

**НАЦІОНАЛЬНА АКАДЕМІЯ УПРАВЛІННЯ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК
КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ**

МЕТОДИЧНІ ВКАЗІВКИ

**щодо виконання лабораторних робіт
з курсу “Об'єктно-орієнтоване програмування”
для студентів спеціальності “Інтелектуальні системи
прийняття рішень” денної та заочної форм навчання**

КИЇВ - 2008

Методичні вказівки щодо виконання лабораторних робіт з курсу “Об'єктно-орієнтоване програмування” для студентів спеціальності “Інтелектуальні системи прийняття рішень” денної та заочної форм навчання / Укл. Баклан І.В., Степанкова Г.А. - Київ: НАУ, 2008. - 24 с.

Затверджено на засіданні Ради факультету комп'ютерних наук від 29 серпня 2008 р.

Лабораторная работа 1: Объектно-ориентированные концепции

Вопросы

С помощью API-документации для Java 2 попробуйте ответить на следующие вопросы:

1. Апплет ClickMe использует `Color.red` для установки красного цвета при рисовании. Какие другие цвета вы можете задать по именам аналогичным способом?
2. Как вы укажете другие цвета, например фиолетовый (`purple`)?

Упражнения

Теперь, используя то, что вы узнали из документации `Javadocs`, выполните следующие модификации для апплета **ClickMe**. Для компиляции программы вам понадобится файл **Spot.java**.

-
1. Измените апплет так, чтобы вместо красного пятна рисовать зеленый квадрат.
 2. Измените апплет так, чтобы вместо красного пятна рисовать ваше имя фиолетовым цветом.

Приложения

1. Полный код для апплета ClickMe показан ниже. Код, который участвует в обработке событий мыши, выделен красным цветом:

```
import java.applet.Applet;  
import java.awt.*;  
import java.awt.event.*;  
  
public class ClickMe extends Applet  
implements MouseListener {  
    private Spot spot = null;
```

```

private static final int RADIUS = 7;

public void init() {
    addMouseListener(this);
}

public void paint(Graphics g) {
    // рисование черной рамки и белого
фона
    g.setColor(Color.white);
    g.fillRect(0, 0, getSize().width -
1, getSize().height - 1);
    g.setColor(Color.black);
    g.drawRect(0, 0, getSize().width -
1, getSize().height - 1);

    // рисование пятна
    g.setColor(Color.red);
    if (spot != null) {
        g.fillOval(spot.x - RADIUS,
            spot.y - RADIUS,
            RADIUS * 2, RADIUS *
2);
    }
}

public void mousePressed(MouseEvent
event) {
    if (spot == null) {
        spot = new Spot(RADIUS);
    }
    spot.x = event.getX();
    spot.y = event.getY();
    repaint();
}

public void mouseClicked(MouseEvent
event) {}

public void mouseReleased(MouseEvent
event) {}

public void mouseEntered(MouseEvent
event) {}

public void mouseExited(MouseEvent
event) {}
}

```

2. Код программы Spot.java

```
public class Spot {
    public int size;
    public int x, y;

    public Spot(int intSize) {
        size = intSize;
        x = -1;
        y = -1;
    }
}
```

Лабораторная работа 2: Переменные

Вопросы

1. Что из показанного ниже является допустимым для имен переменных?

```
int
anInt
i
i1
1
thing1
lthing
ONE-HUNDRED
ONE_HUNDRED
something2do
```

2. Ответьте на следующие вопросы о программе **BasicsDemo**.

1. Какое имя у каждой переменной, объявленной в программе? Помните, что параметры метода - тоже переменные.
2. Каков тип данных у каждой переменной?
3. Какова сфера каждой переменной?

Упражнения

1. Модифицируйте программу **MaxVariablesDemo** так, чтобы переменная `aBoolean` имела другое значение.
2. Перепишите программу **MaxVariablesDemo** для показа минимального значения каждого целого типа данных. Вы можете догадаться, какие будут имена переменных, или можете посмотреть их в API-документации.
3. Вы можете угадать имя метода в классе **Character**, который сможете использовать вместо `isUpperCase` для определения капитализации символов (преобразования букв в заглавные)? Модифицируйте программу **MaxVariablesDemo** для использования этого метода вместо `isUpperCase`.

