java.lang. Вивчення роботи класу Xterm, який містить методи для введення даних примітивних типів з клавіатури.

Завдання до роботи

Напишіть програму розрахунку заданих арифметичних виразів. Для введення даних із клавіатури необхідно використовувати клас Xterm. У результаті передбачити виведення 5-ти цифр після коми.

Методичні вказівки

Методичні вказівки

■ Лабораторна робота спирається на знання й уміння, отримані при вивченні наступних тем лекційного курсу:

- Прості типи даних: логічний, цілі, дійсні, символи та рядки.
- Оголошення змінних.
- Правила приведення типів.
- Операції присвоювання.
- Методи класу Math пакета java.lang.

Тому під час підготовки до лабораторної роботи рекомендується повторити зазначені розділи дисципліни.

■ Для використання математичних методів класу Math, необхідно імпортувати в програму пакет java.lang: `import java.lang.Math.`

■ Для введення даних із клавіатури треба використовувати клас Scanner, для цього необхідно імпортувати в програму пакет `import java.util.*;`

■ Для виводу розрахункових значень на консоль треба використовувати метод `System.out.printf()`.

При форматованому виведенні використовуються специфікатори формату, деякі з них наведенні у таблиці¹ нижче:

Специфікатор формату	Тип елементу даних
%b	Логічне значення аргументу (boolean)
%c	Одиночний символ
%d	Ціле число типу int
%e	Число з плаваючою точкою
%f	Число з фіксованою точкою типу float
%g	Обирає більш коротке представлення з двох: %e або %f
%s	Рядок символів
%n	Вставка символу нового рядку
%%	Вставка знака %

Також можливе використання специфікаторів з великими літерами: %S (еквівалентно %s). Форматування з їх допомогою забезпечує перевід символів в верхній регістр.

Приклади застосування форматного виводу:

```
System.out.printf("%b %B", true, false);  
true FALSE  
System.out.printf("%-3C, %3c", 'd', 'e');  
D , e  
System.out.printf("%-6dx",5);  
5 x  
System.out.printf("%1.5f", 123.456f);  
123.45600  
System.out.printf("%6.0f", 123.456f);  
123  
System.out.printf("%1.5E", 123.456f);  
1.23456E+02  
System.out.printf("This %s is about %n%S %c", "book","java",'6');  
This book is about  
JAVA 6
```

¹ Повний список специфікаторів формату можете переглянути у документації мови Java. <http://docs.oracle.com/javase/1.5.0/docs/api/java/util/Formatter.html> або на ресурсі <http://crypto.pp.ua/2010/06/formatirovanie-strok-java/>

■ В кожному варіанті лабораторної роботи є три завдання. Друге завдання містить два математичних вирази. Якщо студент правильно склав алгоритм розв'язання задачі, то повинен отримати два однакових результати розрахунку. В третьому завданні необхідно згенерувати випадкові числа у заданому інтервалі з використанням методу `random()`.

Приклад програми

Розрахуйте $y = (-1)^n (2n + 1) / 2$, де n - довільне ціле число, що вводитьься з клавіатури;

Розрахуйте

$$z_1 = 2 \sin^2(3\pi - 2\alpha) \cos^2(5\pi + 2\alpha)$$

$$z_2 = \frac{1}{4} - \frac{1}{4} \sin\left(\frac{5}{2}\pi - 8\alpha\right)$$

```
import java.lang.Math;
import java.util.*;
class raset{
public static void main (String[] args) {
Scanner scan = new Scanner (System.in);
System.out.println("Введіть n->");
int n=scan.nextInt();
double y=Math.pow(-1,n)*(2*n+1)/2.;
System.out.println("Введіть a->");
double a=scan.nextDouble();
double z1,z2;
//double p=10e5;
z1=2*Math.pow(Math.sin(3*Math.PI-
2*a)*Math.cos(5*Math.PI+2*a),2);
z2=1./4.-1./4.*Math.sin(5./2.*Math.PI-8*a);
//System.out.println("Y= "+Math rint(y*p)/p);
//System.out.println("Z1= "+Math rint(z1*p)/p);
//System.out.println("Z2= "+Math rint(z2*p)/p);
System.out.printf("Y= %6.3f",y);
System.out.printf("      Z1=%6.2f",z1);
System.out.printf("      Z2= %6.2f",z2);

}}
```

2.3 Контрольні питання

1. За допомогою якого виразу можна імпортувати пакет у програмний модуль? Приведіть приклади виразів, що імпортують пакет `java.lang` повністю або тільки його клас `Math`.

2. Яким чином здійснити ввід даних за допомогою методів класу **Scanner**. Наведіть приклади вводу змінних цілого та дійсного типів.

3. Для чого призначений метод `random()` класу `Math`? Як з його допомогою можна одержати випадкове число із діапазону $[-15 \div 35]$?

4. Яка відмінність між методами `rint()` і `round()` класу `Math`? Якого типу буде результат виконання методу `round()`, якщо аргумент має тип `double`? `float`?

5. Опишіть призначення методів `ceil()` і `floor()`. Приведіть приклади їхнього використання.

6. Яка різниця між методами `print()` і `println()`?

Варіанти завдань

1. * Розрахуйте $y = n(n + 1) / 2$, где n - довільне ціле число, що вводиться з клавіатури;

* Розрахуйте

$$z_1 = \cos \alpha + \sin \alpha + \cos 3\alpha + \sin 3\alpha$$

$$z_2 = 2\sqrt{2} \cos \alpha \cdot \sin \left(\frac{\pi}{4} + 2\alpha \right)$$

* Виконати завдання 3 згідно варіанту за таблицею 1.

2. * Розрахуйте $y = (2n + n)(n + 1) / 2$, где n - довільне ціле число, що вводиться з клавіатури;

* Розрахуйте

$$z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha + 1 - 2\sin^2 2\alpha}$$

$$z_2 = 2\sin \alpha$$

* Виконати завдання 3 згідно варіанту за таблицею 1.

3. * Розрахуйте $y = n(n + 1)(2n + 1)/6$, где n - довільне ціле число, що вводиться з клавіатури;

* Розрахуйте

$$z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos\alpha - \cos 3\alpha + \cos 5\alpha}$$

$$z_2 = \operatorname{tg} 3\alpha$$

* Виконати завдання 3 згідно варіанту за таблицею 1.

4. * Розрахуйте $y = n^2(n + 1)^2 / 4$, где n - довільне ціле число, що вводиться з клавіатури;

* Розрахуйте

$$z_1 = 1 - \frac{1}{4} \sin^2 2\alpha + \cos 2\alpha$$

$$z_2 = \cos^2 \alpha + \cos^4 \alpha$$

* Виконати завдання 3 згідно варіанту за таблицею 1.

5. * Розрахуйте $y = n(4n^2 - 1)/3$, где n - довільне ціле число, що вводиться з клавіатури;

* Розрахуйте

$$z_1 = \cos \alpha + \cos 2\alpha + \cos 6\alpha + \cos 7\alpha$$

$$z_2 = 4 \cos \frac{\alpha}{2} \cdot \cos \frac{5}{2} \alpha \cdot \cos 4\alpha$$

* Виконати завдання 3 згідно варіанту за таблицею 1.

6. * Розрахуйте $y = n^2(2n^2 - 1)/5$, где n - довільне ціле число, що вводиться з клавіатури;

* Розрахуйте

$$z_1 = \cos^2 \left(\frac{3}{8} \pi - \frac{\alpha}{4} \right) - \cos^2 \left(\frac{11}{8} \pi + \frac{\alpha}{4} \right)$$

$$z_2 = \frac{\sqrt{2}}{2} \sin \frac{\alpha}{2}$$

* Виконати завдання 3 згідно варіанту за таблицею 1.

7. * Розрахуйте $y = n(n+1)(2n+1)(3n^2-3n-1)/30$, где n - довільне ціле число, що вводиться з клавіатури;

* Розрахуйте

$$z_1 = \cos^4 x + \sin^2 y + \frac{1}{4} \sin^2 2x - 1$$

$$z_2 = \sin(y+x) \cdot \sin(y-x)$$

* Виконати завдання 3 згідно варіанту за таблицею 1.

