



المملكة العربية السعودية
المؤسسة العامة للتدريب التقني والمهني
الإدارة العامة لتصميم وتطوير المناهج

```
Project1 - frmBmi (Code)
cmdCalc
Private Sub cmdCalc
    txtDisplay.Text
End Sub

If Len(rsMsg) = 0 Then
    Screen.MousePointer =
    frmMDI.stsStatusBar.Pane1/
Else
    If rPauseFlag Then
        frmMDI.stsStatusBar.Pane1/
    Else
        frmMDI.stsStatusBar.Pane1/
    End Sub
End Sub

<script language="JavaScript">

function animateAnchor() {
    var el=event.srcElement;
    if ("A"==el.tagName) { // Initialize effect
        if (null==el.effect) el.effect = "high"
    }
}
```

تخصص برمجيات

برمجة الحاسب

(عملي)

154 حاب

مقدمة

الحمد لله وحده، والصلاة والسلام على من لا نبي بعده، محمد وعلى آله وصحبه، وبعد:

تسعى المؤسسة العامة للتدريب التقني والمهني لتأهيل الكوادر الوطنية المدربة القادرة على شغل الوظائف التقنية والفنية والمهنية المتوفرة في سوق العمل، ويأتي هذا الاهتمام نتيجة للتوجهات السديدة من لدن قادة هذا الوطن التي تصب في مجملها نحو إيجاد وطن متكامل يعتمد ذاتياً على موارده وعلى قوة شبابه المسلح بالعلم والإيمان من أجل الاستمرار قدماً في دفع عجلة التقدم التتموي: لتصل بعون الله تعالى لمصاف الدول المتقدمة صناعياً.

وقد خطت الإدارة العامة لتصميم وتطوير المناهج خطوة إيجابية تتفق مع التجارب الدولية المتقدمة في بناء البرامج التدريبية، وفق أساليب علمية حديثة تحاكي متطلبات سوق العمل بكافة تخصصاته لتلبي متطلباته، وقد تمثلت هذه الخطوة في مشروع إعداد المعايير المهنية الوطنية الذي يمثل الركيزة الأساسية في بناء البرامج التدريبية، إذ تعتمد المعايير في بنائها على تشكيل لجان تخصصية تمثل سوق العمل والمؤسسة العامة للتدريب التقني والمهني بحيث تتوافق الرؤية العلمية مع الواقع العملي الذي تفرضه متطلبات سوق العمل، لتخرج هذه اللجان في النهاية بنظرة متكاملة لبرنامج تدريبي أكثر التصاقاً بسوق العمل، وأكثر واقعية في تحقيق متطلباته الأساسية.

وتتناول هذه الحقيبة التدريبية " برمجة الحاسب (عملي) " لمتدربي تخصص " برمجيات " لمعاهد التدريب العسكري المهني موضوعات حيوية تتناول كيفية اكتساب المهارات اللازمة لهذا التخصص. والإدارة العامة لتصميم وتطوير المناهج وهي تضع بين يديك هذه الحقيبة التدريبية تأمل من الله عز وجل أن تسهم بشكل مباشر في تأصيل المهارات الضرورية اللازمة، بأسلوب مبسط يخلو من التعقيد، وبالاستعانة بالتطبيقات والأشكال التي تدعم عملية اكتساب هذه المهارات.

والله نسأل أن يوفق القائمين على إعدادها والمستفيدين منها لما يحبه ويرضاه: إنه سميع مجيب

الدعاء.

الإدارة العامة لتصميم وتطوير المناهج

برمجة الحاسب (عملي)

المصفوفات

التدريب الأول

(المصفوفات ذات البعد الواحد)

الأهداف:

- 1- التعرف على كيفية تعريف المصفوفات ذات البعد الواحد.
- 2- التعرف على كيفية الوصول للعناصر المخزنة في المصفوفة عن طريق رقم الموقع (Index).
- 3- التعرف على كيفية طباعة جميع محتويات المصفوفة عن طريق إحدى جمل الدوران.

التمارين:

تمرين رقم (1)

// Array1.java

```

1. public class Array1{
2. public static void main(String args[]){
3. int a[] = new int[5];
4. a[1] = 34;
5. a[4] = 67;
6. for(int i=0; i<5; i++)
7. System.out.print(a[i]+ " ");
8. }
9. }

```

طباعة جميع العناصر المخزنة في
المصفوفة a كما يلي:

0 34 0 0 67

يتم تخزين العدد 34 في الموقع (Index) رقم 1
والعدد 67 في الموقع (Index) رقم 4 في
المصفوفة a.

تعريف المصفوفة a من نوع
int، وحجز 5 مواقع لتخزين
أعداد صحيحة فيها.

0	0
1	0
2	0
3	0
4	0

عند التعريف يتم تخزين
القيمة الافتراضية (Default
Value) للنوع int وهي
الصفري.