Приложения

1. Полный код **BasicsDemo**:

```

public class BasicsDemo {
    public static void main(String[] args)
    {
        int sum = 0;
        for (int current = 1; current <=
10; current++) {
            sum += current;
        }
        System.out.println("Sum = " + sum);
    }
}

```

```

public class MaxVariablesDemo {
    public static void main(String args[])
    {

        // integers
        byte largestByte = Byte.MAX_VALUE;
        short largestShort =
Short.MAX_VALUE;
        int largestInteger =
Integer.MAX_VALUE;
        long largestLong = Long.MAX_VALUE;

        // real numbers
        float largestFloat =
Float.MAX_VALUE;
        double largestDouble =
Double.MAX_VALUE;

        // other primitive types
        char aChar = 'S';
        boolean aBoolean = true;

        // display them all
        System.out.println("The largest
byte value is " + largestByte);
        System.out.println("The largest
short value is " + largestShort);
        System.out.println("The largest
integer value is " + largestInteger);
        System.out.println("The largest
long value is " + largestLong);

```

```

        System.out.println("The largest
float value is " + largestFloat);
        System.out.println("The largest
double value is " + largestDouble);

        if (Character.isUpperCase(aChar)) {
            System.out.println("The
character " + aChar + " is upper case.");
        } else {
            System.out.println("The
character " + aChar + " is lower case.");
        }
        System.out.println("The value of
aBoolean is " + aBoolean);
    }
}

```

2. Полный код **MaxVariablesDemo**:

```

public class MaxVariablesDemo {
    public static void main(String args[])
    {

        // integers
        byte largestByte = Byte.MAX_VALUE;
        short largestShort =
Short.MAX_VALUE;
        int largestInteger =
Integer.MAX_VALUE;
        long largestLong = Long.MAX_VALUE;

        // real numbers
        float largestFloat =
Float.MAX_VALUE;
        double largestDouble =
Double.MAX_VALUE;

        // other primitive types
        char aChar = 'S';
        boolean aBoolean = true;

        // display them all
        System.out.println("The largest
byte value is " + largestByte);
        System.out.println("The largest

```

```
short value is " + largestShort);
    System.out.println("The largest
integer value is " + largestInteger);
    System.out.println("The largest
long value is " + largestLong);

    System.out.println("The largest
float value is " + largestFloat);
    System.out.println("The largest
double value is " + largestDouble);

    if (Character.isUpperCase(aChar)) {
        System.out.println("The
character " + aChar + " is upper case.");
    } else {
        System.out.println("The
character " + aChar + " is lower case.");
    }
    System.out.println("The value of
aBoolean is " + aBoolean);
}
}
```

Лабораторная работа 3: Операторы

Вопросы

1. Рассмотрим следующий фрагмент кода:

```
arrayOfInts[j] > arrayOfInts[j+1]
```

Какие операторы содержит код?

2. Рассмотрим следующий фрагмент кода:

```
int i = 10;  
int n = i++%5;
```

- a. Какие значения будут иметь *i* и *n* после выполнения кода?
 - b. Какие окончательные значения будут иметь *i* и *n*, если вместо постфиксного оператора увеличения (*i++*) вы будете использовать префиксную форму (*++i*)?
3. Каким будет значение *i* после выполнения следующего фрагмента кода?

```
int i = 8;  
i >>=2;
```

4. Каким будет значение *i* после выполнения следующего фрагмента кода?

```
int i = 17;  
i >>=1;
```