8. * Розрахуйте $y = n^3(n-1)(2n+1)/30$, где n - довільне ціле число, що вводиться з клавіатури;

* Розрахуйте

$$z_1 = (\cos \alpha - \cos \beta)^2 - (\sin \alpha - \sin \beta)^2$$

$$z_2 = -4 \sin^2 \frac{\alpha - \beta}{2} \cdot \cos(\alpha + \beta)$$

* Виконати завдання 3 згідно варіанту за таблицею 1.

9. * Розрахуйте $y = n\sqrt{2n+1}/3$, где n - довільне ціле число, що вводиться з клавіатури;

* Розрахуйте

$$z_1 = \frac{\sin\left(\frac{\pi}{2} + 3\alpha\right)}{1 - \sin(3\alpha - \pi)}$$

$$z_2 = \operatorname{ctg}\left(\frac{5}{4}\pi + \frac{3}{2}\alpha\right)$$

* Виконати завдання 3 згідно варіанту за таблицею 1.

10. * Розрахуйте $y = \sqrt{8+n^3+n^2}/20$, где n - довільне ціле число, що вводиться з клавіатури;

* Розрахуйте

$$z_1 = \frac{1 - 2\sin^2 \alpha}{1 + \sin 2\alpha}$$

$$z_2 = \frac{1 - \operatorname{tg} \alpha}{1 + \operatorname{tg} \alpha}$$

* Виконати завдання 3 згідно варіанту за таблицею 1.

Завдання 3. Привласнити випадково (за допомогою методу random()) значення числовим цілим змінним **a1, a2, ..., a5** у зазначеному в квадратних дужках діапазоні. Вивести результат на екран по одному значенню, використовуючи методи print() або println().

Таблиця 1 - Варіанти завдання до лабораторної роботи №2

Варіант 1	Варіант 2	Варіант 3	Варіант 4	Варіант 5
[15÷25] a1 a2 a3 a4 a5	[100÷200] a1 a2 a3 a4 a5	[1÷25] a1 a2 a3 a4 a5	[-10÷10] a1 a2 a3 a4 a5	[-5÷25] a1 a2 a3 a4 a5
Варіант 6	Варіант 7	Варіант 8	Варіант 9	Варіант 10
[-150÷250] a1 a2 a3 a4 A5	[-35÷20] a1 a2 a3 a4 a5	[-75÷15] a1 a2 a3 a4 a5	[-5÷65] a1 a2 a3 a4 a5	[10÷100] a1 a2 a3 a4 a5

Завдання № 3

Тема: «Організація розгалуження з використанням умовного оператора та оператора switch. Умовна операція»

Мета роботи

Отримання навичок програмування алгоритмів розгалуженої структури за допомогою умовного оператора if та оператора варіанту switch.

Завдання до роботи

Використовуючи алгоритми, розглянуті на практичному занятті, скласти програму розрахунку заданих величин.

Методичні вказівки

■ Робота спирається на знання й уміння, отримані при вивченні наступних тем лекційного курсу:

- Умовна операція.
- Пріоритет операцій. Вираження.
- Організація розгалужень у програмі. Умовний оператор if.
- Оператор вибору switch.
- Мітки. Організація переривань за допомогою оператора break.

Тому під час підготовки до лабораторної роботи рекомендується повторити зазначені розділи дисципліни.

■ Алгоритми розгалуженої структури (розгалуження) - це така схема, у якій передбачене розгалуження зазначеної послідовності дій на два напрямки залежно від результату перевірки заданої умови. До операторів розгалуження в мові Java відносяться if, if else, ?: і switch.

■ В кожному варіанті лабораторної роботи є три завдання. Перше завдання треба виконати використовуючи **умовну операцію**. Друге завдання передбачає використання **вкладених умовних операторів**, а третє - **оператора вибору switch**.

■ При рішенні завдань можуть бути корисними наступні математичні формули:

■ Прямі $a_1 \cdot x + b_1 \cdot y + c_1 = 0$ і $a_2 \cdot x + b_2 \cdot y + c_2 = 0$ збігаються, якщо виконується наступна умова $\frac{a_1}{a_2} = \frac{b_1}{b_2} = \frac{c_1}{c_2}$, паралельні, якщо

$$\frac{a_1}{a_2} = \frac{b_1}{b_2} \neq \frac{c_1}{c_2}.$$

Якщо прямі перетинаються, то координати точки перетинання розраховуються за наступними формулами:

$$x = \frac{(c_1 b_2 - c_2 b_1)}{(b_1 a_2 - b_2 a_1)}; \quad y = \frac{(c_2 a_1 - c_1 a_2)}{(b_1 a_2 - b_2 a_1)}.$$

■ Відстань між двома точками, які задані координатами $a(x_1, y_1)$ і $b(x_2, y_2)$ розраховується за формулю: $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

- Площа трикутника за формулюю Герона

$$S = \sqrt{p \cdot (p - a) \cdot (p - b) \cdot (p - c)}, \text{ де } p = \frac{a + b + c}{2}.$$

$$a^2 = b^2 + c^2 - 2 \cdot b \cdot c \cdot \cos A$$

■ Теорема косинусів $b^2 = a^2 + c^2 - 2 \cdot a \cdot c \cdot \cos B$

$$c^2 = a^2 + b^2 - 2 \cdot a \cdot b \cdot \cos C$$

■ Площа довільного трикутника з відомими основою і висотою

$$S = \frac{1}{2} \cdot a \cdot h \cdot \sin C$$

■ Площа рівностороннього трикутника $S = \frac{a^2 \sqrt{3}}{4}$

■ Висота рівностороннього трикутника $h = \frac{a \sqrt{3}}{2}$

■ Площа трикутника, вписаного в коло $S = \frac{abc}{4R}$

■ Площа трикутника, описаного навколо кола $S = \frac{(a + b + c) \cdot r}{2}$

Приклад програми

* Обчислити значення функції з використанням умовної операції:

$$F(x) = \begin{cases} \cos(x), & \text{якщо } x > 5 \\ \sin(x), & \text{якщо } x \leq 5 \end{cases}$$

* Знайти суму мінімального та максимального з чисел a, b, c .

* Скласти програму, яка згідно введеному номеру місяця (1-12) виводить номер кварталу до якого він належить. Наприклад, 2 – I квартал.

```
import java.lang.Math;
import java.util.*;
class lab_3{
public static void main (String[] args){
//***** I *****/
Scanner scan = new Scanner (System.in);
```

```

System.out.println("Введіть x->");
double x=scan.nextDouble();
double f=(x>5)?Math.cos(x):Math.sin(x);
System.out.printf("F(x)= %5.3f",f);
System.out.println();
//***** II *****
System.out.println("Введіть a->");
double a=scan.nextDouble();
System.out.println("Введіть b->");
double b=scan.nextDouble();
System.out.println("Введіть c->");
double c=scan.nextDouble();
double min=a, max=a;
if(b>max) max=b; else min=b;
if(c>max) max=c;
if(c<min) min=c;
System.out.printf("Сума= %5.2f", (min+max));
System.out.println();
//***** III *****
System.out.println("Введіть n->");
int n=scan.nextInt();
switch (n) {
case 1:case 2:case 3:System.out.println("I квартал");break;
case 4:case 5:case 6:System.out.println("II квартал"); break;
case 7:case 8:case 9:System.out.println("III квартал");break;
case 10:case 11:case 12:System.out.println("IV квартал");
break;
default:System.out.println("Невірний номер"); }
}}

```

Контрольні питання

1. Який алгоритм називається алгоритмом розгалуженої структури?
2. Опишіть алгоритм роботи умовного оператора. Яка форма умовного оператора є повною? Скороченою?
3. Дайте визначення блоку операторів. У яких випадках він використовується?
4. Які умовні оператори називаються вкладеними? Назвіть правило роботи з вкладеними умовними операторами?
5. Приведіть синтаксис запису умовної операції.
6. Опишіть алгоритм роботи оператора вибору switch.

7. Для чого призначений оператор break?
8. Поясніть правила запису і використання міток в Java. Який блок називається позначеним?
9. Знайдіть і виправте помилку в наступному фрагменті програми.

```
double a = 6;
int b = 2, c;
switch (a/b) {
case 1: case 2: c = 1; break;
case 3: case 4: c = 2; break;
default: c = 3;}

```

Варіанти завдань

1. * Обчислити значення функції з використанням умовної операції:

$$F(x) = \begin{cases} \frac{a}{|a - |b - x||} \\ 1, \text{ якщо } _ \text{ знаменник} = 0 \end{cases}$$

* На площині задана точка a с координатами (x,y). Указати, де вона розташована: на якій осі або в якій координатній чверті.