Упражнения

1. Напишите программу, которая проверяет, является ли число с плавающей точкой нулем. (Совет: Вам не следует использовать оператор точного равенства `==` для чисел с плавающей точкой, потому что эти числа по своей природе таковы, что их трудно сравнивать. Вместо этого проверьте, достаточно ли близко в окрестности нуля расположено число).
2. Напишите программу, которая вычисляет количество долларов США, эквивалентное указанному количеству французских франков. Для обмена примите курс 6.85062 франка за доллар.
3. Напишите программу, которая использует биты в одном целом числе для представления данных `true/false`, показанных на следующей иллюстрации:



4. Включите в программу переменную с именем `status` и заставьте программу распечатывать состояние - значение `status`. Например, если `status` равняется 1 (установлен только бит 0), то программа должна выдавать нечто вроде:

Ready to receive requests

- a. Покажите ваш код.
- b. Какая будет выдача, если `status` равняется 8?
- c. Какая будет выдача, если `status` равняется 7?

Лабораторная работа 4: Выражения, инструкции и блоки

Вопросы

1. Какой тип будут иметь следующие выражения, если для `i` подразумевается тип `int`?

`i > 0`

`i = 0`

`i++`

`(float)i`

`i == 0`

`"aString" + i`

2. Рассмотрим следующее выражение:

`i--%5>0`

a. Какой будет результат выражения, если считать `i` первоначально равным 10?

b. Модифицируйте выражение таким образом, чтобы оно имело такой же результат, но было проще для чтения программистами.

Упражнения

1. Напишите программу для подтверждения ваших ответов на вопросы 2a и 2b.

Лабораторная работа 5: Управление ходом исполнения

Вопросы

1. Посмотрите программу **SortDemo**. Какие в ней есть инструкции, управляющие ходом выполнения?
2. Что неправильного в следующем фрагменте кода:

```
3. if (i = 1) {  
    /* do something */  
}
```
4. Посмотрите на программу **WhileDemo** и на программу **DoWhileDemo**. Какую выдачу будут давать эти программы, если вы измените в каждой из программ строковое значение `copyFromMe` string на `golly gee. this is fun`. Поясните свой прогноз для каждой из программ.

Упражнения

1. Рассмотрим следующий фрагмент кода.

```
if (aNumber >= 0)  
    if (aNumber == 0)  
        System.out.println("first string");  
    else System.out.println("second string");  
        System.out.println("third string");
```

 - a. Какую выдачу вы ожидаете от этого кода, если `aNumber` равняется 3?
 - b. Напишите тестовую программу, содержащую фрагмент кода, и сделайте `aNumber` 3. Какая выдача будет получена из программы? Это совпадает с вашим прогнозом? Поясните, почему выдача получилась такой. Другими словами, что управляет ходом выполнения для этого фрагмента кода?
 - c. Используя только пробелы и окончания строк, переформатируйте фрагмент кода, чтобы сделать ход управления проще для понимания.
 - d. Используйте скобки `{` и `}` для улучшения ясности кода и уменьшения возможных ошибок при будущих модификациях кода.

Лабораторная работа 6: Создание и применение объектов

Вопросы

1. Что неправильно в следующей программе?

```
public class SomethingIsWrong {
    public static void main(String[]
args) {
        Rectangle myRect;
        myRect.width = 40;
        myRect.height = 50;
        System.out.println("myRect's
area is " + myRect.area());
    }
}
```

2. Показанный ниже код создает один объект Point и один объект Rectangle. Сколько ссылок к этим объектам существует после того, как код будет выполнен? Будет ли какой-нибудь из объектов кандидатом для "сборки мусора"?

```
...
    Point point = new Point(2,4);
    Rectangle rectangle = new
Rectangle(point, 20, 20);
    point = null;
    ...
```

3. Как программа уничтожает объект, который она создает?

Упражнения

1. Исправьте программу по имени **SomethingIsWrong**, показанную в Вопросе 1.

2. Имея класс с названием NumberHolder, напишите некоторый код, который создает экземпляр из этого класса, инициализирует его две переменных (member variables), и затем покажет значение каждой из переменных.

```
public class NumberHolder {
    public int anInt;
    public float aFloat;
}
```

Лабораторная работа 7: Основы работы со строками

Вопросы

1. Какова первоначальная мощность (capacity) следующего строкового буфера?

```
StringBuffer sb = new  
StringBuffer("Able was I ere I saw  
Elba.");
```

2. Рассмотрим следующую строку:

```
String hannah = "Did Hannah see bees?  
Hannah did.";
```

- a. Какое значение покажет выражение `hannah.length()`?
- b. Какое значение вернет вызов метода `hannah.charAt(12)`?
- c. Напишите выражение, которое ссылается на букву в строке, к которой ссылаются по имени `hannah`.

3. Какой длины строка возвращается следующим выражением? Какая это строка?

```
"Was it a car or a cat I  
saw?".substring(9, 12)
```

4. В следующей программе с названием **ComputeResult** каким будет значение `result` после исполнения каждой пронумерованной строки?

```
public class ComputeResult {  
    public static void main(String[]  
args) {  
        String original = "software";  
        StringBuffer result = new  
StringBuffer("hi");  
        int index =  
original.indexOf('a');  
  
        /*1*/ result.setCharAt(0,  
original.charAt(0));  
        /*2*/ result.setCharAt(1,  
original.charAt(original.length()-1));
```

```

    /*3*/    result.insert(1,
original.charAt(4));
    /*4*/
result.append(original.substring(1,4));
    /*5*/    result.insert(3,
(original.substring(index, index+2) + "
"));

        System.out.println(result);
    }
}

```

Упражнения

1. Покажите два способа конкатенации следующих двух строк, чтобы получить строку "Hi, mom.":

```
String hi = "Hi, ";
String mom = "mom.";
```
2. Напишите программу, которая из вашего полного имени вычисляет и показывает ваши инициалы.
3. Анаграмма - это слово или фраза, созданная перестановкой букв другого слова или фразы. Например, слово "parliament" является анаграммой фразы "partial men," а слово "software" является анаграммой фразы "swear oft." Напишите программу, которая определяет, является ли строка анаграммой другой строки. Программа должна игнорировать пробелы и знаки пунктуации.

Лабораторная работа 8: Основы работы с числами

Вопросы

1. Пользуясь API-документацией, найдите ответы на следующие вопросы:

a. Какой метод `Integer` вы можете использовать для преобразования `int` в строку, которая представляет число в 16-ричном виде?

Например, какой метод преобразует целое 65 в строку "41"?

b. Какие два метода `Integer` для преобразования строки, представленной по основанию 5, в эквивалентное `int`? Например, как вы преобразуете строку "230" в целое значение 65? Покажите код, который вы будете использовать для выполнения этой задачи.

c. Какой метод `Double` вы можете использовать, чтобы определить, будет ли число с плавающей точкой (floating-point number) иметь специальное значение `Not a Number (NaN)`?

2. Каким будет значение следующего выражения, и почему?

```
new Integer(1).equals(new Long(1))
```

Упражнения

1. Измените **MaxVariablesDemo** так, чтобы показать минимальные значения вместо максимальных. Вы можете удалить весь код, связанный с переменными `aChar` и `aBoolean`. Какой будет выдача?

2. Создайте программу, которая читает из командной строки неуказанное количество аргументов и складывает их вместе. Например, пусть вы ввели следующее:

```
java Adder 1 3 2 10
```

Программа должна показать 16 и затем выполнить выход. Программа должна показать сообщение об ошибке, если пользователь ввел только один аргумент. Вы можете создать свою программу на **ValueOfDemo.java**.

3. Создайте программу, аналогичную предыдущей, но со следующими отличиями:

- Вместо чтения целочисленных аргументов она читает аргументы с плавающей точкой.
- Она показывает сумму аргументов, используя ровно два разряда правее десятичной точки.

Например, пусть вы ввели следующее:

```
java FPAdder 1 1e2 3.0 4.754
```

Программа должна показать 108.75. В зависимости от ваших региональных установок (locale), в позиции десятичной дроби может быть запятая (,) вместо точки (.).