* Нехай елементами прямокутного рівнобедреного трикутника є: 1) катет a; 2) гіпотенуза b; 3) висота, яка опущена з вершини прямого кута на гіпотенузу h; 4) площа S. Скласти програму, яка за заданим номером і значенням відповідного елемента обчислювала б значення всіх інших елементів трикутника.

2. * Дано точка M(x, y). Привласнити z = 1, якщо точка належить окружності з радіусом R і центром у точці (a, b) або z = 0 у протилежному випадку. Завдання вирішити з використанням умовної операції.

* Дано числа a1,b1,c1 и a2,b2,c2, які задають прямі a1·x+b1·y+c1=0 і a2·x+b2·y+c2=0, надрукувати координати точки перетинання прямих, або надрукувати, що прямі збігаються або паралельні.

* Написати програму, яка за номером дня тижня (натуральному числу від 1 до 7) видає назву дня тижня. Указати робочі та вихідні дні.

3. * Скласти програму рішення рівняння $y = \frac{1}{x^2 - 4}$ з використанням умовної операції.

* Дано три числа a , b , c . Визначити, яке з них дорівнює d . Якщо жодне не дорівнює d , то знайти $\max\{d-a, d-b, d-c\}$.

* Нехай елементами кола є радіус (перший елемент), діаметр (другий елемент) і довжина кола (третій елемент). Скласти програму, що за номером елемента запитувала б його відповідне значення та обчислювала б площу круга.

4. * Скласти програму обчислення модуля $y=|5x - 4|$ з використанням умовної операції.

* Дано три точки $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$. Визначити, чи будуть вони розташовані на одній прямій. Якщо ні, то обчислити периметр трикутника ABC .

* Знайти добуток цифр заданого n -значного числа. Допускаються числа до п'яти знаків включно.

5. * Скласти програму обчислення кореня $y = \sqrt{4x - 6}$ з використанням умовної операції.

* Визначити і надрукувати, який вигляд має трикутник з довільно заданими довжинами сторін a, b, c (довільний, рівнобедрений або взагалі не існує).

* Написати програму, яка б за уведеним номером одиниці виміру (1 — дециметр, 2 — кілометр, 3 — метр, 4 — міліметр, 5 — сантиметр) і довжині відрізка L видавала б відповідне значення довжини відрізка в метрах.

6. * Скласти програму обчислення функції $y=\ln(3x-6)$ з використанням умовної операції.

* Дано число x . Надрукувати в порядку зростання числа: $\sin x$, $\cos x$, $\ln x$. Якщо при якому-небудь x деякі з виражень не існують, то вивести відповідні повідомлення і порівнювати значення тільки ті, значення яких існують.

* Дано три числа a , b , c , які задовольняють аксіомі трикутника і число p . Виконати наступні дії: якщо $p=1$ - знайти периметр трикутника; якщо $p=2$ - знайти площу трикутника; якщо $p=3$ - знайти кут a . Інакше надрукувати слово "трикутник".

7. * Задано коло з радіусом R і прямокутний отвір зі сторонами a, b . Привласнити $f = 1$, якщо коло пройде через отвір і $f = 0$ у протилежному випадку. Завдання вирішити з використанням умовної операції.

* Відомо, що із чотирьох чисел a_1, a_2, a_3 і a_4 одне відмінне від трьох інших, рівних між собою; привласнити номер цього числа змінної n .

* Є серія вимірів елементів трикутника. Групи елементів пронумеровані. У серії в довільному порядку можуть зустрічатися такі групи елементів трикутника: 1. Основа та висота; 2. Дві сторони і кут між ними (кут заданий у радіанах); 3. Три сторони.

Розробити програму, яка запитує номер групи елементів, вводить відповідні елементи та обчислює площу трикутника. Обчислення припинити, якщо введений номер групи 0.

8. *Задано прямокутник зі сторонами a, b і круглий отвір радіуса R . Привласнити $f = 1$, якщо прямокутник пройде в отвір або $f = 0$ у протилежному випадку. Завдання вирішити з використанням умовної операції.

* Дано дійсні числа a, b, c ($a > 0$). Повністю дослідити біквадратне рівняння $ax^4 + bx^2 + c = 0$, тобто якщо дійсних коренів немає, то повинне бути видане повідомлення про це, інакше знайти дійсні корені, повідомивши, скільки з них є різними.

* Для цілого числа k від 1 до 99 надрукувати фразу "Мені k років", з розрахунком на те, що при деяких значеннях k слово "років" треба замінити на слово "рік" або "року". Наприклад, 11 років, 22 року, 51 рік.

9. * Обчислити значення функції з використанням умовної операції:

$$F(x) = \begin{cases} \frac{\cos(a)}{2a - |3b - x|} \\ 1, \text{ якщо } _ \text{ знаменник} = 0 \end{cases}$$

* За номером деякого року $Year$ ($Year > 0$), визначити номер сторіччя. При цьому вважати, що сторіччя починається в першому році, а не в нульовому.

* Для цілого k у діапазоні від 1 до 100 надрукувати фразу "У лісі ми знайшли k грибів". Обов'язково узгодити закінчення слова "гриб" з числом k .

10. *Скласти програму обчислення кореня $y = \sqrt{3 - 6x}$ з використанням умовної операції.

* Дано відрізки $[a, b]$ і $[c, d]$ і точка A з координатою x . Визначити, чи належить дана точка одному із цих відрізків, обом або лежить поза них.

* Нехай елементами рівностороннього трикутника є: 1) сторона a ; 2)

площа S ; 3) висота h ; 4) радіус вписаного кола r ; 5) радіус описаного кола R . Скласти програму, що по заданому номеру i значенню відповідного елемента обчислювала б значення всіх інших елементів трикутника.

Завдання № 4

Тема: «Складання програм циклічної структури. Цикли `while`, `do i for`»

Мета роботи

Отримання навичок програмування ітераційних алгоритмів та алгоритмів циклічної структури за допомогою циклів `while`, `do i for`.

Завдання до роботи

Використовуючи алгоритми, розглянуті на практичному занятті, скласти програму розрахунку заданих величин.

Методичні вказівки

■ Робота спирається на знання й уміння, отримані при вивченні наступних тем лекційного курсу:

- Цикли. Ітераційні алгоритми.
- Циклічні конструкції `while i do- while`.
- Цикл з параметром `for`.
- Оператор кома. Оператор `continue i return`.
- Мітки. Організація переривань за допомогою оператора `break`.

Тому під час підготовки до лабораторної роботи рекомендується повторити зазначені розділи дисципліни.

■ При рішенні ряду завдань виникає необхідність багаторазового повторення однотипних дій при різних значеннях параметрів, що визначають ці дії. Алгоритми, що реалізують такі обчислення, називаються циклічними, а повторювані ділянки обчислень - циклами. Використання циклів дозволяє виконати більші об'єми обчислень за допомогою компактних програм. Розрізняють цикли із заданим і невідомим числом повторень. До останнього відносяться ітераційні цикли з необхідною

точністю. Бувають цикли з перевіркою умови перед початком виконання тіла циклу, по закінченню виконання тіла або усередині тіла циклу.

Приклад програми

* Знайти $y = \ln(2 + \ln(4 + \dots (20 + \ln(22)) \dots))$;

* Знайти суму членів ряду $\sum_{n=1}^{\infty} \frac{n+1}{2n!}$ с заданою точністю $EPS < 10^{-4}$

```
import java.lang.Math;
class lab_4{
public static void main (String[] args){
//***** I *****
double y=Math.log(22);
for(int i=20; i>=2; i-=2)
y=Math.log(i+y);
System.out.printf("Y= %6.3f",y);
System.out.println();
//***** II *****
double sum=0;
double EPS=1E-4;
double n=1, f=1;
while(Math.abs((n+1)/(2*f))>=EPS)
{sum+=(n+1)/(2*f); n++; f*=n;}
System.out.printf("Сумма= %5.2f",sum);
System.out.println();
}}
```

Контрольні питання

1. Який алгоритм називається ітераційним? Дайте визначення ітераційним циклам.

2. Опишіть алгоритми виконання циклів while і do. У чому їхня відмінність?