Лабораторная работа 9: Основы работы с массивами

Вопросы

1. Чему равен индекс для Brighton в следующем массиве?

```
String[] skiResorts = {  
    "Whistler Blackcomb", "Squaw  
    Valley", "Brighton",  
    "Snowmass", "Sun Valley", "Taos"  
};
```
2. Напишите выражение, которое обращается к строке Brighton внутри массива.
3. Чему равно значение выражения `skiResorts.length`?
4. Какой индекс у последнего элемента в массиве?
5. Какое значение имеет выражение `skiResorts[4]`?

Упражнения

1. Следующая программа WhatHappens содержит ошибку. Найдите и исправьте ее.

```
//  
// This program compiles but won't run  
// successfully.  
//  
public class WhatHappens {  
    public static void main(String[]  
args) {  
        StringBuffer[] stringBufferers =  
new StringBuffer[10];  
  
        for (int i = 0; i <  
stringBufferers.length; i++) {  
            stringBufferers[i].append("St  
ringBuffer at index " + i);  
        }  
    }  
}
```

Лабораторная работа 10: Управление наследованием

Questions

1. Рассмотрим следующие два класса:

```
public class ClassA {
    public void methodOne(int i) {
    }
    public void methodTwo(int i) {
    }
    public static void methodThree(int i) {
    }
    public static void methodFour(int i) {
    }
}
```

```
public class ClassB extends ClassA {
    public static void methodOne(int i) {
    }
    public void methodTwo(int i) {
    }
    public void methodThree(int i) {
    }
    public static void methodFour(int i) {
    }
}
```

- Какой из методов переопределяет метод в суперклассе?
- Какой из методов скрывает метод в суперклассе?
- Что делают другие методы?

2. Рассмотрим классы Card, Deck и DisplayDeck, которые вы написали в упражнениях предыдущего раздела. Какие из методов Object должен переопределять каждый из этих классов?

Упражнения

- Напишите реализацию для методов, о которых вы отвечали в вопросе 2.
- Напишите абстрактный класс. Напишите как минимум два его не-абстрактных subclasses.

Лабораторная работа 11: Реализация вложенных классов

Вопросы

1. Сопоставьте ситуацию в первой колонке с соответствующим типом вложенного класса во второй колонке.

a. Единственными пользователями этого вложенного класса будут экземпляры включающего класса или экземпляры суб-классов включающего класса.

1. anonymous inner class

b. Этот вложенный класс может использовать кто угодно.

2. protected inner class

c. Использовать это вложенный класс нужно только экземплярам объявляющего класса, и конкретный экземпляр может использовать его несколько раз.

3. public static nested class

d. Этот небольшой вложенный класс используется лишь один раз, для создания объекта, который реализует интерфейс.

4. protected static nested class

e. Этот вложенный класс имеет информацию о включающем его классе (но не экземплярах включающего класса) и используется только включающим его классом и, возможно, суб-классами этого класса.

5. private static nested class

f. Ситуация, аналогичная предыдущей (вариант e), но не предназначена для использования суб-классами.

6. private inner class

2. Программа Problem.java не компилируется. Что вам нужно сделать, чтобы она компилировалась? Почему?

3. Воспользуйтесь 1.3 API-документацией для класса Vox (в библиотеке javax.swing) для помощи при ответе на следующие вопросы.

a. Какой статический вложенный (static nested)

- класс определяет Box?
- b. Какой внутренний (inner) класс определяет Box?
 - c. Что является суперклассом для внутреннего (inner) класса Box?
 - d. Какие внутренние (nested) классы класса Box вы можете использовать из любого класса?
 - e. Как вы создадите экземпляр класса Box's Filler?

Упражнения

1. Сначала получите исходный файл InnerClassDemo.java

a. Скомпилируйте и запустите InnerClassDemo.java.

b. Сделайте копию InnerClassDemo. Добавьте в нее внутренний (inner) класс с именем MyActionListener, который реализует интерфейс ActionListener. Интерфейс ActionListener определяет один метод. Поместите следующий код в вашу реализацию метода: quit();

Удалите две косые черты (//) в начале следующей строки кода:

```
//button.addActionListener(new  
MyActionListener());
```

Теперь скомпилируйте и запустите программу. Чем отличается поведение этой версии и предыдущей версии InnerClassDemo?

c. Сделайте копию программы, которую вы создали в упражнении 1b. Измените вашу реализацию ActionListener, чтобы это был анонимный внутренний (inner) класс. (Совет: Программа имеет другой анонимный внутренний класс WindowAdapter, который вы можете использовать как справку для помощи по синтаксису).

2. Возьмите файл Class1.java.

a. Скомпилируйте и запустите Class1. Что получится на выдачев?

b. Создайте файл с именем Class2.java, который определяет суб-классы класса Class1 и его внутреннего класса InnerClass1. (Вызывает суб-классы Class2 и InnerClass2 соответственно). InnerClass2 должен переопределить метод getAnotherString для возврата "InnerClass2 version of getAnotherString invoked". Class2 должен определить один конструктор и один метод:

- Конструктор без аргументов должен инициализировать унаследованную переменную экземпляра (instance variable) ic, чтобы она была в экземпляре InnerClass2
- Метод main, который создает экземпляр Class2 и вызывает displayStrings для этого экземпляра

Какой будет вызача при запуске Class2?

Лабораторная работа 12: Создание интерфейсов

Вопросы

1. Какие методы должен реализовать класс, реализующий интерфейс `java.util.Iterator`?
2. Что неправильно в следующем интерфейсе?

```
public interface SomethingIsWrong {  
    public void aMethod(int aValue){  
        System.out.println("Hi Mom");  
    }  
}
```

3. Исправьте интерфейс из Вопроса 2.
4. Правильный ли следующий интерфейс?

```
public interface Marker {  
}
```

Упражнения

1. Напишите класс, который реализует интерфейс `Iterator`, находящийся в библиотеке `java.util`. Упорядоченными данными для этого упражнения являются 13 карт одной масти из колоды карт. Первый вызов `next` возвращает 2, следующий вызов возвращает следующую по старшинству карту 3, и т.д., до Туза (Ace). Напишите небольшой метод `main` для проверки вашего класса.
2. Предположим, вам нужно написать сервер времени, который периодически извещает своих клиентов о текущей дате и времени. Напишите интерфейс, который сервер сможет использовать для предложения конкретного протокола своим клиентам.

Лабораторная работа 13: Создание и применение библиотек

Вопросы

1. Допустим, вам нужно написать некоторые классы. Вы с опозданием решили, что они должны быть разбиты на три библиотеки, перечисленные ниже в таблице. Более того, предположим, что в данный момент классы находятся в библиотеке по умолчанию - default package (у них нет инструкций package).

Имя Библиотеки	Имя Класса
mygame.server	Server
mygame.shared	Utilities
mygame.client	Client

а. Какую строку кода вам нужно добавить в каждый исходный файл, чтобы каждый класс попал в нужную библиотеку?

б. Для привязки к структуре директорий вам нужно создать в директории вашей разработки некоторые поддиректории и поместить исходные файлы в правильные поддиректории. Какие поддиректории вам нужно создать? В какие поддиректории нужно поместить каждый исходный файл?

в. Вы считаете, что вам нужно выполнить какие-то другие изменения для исходных файлов, чтобы сделать их компиляцию корректной. Если да, то какие?

Упражнения

1. Выгрузите себе три исходных файла, показанные на этой странице.

- Client
- Server
- Utilities

а. Реализуйте изменения, которые предполагались в вопросе 1, используя полученные исходные файлы.

б. Скомпилируйте исправленные исходные файлы. (Совет: Если вы вызываете компилятор из командной строки (в противовес применению builder), - вызовите компилятор из директории, которая содержит только что созданную вами директорию тугаме).