3. Приведіть синтаксис запису оператора циклу for і опишіть алгоритм його роботи.

4. Як організувати нескінченний цикл за допомогою оператора for? While?

5. Чим відрізняється дія операторів break и continue?
6. Яке призначення має оператор кома?

Варіанти завдань

1. *Обчислити найбільший спільний дільник натуральних чисел А і В.
*Обчислити $y = \sin 1 + \sin 1.1 + \sin 1.2 + \dots + \sin 1.N$ (де N - ціле число ≤ 10).

*Використовуючи формулу прямокутників, приблизно обчислити визначений інтеграл в межах 0 - π від функції $\ln(2 + \sin x)$. Відповідно цій формулі інтеграл в межах від А до В для функції f(x) дорівнює

$$h * (f(x_0) + f(x_1) + \dots + f(x_{N-1})),$$

где $h = (B-A)/N$, $x_i = A + i * h - h/2$. В нашому випадку вважати N=100.

2. *Знайти перший від'ємний член послідовності $\cos(\text{ctg } N)$, коли N приймає значення 1, 2, 3,

*Знайти, на якому кроці в ряду $\sin X$, $\sin(\sin X)$, $\sin(\sin(\sin X))$, ... зустрінеться перше число, що за модулем менше за $5 \cdot 10^{-2}$. Надрукувати номер крока і число.

*Обчислити суму квадратів всіх цілих чисел, які потрапляють в інтервал $(\ln X, e^X)$, явно, що X повинен бути більше 1.

3. * Обчислити $p = \left(1 - \frac{1}{2^2}\right) \cdot \left(1 - \frac{1}{3^2}\right) \cdot \dots \cdot \left(1 - \frac{1}{n^2}\right)$, при $n \geq 10$.

* Обчислити $y = \cos X + \cos X^2 + \dots + \cos X^N$.

* Обчислити $1! + 2! + 3! + \dots + n!$

4. *Обчислити $y = \cos(1 + \cos(2 + \dots \cos((n-1) + \cos(n) \dots))$, при $n \geq 40$.

*Логічній змінній присвоїти значення true, якщо серед чисел $\sin X^N$ є хоча б одне від'ємне число. N повинно змінюватися від 1 до 30.

*Обчислити
$$S = \sum_{k=1}^{12} \frac{\sum_{n=1}^k \sin kn}{k!}$$

5. *Підрахувати кількість цифр в десятковому запису цілого невід'ємного числа K.

*Обчислити
$$\sqrt{3 + \sqrt{6 + \dots + \sqrt{96 + \sqrt{99}}}}$$

* Обчислити
$$p = \prod_{i,j=1}^{10} \frac{1}{i + j^2}$$

6. *Обчислити за схемою Горнера значення багаточлена виду:

$$y = x^{10} + 2x^9 + 3x^8 + \dots + 10x + 11.$$

* Обчислити $y = (2n)!! = 2 \cdot 4 \cdot \dots \cdot (2n)$, n ввести > 0 .

*Логічній змінній присвоїти true, якщо введене ціле число N являється простим. Інакше логічна змінна повинна отримати значення false.

7. *Обчислити за схемою Горнера значення багаточлена виду:

$$y = 11x^{10} + 10x^9 + 9x^8 + \dots + 2x + 1.$$

*Обчислити $y = (2n-1)!! = 1 \cdot 3 \cdot 5 \cdot \dots \cdot (2n-1)$, n ввести ($n > 0$).

*Визначити кількість п'ятизначних натуральних чисел, сума цифр яких дорівнює заданому числу N (видно, що $1 \leq N \leq 45$). При обчисленнях неможна використовувати операції ділення ($/$, $\%$).

8. *Не використовуючи стандартні функції (за виключенням abs), Обчислити з точністю $Eps > 0$: $y = e^x = 1 + x/1! + x^2/2! + \dots + x^n/n! + \dots$. Вважати, що необхідна точність досягнена, якщо чергова складова за модулем менше Eps , — всі останні складові можна не враховувати.

*Визначити чи являється задане число досконалим, тобто дорівнює сумі всіх (додатних) дільників, крім самого цього числа (наприклад, число 6 досконале: $6 = 1 + 2 + 3$).

*Знайти двохзначне число, яке дорівнює строєному добутку його цифр.

9. *Обчислити з похибкою не більше 10^{-4} суму ряду

$$S = 1 + x^2/2! + x^4/4! + x^6/6! + \dots + x^{2n}/(2n)! + \dots$$

*Знайти тризначне число, що дорівнює добутку його цифр, помноженому на вісім (8). Не використовувати операції $/$, $\%$.

*Використовуючи формулу прямокутників, приблизно обчислити визначений інтеграл на інтервалі $[-\pi/2, \pi/2]$ від функції $\sqrt{|x| + \cos^2(x)}$.

Відповідно цій формулі інтеграл в межах від A до B для функції $f(x)$ дорівнює $h \cdot (f(x_0) + f(x_1) + \dots + f(x_{N-1}))$,

де $h = (B-A)/N$, $x_i = A + i \cdot h - h/2$. В нашому випадку $N=80$.

10. *Не використовуючи стандартні функції (за виключенням Abs), обчислити з точністю $Eps > 0$: $y = \cos(x) = 1 - x^2/2! + x^4/4! - x^6/6! + \dots + (-$

$$1)^n \cdot x^{2n} / (2n)!$$

Вважати, що необхідна точність досягнена, якщо чергова складова по модулю менше ϵ_{ps} , — всі останні складові можна не враховувати.

* В яких двохзначних числах здвоєна сума цифр дорівнює їх добутку?

* Знайти суму квадратів парних чисел від 2 до 20 (2, 4, 6, 8, 10, 12, 14, 16, 18, 20).

Завдання № 5

Тема: «Лінійні масиви»

Мета роботи

Отримання практичних навичок програмування алгоритмів обробки масивів. Вивчення алгоритмів упорядкування масивів методом вставки, вибору та лінійного сортування.

Завдання до роботи

Використовуючи алгоритми, розглянуті на практичному занятті, скласти програму розрахунку заданих величин.

У роботі для ініціалізації елементів масиву слід використовувати генератор випадкових чисел. Вибирати $n > 20$. Виводити значення в рядок по 10 елементів.

Методичні вказівки

■ Робота спирається на знання й уміння, отримані при вивченні наступних тем лекційного курсу:

- Масиви. Опис масивів.
- Операції над масивами.
- Методи сортувань елементів масивів.

Тому під час підготовки до лабораторної роботи рекомендується повторити зазначені розділи дисципліни.

■ Розглянемо деякі прості алгоритми сортувань лінійних масивів:

Алгоритм сортування лінійного цілого масиву за зростанням методом вибору

```
int i, j, nom, min;
for( i=0; i < n-1; i++) {
```

```

        nom=i; min=a[i];
        for( j=i+1; j < n; j++)
            if ( a[j] < min )
    {   nom=j; min=a[j]; }
        a[nom] = a[i]; a[i] = min; }

```

Алгоритм сортування цілого масиву за зростанням методом лінійного сортування

```

int i, j, x;
for( i=0; i < n-1; i++) {
    for( j = i+1; j < n; j++) {
        if ( a[i] > a[j] ) {
            x=a[i]; a[j]=a[i]; a[j]=x;
        }
    }
}

```

Алгоритм сортування лінійного цілого масиву за зростанням методом вставки

```

int i, j, x;
for (i=1; i < n; i++)
    { x = a[i];
    for ( j=i-1; j>=0 && a[j] > x; j--)
        a[j+1] = a[j]; a[j+1] = x;}

```

Приклад програми

Для лінійного масиву, що містить n дійсних елементів, розрахувати суму за модулем всіх його елементів. "Стиснути" масив, видаливши з нього всі від'ємні елементи. Упорядкувати елементи нового масиву за зростанням методом вставки.

```

import java.util.*;
class lab_5{
public static void main (String[] args) {
Scanner scan = new Scanner (System.in);
System.out.println("Введіть n->");
int n=scan.nextInt();
double[] a=new double[n];
for(int i=0;i<a.length;i++)
a[i]=Math.random()*20-10;

System.out.println("Исходный массив");
for(int i=0;i<a.length;i++)
{System.out.printf("%7.2f",a[i]);
if(i%10==9) System.out.print("\n");}

double sum=0;

```

```

for(int i=0;i<a.length;i++)
sum+=Math.abs(a[i]);
System.out.print("\n");
System.out.printf("Сумма = %7.2f",sum);
System.out.println();

int col=0;
double[] b=new double[n];
for(int i=0;i<a.length;i++)
if(a[i]>=0) {b[col]=a[i]; col++;}
System.out.print("\n");

System.out.println("Сжатый массив");
for(int i=0;i<col;i++)
{System.out.printf("%7.2f",b[i]);
if(i%10==9) System.out.print("\n");}
int i,j;
double x;
for(i=1; i<col; i++)
{ x=b[i];
for(j=i-1; j>=0&& b[j]>x;j--)
b[j+1]=b[j]; b[j+1]=x;}
System.out.print("\n");

System.out.println("Упорядоченный массив");
for(i=0;i<col;i++)
{System.out.printf("%7.2f",b[i]);
if(i%10==9) System.out.print("\n");}
System.out.print("\n");
}}

```

Контрольні питання

1. Дайте визначення масиву.
2. Який формат має оголошення масиву?
3. Як здійснюється виділення пам'яті під масив? Приведіть приклади.
4. Як здійснюється доступ до елементів масиву? Значення яких типів можуть бути індексами масиву? Як визначити довжину масиву?
5. Які операції можуть бути застосовані до посилань на масиви?
6. Як у програмі організувати введення та виведення масивів?
7. Опишіть процедури упорядкування елементів масиву при використанні різних методів сортувань?

Варіанти завдань

1. Дано масив дійсних чисел, що містить n елементів. Елементи масиву не можуть дорівнювати нулю. Розрахувати корінь ступеня n з абсолютного значення добутку всіх елементів масиву (середнє геометричне всіх елементів). "Стиснути" масив, видаливши з нього кожний другий елемент. Упорядкувати елементи нового масиву за зростанням **методом лінійного сортування**.
2. Дано масив дійсних чисел, що містить n елементів. Обчислити суму всіх елементів масиву так, щоб парні елементи входили до неї зі зворотними знаками. Сформуйте два масиви: у перший включіть елементи вихідного масиву з парними номерами, а в другий - з непарними. Масив з парними номерами відсортуйте за убутанням **методом вибору**.
3. Дано масив додатних цілих чисел, що містить n елементів. Обчислити суму добутків першого елемента на останній, другого на передостанній і т.д. Утворити новий масив, елементами якого будуть елементи вихідного, що закінчуються на цифру k . Упорядкувати новий масив за зростанням **методом вставки**.
4. Дано масив дійсних чисел, що містить n елементів. Обчислити суму квадратів всіх елементів масиву. Знайти в масиві мінімальний і максимальний елемент. Упорядкувати елементи масиву, що розташовані між мінімальним і максимальним за убутанням **методом лінійного сортування**.
5. Дано масив дійсних чисел, що містить n елементів. Знайти середнє арифметичне всіх елементів масиву. В одномірному масиві всі від'ємні елементи слід перемістити в початок масиву, а інші - у кінець зі збереженням порядку розташування. Від'ємні елементи упорядкувати за зростанням **методом вибору**.
6. Дано масив цілих чисел, що містить n елементів. Обчислити значення багаточлена ступеня n у точці X_0 , вважаючи елементи масиву коефіцієнтами цього багаточлена. "Стиснути" масив, видаливши з нього всі нульові елементи. Упорядкувати новий масив за убутанням **методом вставки**.

7. Дано масив дійсних чисел, що містить n елементів. Знайти суму добутків непарних елементів (елементів з непарними номерами) на наступні за ними парні елементи (з парними номерами). Переставити елементи масиву так, щоб на початку стояли додатні, потім нульові і від'ємні значення в порядку їхнього розташування. Від'ємні елементи упорядкувати за зростанням **методом лінійного сортування**.

8. Дано масив дійсних чисел, що містить n елементів. Знайти кількість елементів більших за середнє арифметичне. "Стиснути" масив, видаливши з нього всі від'ємні елементи. Упорядкувати першу половину нового масиву за убутанням **методом вибору**.

9. Дано масив додатних цілих чисел, що містить n елементів. Перевірити, чи виконується для елементів масиву наступне відношення $a[i] \leq a[i+1] \leq a[i+2]$ і видати відповідне повідомлення. Утворити новий масив, елементами якого будуть елементи вихідного, кратні k . Упорядкувати новий масив за зростанням **методом вставки**.

10. Дано масив цілих чисел, що містить n елементів. Знайти номер останнього парного від'ємного елемента. Якщо цей елемент перебуває в першій половині масиву, то знайти кількість елементів рівних першому парному від'ємному елементу. Упорядкувати другу половину масиву за убутанням **методом лінійного сортування**.

3. Організація контролю знань та вмінь студентів

3.1. Система контролю знань та вмінь студентів

Контроль знань та вмінь студентів, що навчаються за заочною формою, здійснюється за допомогою системи контролюючих заходів. Вони складаються з заходів поточного та підсумкового контролю. Поточний контроль здійснюється на протязі всього навчального року (семестру) та включає заходи контролю самостійної роботи студента під час вивчення навчальної дисципліни поза межами університету та роботи студента на лабораторних заняттях у період заліково-екзаменаційної сесії.

Підсумковий контроль здійснюється під час заліково-екзаменаційної сесії та має на меті встановлення рівня знань та вмінь, які опанував студент

після вивчення навчальної дисципліни. Форма підсумкового контролю - іспит - встановлюється навчальним планом дисципліни.

При вивченні дисципліни „Алгоритмізація та програмування” на 1 курсі заочного факультету використовується накопичувальна система оцінювання. Форма підсумкового контролю – іспит. Головними рисами накопичувальної системи є підсумкова оцінка знань студента складається як арифметична сума оцінки, яку накопив студент, виконуючи заходи поточного контролю, та оцінки, яку отримав студент на підсумковому контролі (іспит).

3.2 Форми контролю знань та вмінь студентів

1) Оцінка самостійної роботи студента до екзаменаційно-залікової сесії здійснюється у формі оцінки виконання контрольної роботи.

Максимальний бал за контрольну роботу №1 становить 60 балів, з них на теоретичну частину контрольної роботи припадає 30 балів і на практичну частину – 30 балів.

Контрольна робота вважається зарахованою, якщо студент одержав сумарну оцінку не менше 36 балів (18 балів з теоретичної частини і 18 балів з практичної частини), тобто не менше 60% від максимальної суми в 60 балів. Не зарахована контрольна робота свідчить про те, що студент одержав сумарну оцінку меншу за 36 балів, в цьому випадку вона повертається на доробку. Зарахована контрольна робота є допуском до здачі іспиту.

2) Оцінки роботи студента при проведенні лабораторних занять по дисципліні під час екзаменаційно-залікової сесії.

Загальна максимальна оцінка за цей вид поточного контролю складає 40 балів і включає: оцінку з теоретичних модулів (виконання контрольної роботи – 9 балів, захист реферату – 10 балів), оцінку з практичних модулів (усне опитування під час лабораторних робіт – 9 бали, захист виконаних лабораторних робіт – 12 балів).

- Кількісну оцінку заходу підсумкового контролю (ОПК).

Цей захід передбачає оцінювання результатів іспиту, який виконується в період заліково-екзаменаційної сесії.

В екзаменаційних білетах для 1 курсу використовуються екзаменаційні білети у вигляді тестових завдань закритого типу, загальна екзаменаційна оцінка (бал успішності) еквівалентна відсотку правильних відповідей із загального обсягу питань екзаменаційного білету (20 питань).

Таблиця для визначення кількісних показників успішності за підсумками іспиту на 1 курсі:

За системою університету (у відсотках)	Визначення
90 – 100	відмінне виконання лише з незначною кількістю помилок
85 – 89	вище середнього рівня з кількома помилками
75 – 84	в загальному правильна робота з певною кількістю грубих помилок
68 – 74	непогано, але зі значною кількістю помилок
60 – 67	виконання задовольняє мінімальні критерії
35 – 59	з можливістю перескласти
1 – 34	з обов'язковим повторним курсом навчання

- Система накопичувальної підсумкової оцінки засвоєння студентами початкової дисципліни (ПО).

Накопичена підсумкова оцінка засвоєння студентами заочної форми навчальної дисципліни розраховується, як:

$$ПО=0,5ОПК+0,25(ОЗЕ+ОМ).$$

Загальна підсумкова оцінка знань студента з іспиту складається як арифметична сума, яку накопив студент у міжсесійний період, під час заліково-екзаменаційної сесії та складання іспиту. Таким чином студент може одержати максимально 100 балів.

Якісна оцінка є такою:

- 90 і більше – відмінно;
- 76-89 балів – добре;
- 60-74 балів – задовільно;
- менше 60 балів – незадовільно.
- менше 60 балів – незадовільно.

Етапи виконання Java-програми та особливості роботи із середовищем розробки BlueJ

Установка BlueJ

BlueJ є середовищем розробки мовою Java, що було створено командою розроблювачів з Університету Monash, Мельбурн, Австралія спеціально для навчання на початковому рівні. Система вільна для поширення. Додаткова інформація про BlueJ доступна в Internet за адресою <http://www.bluej.monash.edu>.

BlueJ поширюється у вигляді архіву класів Java, у «jar» форматі. Обов'язковою вимогою до установки є наявність встановленого спеціального програмного забезпечення - Java Development Kit (JDK). Це програмне забезпечення можна безкоштовно завантажити з веб-сайту компанії Oracle (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>).

Дистрибутивний файл із BlueJ має ім'я *bluej-xxx.jar*, де xxx – номер версії. При установці середовища розробки в Windows слід виконати подвійне клацання мишею на файлі *bluej-xxx.jar*. Далі буде створений пакетний файл *bluej* у каталозі BlueJ, після чого можна запустити BlueJ просто подвійним клацанням з GUI інтерфейсу.

Розглянемо найбільш важливі на першому етапі вивчення редактора особливості створення, редагування і виконання програм в BlueJ.

Створення проекту

Для створення нового проекту з меню слід вибрати Project – New... Відкриється діалогове вікно, у якому можна вказати ім'я та розташування нового проекту. Після натискання ОК, буде створена директорія з відповідним ім'ям і відкриється порожнє вікно нового проекту подібно тому, що представлено на рис.1.

Створення класу

Будь-яка програма мовою Java оформляється у вигляді класу. Створити свій власний клас можна, клацнувши на кнопку *New Class* на панелі інструментів у вікні проекту. BlueJ запросить ім'я класу, яке повинне бути припустимим в Java ідентифікатором. Після створення класу, він буде представлений окремою піктограмою у вікні проекту. Двічі клацнувши на даній піктограмі можна відкрити редактор з новим класом, у

якому частина класу вже створена для полегшення початку роботи (рис.2). Далі можна приступати до написання коду програми.

Збільшити розмір шрифту у вікні редактора класу можна нажавши в головному меню редактора кнопку *Options-Preferences...* У діалоговому вікні *Preferences* у вкладці *Editor* можна задати розмір шрифту.

Щоб видалити клас із проекту, потрібно виділити його й потім вибрати *Remove Class* з меню *Edit* або вибрати *Remove* з контекстного меню класу.

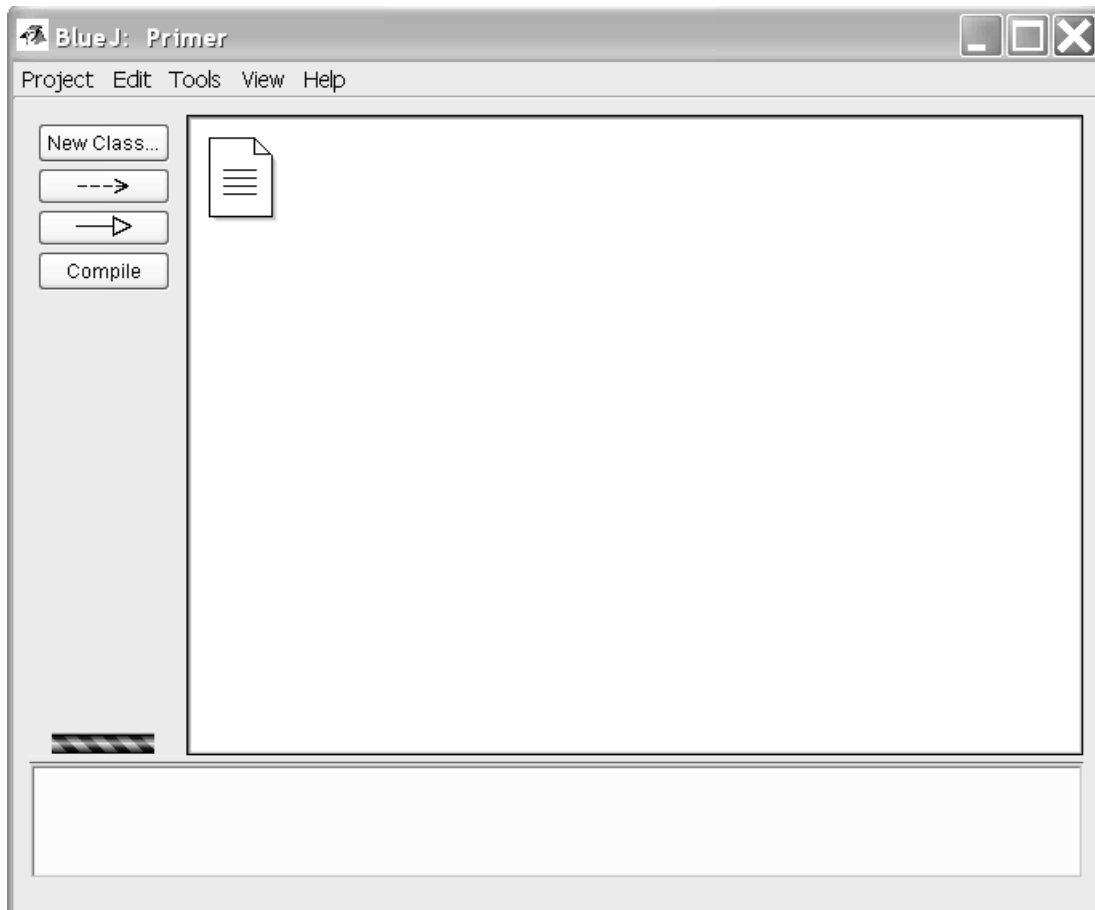


Рисунок А.1 – Вид головного вікна BlueJ

Компіляція

Перед запуском програми на виконання її треба перевірити на наявність синтаксичних помилок, тобто відкомпілювати. Зверніть увагу, що на піктограмі класу `my_class` рис.2 є заштрихована область. Штрихування позначає клас, що не був відкомпільований після останніх змін.

На панелі інструментів вікна редактора, розташовуються кнопки для функцій, що часто використовуються. Одна з них *Compile*. Ця кнопка дозволяє відкомпілювати клас прямо з вікна редактора. Клацнувши на

кнопку *Compile*, за умови, що в кодї програми немає помилок, унизу вікна редактора з'явиться повідомлення, про те, що компіляція виконана (рис.3). Якщо є синтаксичні помилки, рядок, що містить її, буде підсвічений й повідомлення про помилку з'явиться в інформаційній області.

Панель інструментів вікна проекту також має кнопку *Compile*. Однак натискання на неї приводить до компіляції всього проекту. Відкомпілювати ж окремий клас можна з вікна проекту викликавши на відповідній піктограмі класу контекстне меню і пункт *Compile*.

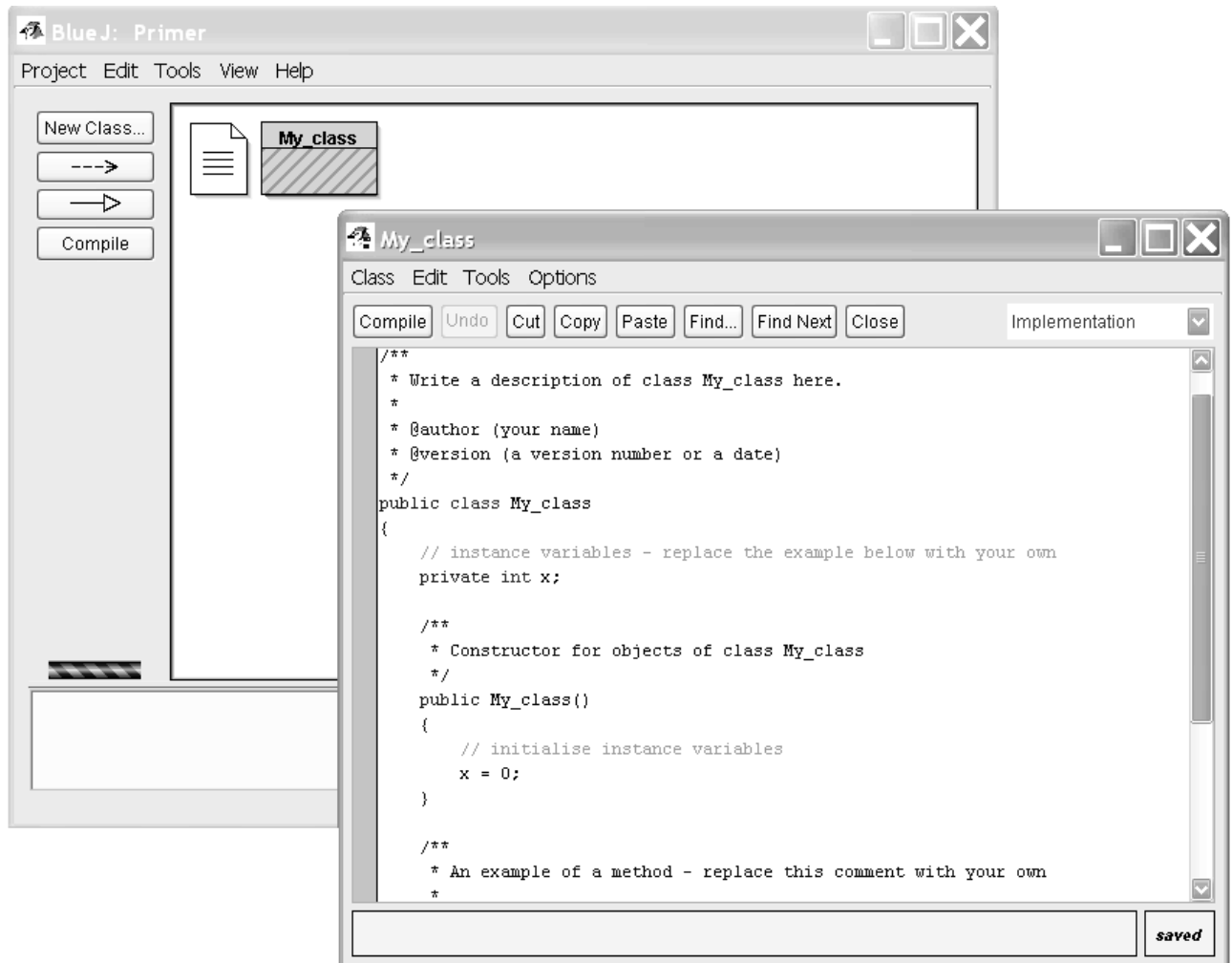


Рисунок А.2 – Вид вікна редактора класу

Пам'ятайте, що немає необхідності примусово зберігати вихідний код. Він зберігається автоматично при закритті редактора та перед компіляцією.

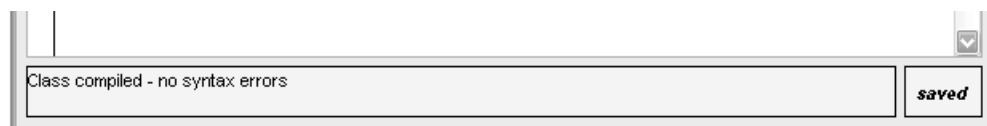


Рисунок А.3 – Вигляд повідомлення про успішну компіляцію програми

Допомога при помилках компіляції

Повідомлення про помилки компіляції висвітлюються в інформаційній області. У правому нижньому куті з'являється знак питання, клацнувши на який, можна одержати додаткову інформацію про тип даної помилки (рис.4). На жаль, додаткова інформація доступна не для всіх помилок, але багато з них описані.

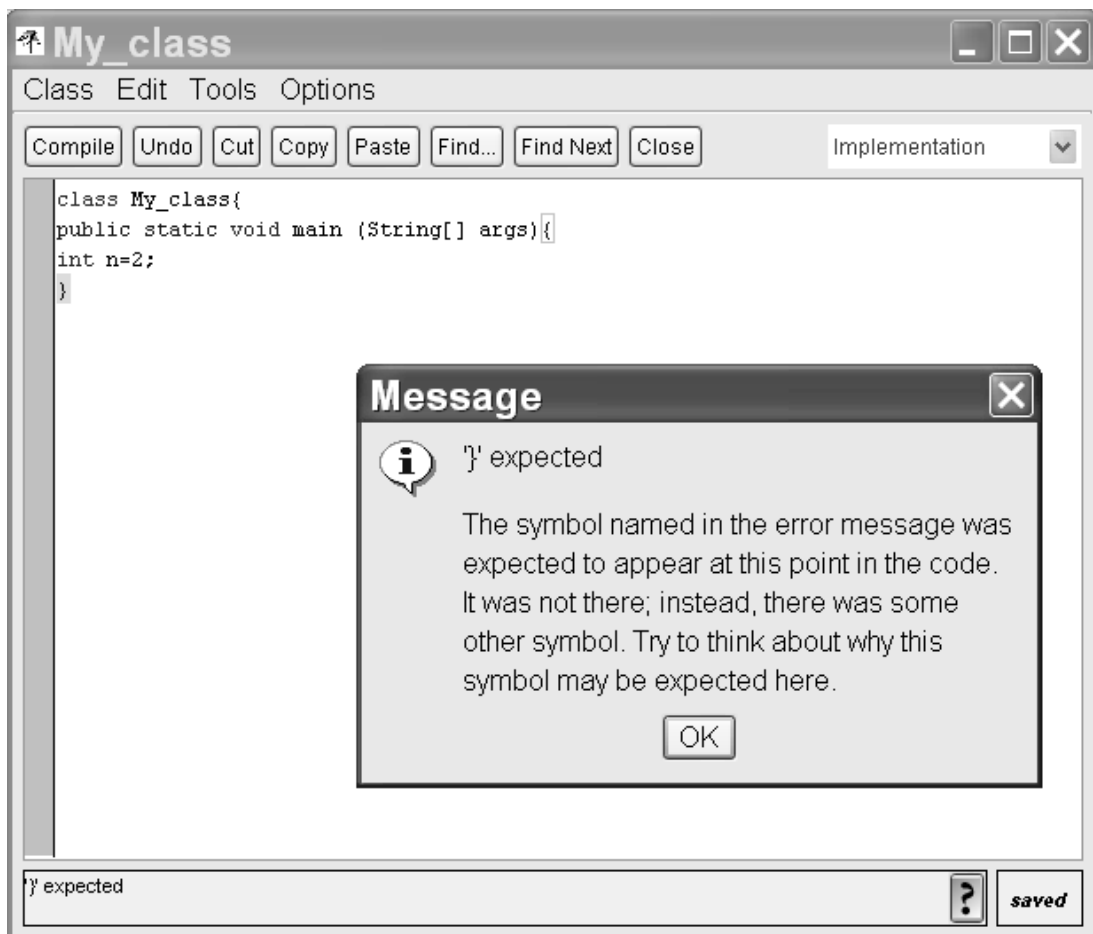


Рисунок А.4 - Помилка компіляції і кнопка допомоги

Запуск програми на виконання і виклик методу main()

Запуск програми на виконання звичайно починається з виклику методу main(), який в BlueJ може бути викликаний безпосередньо з контекстного меню класу. Якщо клацнути на класі правою кнопкою миші, то метод main() з'явиться в контекстному меню (мал.5). Стандартний метод

main() припускає передачу в якості параметра масиву рядків. Можна передати масив рядків, використовуючи стандартну нотацію Java для масивів констант. Наприклад: {"one", "two", "three"} обов'язково, включаючи, лапки.

Додавання існуючих класів у проект

Часто виникає необхідність включити в проект раніше створені класи, які використовуються десь в іншому місці. Наприклад, у лабораторних роботах для введення даних із клавіатури буде використовуватися методи класу Xterm. Для того, щоб можна було їх викликати, треба додати даний клас у проект. Це можна зробити використовуючи меню - Edit - Add Class from File.... Необхідний Java файл буде знайдений і включений у проект.

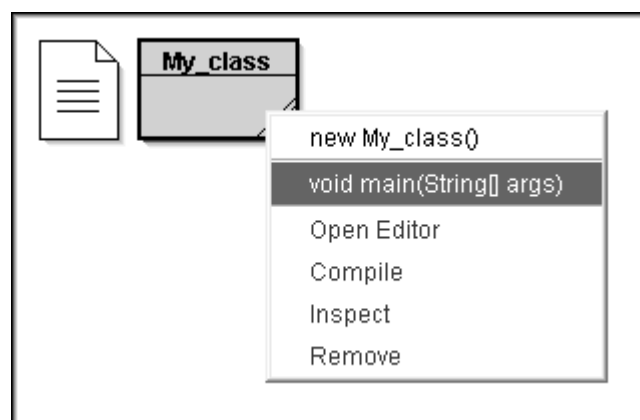


Рисунок А.5 – Виклик методу main()

Після імпортування класу в проект, його копія буде створена в каталозі цього проекту. Альтернативний метод - скопіювати вихідний код класу в каталог без участі BlueJ.

Переривання виконання програми

Іноді, виникає необхідність примусово перервати виконання програми. Це особливо актуально в ситуації її "зависання". Стан виконання програми відображається у світному барі, розташованому внизу вікна редактора проекту. Перервати виконання програми можна викликавши контекстне меню на барі стану та вибрав пункт *Reset* (мал.6) або набравши комбінацію *Ctrl+Shift+R*.



Рисунок А.6 – Переривання виконання програми Налагодження програми

Розглянемо найбільш важливі аспекти налагодження програми в BlueJ. Функціональність відладчика зведена до трьох завдань:

- призначення крапок переривання;
- покроковий прохід по коду;
- перевірка значень змінних.

Установка крапок переривання

Установка крапок переривання дозволяє перервати виконання в певній крапці коду. Після того, як переривання здійсниться, стає можливим контролювати стан змінних. Це часто допомагає зрозуміти, що відбувається в коді. У редакторі, ліворуч від тексту, розташовується зона для установки крапок переривання (рис.7). Їх можна встановити просто клацнувши в цій області мишею. Невелика піктограма "stop" відзначить крапку переривання.

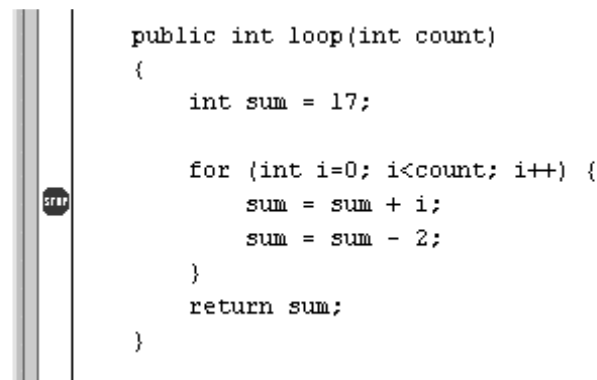


Рисунок А.7 - Крапка переривання

Як тільки виконання коду дійде до цього рядка - відбудеться припинення виконання програми. Як тільки крапка "stop" буде досягнута, відкриється редактор і вікно відладчика *Debugger*, як на рис.8.

Підсвічений рядок указує на рядок, що буде виконаний в наступний момент.

Покрокове виконання коду

Після припинення програми можна здійснити покрокове виконання коду. Для того, щоб виконати наступний рядок програми, треба клацнути на кнопку Step у вікні відладчика. Підсвічений рядок зрушиться на крок

униз. Щоразу при натисканні кнопки *Step*, один рядок коду буде виконуватися, а виконання програми припинятися. Значення змінних у вікні відладчика теж змінюються, у такий спосіб можна просуватися крок за кроком і спостерігати, що при цьому відбувається зі змінними. Якщо потрібно перервати покрокове виконання програми, то слід клацнути на крапку переривання знову, щоб усунути її. Для продовження роботи додатка у звичайному режимі треба клацнути на кнопку *Continue*.

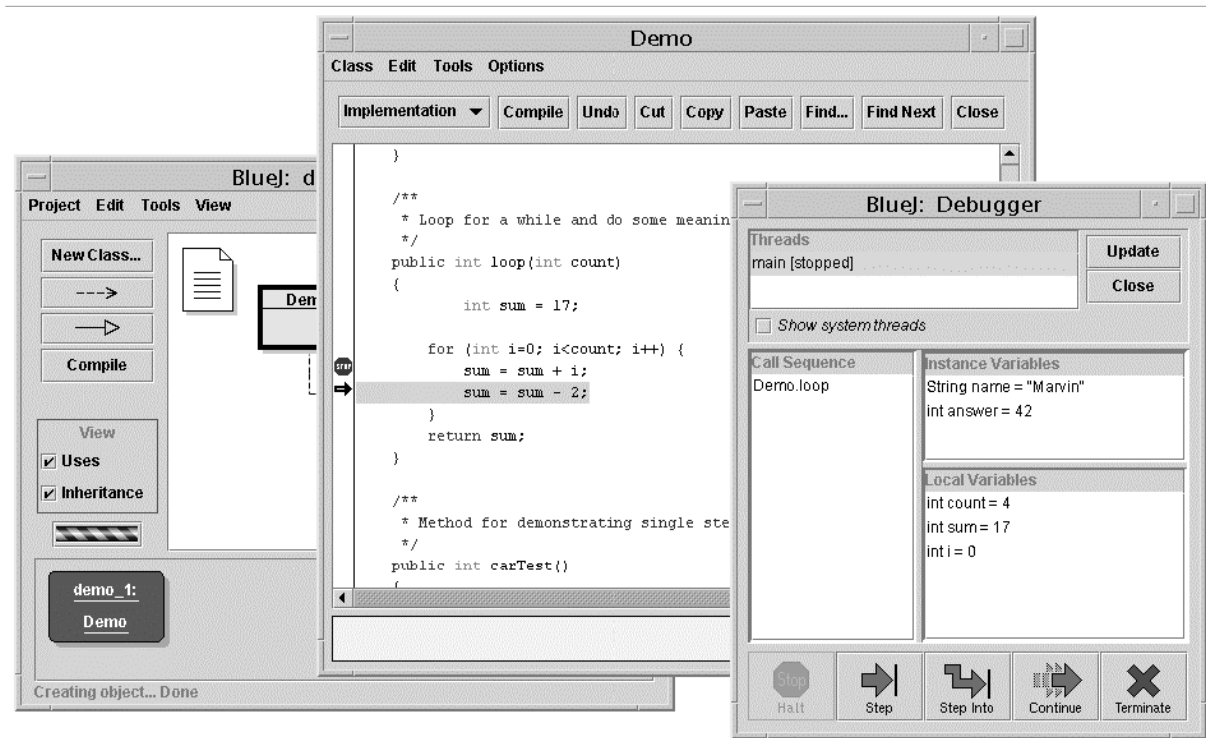


Рисунок А.8 - Вікно відладчика

У випадку коли програма працює занадто довго (перебуває в режимі зависання), можна відкрити вікно відладчика з меню *View - ShowDebugger* і клацнути на кнопку *Halt*. Виконання перерветься, начебто вставили крапку переривання. Далі можна зробити декілька кроків по програмі і переконатися, що все в порядку. Можна використовувати *Continue* і *Halt* стільки разів, скільки буде необхідно. Якщо немає необхідності продовжувати роботу додатка, то можна клацнути на кнопку *Terminate*, і виконання припиниться. *Terminate* не можна використовувати часто, тому що ця операція залишає об'єкти в невизначеному стані, тому цей засіб краще застосовувати для кінцевої зупинки додатка.

МЕТОДИЧНІ ВКАЗІВКИ

до самостійної роботи студентів та
виконання контрольної роботи з дисципліни

“Алгоритмізація та програмування”

для студентів I року заочної форми навчання
Спеціальність – комп’ютерні науки

Укладачі: доц. Кузніченко С.Д., доц. Коваленко Л.Б.

Підп. до друку

Формат 60x84/16 Папір офс.

Умовн. а.а.

Тираж

Замовл.

Одеський державний екологічний університет,
65016, м. Одеса, вул. Львівська, 15