تمرين رقم (2)

// Array.java

```

1. public class Array2{
2.   public static void main(String args[]){
3.     int a[] = {5, 6, 12, 19, 14};
4.     System.out.println(a[3]);
5.     for(int i=0; i<5; i++)
6.       System.out.println(a[i]);
7.   }
8. }

```

طباعة جميع العناصر
المخزنة في المصفوفة a
كما يلي:
5
6
12
19
14

طباعة العنصر المخزن في
الموقع 3 في المصفوفة a.
نتج الجملة هو طباعة
الرقم 19.

تعريف المصفوفة a من
نوع int، وتخزين قيم
ابتدائية فيها.

0	5
1	6
2	12
3	19
4	14

تمرين رقم (3)

// Array3.java

```

1. import javax.swing.*;
2. public class Array3{
3.   public static void main(String args[]){
4.     int marks[] = new int[10];
5.     int sum=0;
6.     String s, output="";
7.     double average;
8.     for(int i=0; i<10; i++){
9.       s=JOptionPane.showInputDialog("Enter the mark number " + i + " :");
10.      marks[i]=Integer.parseInt(s);
11.      sum=sum+marks[i];
12.    }
13.    output=output + "\n\nThe 10 marks are: ";
14.    for(int i=0; i<10; i++)
15.      output=output + marks[i] + " "; output=output + "\n\nThe sum of the 10 marks
16.      is: " + sum + "\n\n";
17.    average=(double)sum/10;
18.    output=output + "The average of the 10 marks is: " + average + "\n\n";

```

```
18. JOptionPane.showMessageDialog(null, output);
19. System.exit(0);
20. }
21. }
```

يقوم هذا البرنامج بطلب إدخال 10 درجات عن طريق لوحة المفاتيح ويخزنها في مصفوفة اسمها (marks)، حيث يقوم بطباعة الدرجات العشرة، وحساب وطباعة مجموع (sum) ومتوسط (average) هذه الدرجات.

الواجبات:

واجب رقم (1)

اكتب برنامج يقوم بتعريف مصفوفة اسمها numbers ويعطيها القيم الابتدائية التالية: (3, 12, 6, 2, 19, 44, 12)، بحيث يقوم البرنامج بطباعة الأعداد المخزنة في المواقع الفردية في المصفوفة.

واجب رقم (2)

اكتب برنامج لإدخال 8 أعداد وتخزينها في مصفوفة بحيث يقوم البرنامج بطباعة ما يلي:

1. جميع الأعداد المخزنة في المصفوفة.

2. مجموع الأعداد السالبة.

3. عدد الأعداد الزوجية.

4. معدل جميع الأعداد.

واجب رقم (3)

اكتب برنامج لإدخال 10 أعداد وتخزينها في مصفوفة بحيث يقوم البرنامج بطباعة ما يلي:

1- الأعداد المخزنة في المصفوفة بشكل عكسي، أي في بداية الطباعة يقوم البرنامج بطباعة آخر

عدد في المصفوفة ثم يطبع العدد الذي قبله وفي النهاية يطبع أول عدد في المصفوفة.

2- أكبر عدد مخزن في المصفوفة وموقعه (Index) في المصفوفة.

3- أصغر عدد مخزن في المصفوفة وموقعه (Index) في المصفوفة.

التدريب الثاني

(ترتيب المصفوفات)

الأهداف:

1- التعرف على كيفية ترتيب محتويات المصفوفات ترتيباً تصاعدياً وترتيباً تنازلياً.

التمارين:

تمرين رقم (1)

```
// array1.java
1. public class array1{
2. public static void main(String[] args) {
3. int numbers[]={ 21, 24, 43, 54, 15, 63, 27, 84,
4. 29, 10, 17, 42, 28, 33, 41, 67};
5. int temp;
6. System.out.println("\nThe contents of array numbers are:");
7. for(int i=0; i<numbers.length; i++)
8. System.out.print(numbers[i]+" ");
9. for(int i=1; i<numbers.length; i++)
10. for(int j=0; j<numbers.length-1; j++)
11. if(numbers[j]>numbers[j+1]){
12. temp=numbers[j];
13. numbers[j]=numbers[j+1];
14. numbers[j+1]=temp;
15. }
16. System.out.println("\n\nThe contents of array numbers in ascending order
17. are:");
18. for(int i=0; i<numbers.length; i++)
19. System.out.print(numbers[i]+" ");
20. System.out.println("\n");
21. }
22. }
```


يقوم هذا البرنامج بترتيب محتويات المصفوفة numbers ترتيباً تصاعدياً وذلك بعد تنفيذ الحمل الموجودة في السطر (8 - 14). والشكل التالي يبين مخرجات هذا البرنامج.

```

C:\Program Files\Xinox Software\JCreatorV3 LE\GE2001.exe
The contents of array numbers are:
21 24 43 54 15 63 27 84 29 10 17 42 28 33 41 67

The contents of array numbers in ascending order are:
10 15 17 21 24 27 28 29 33 41 42 43 54 63 67 84

Press any key to continue...

```

تمرين رقم (2)

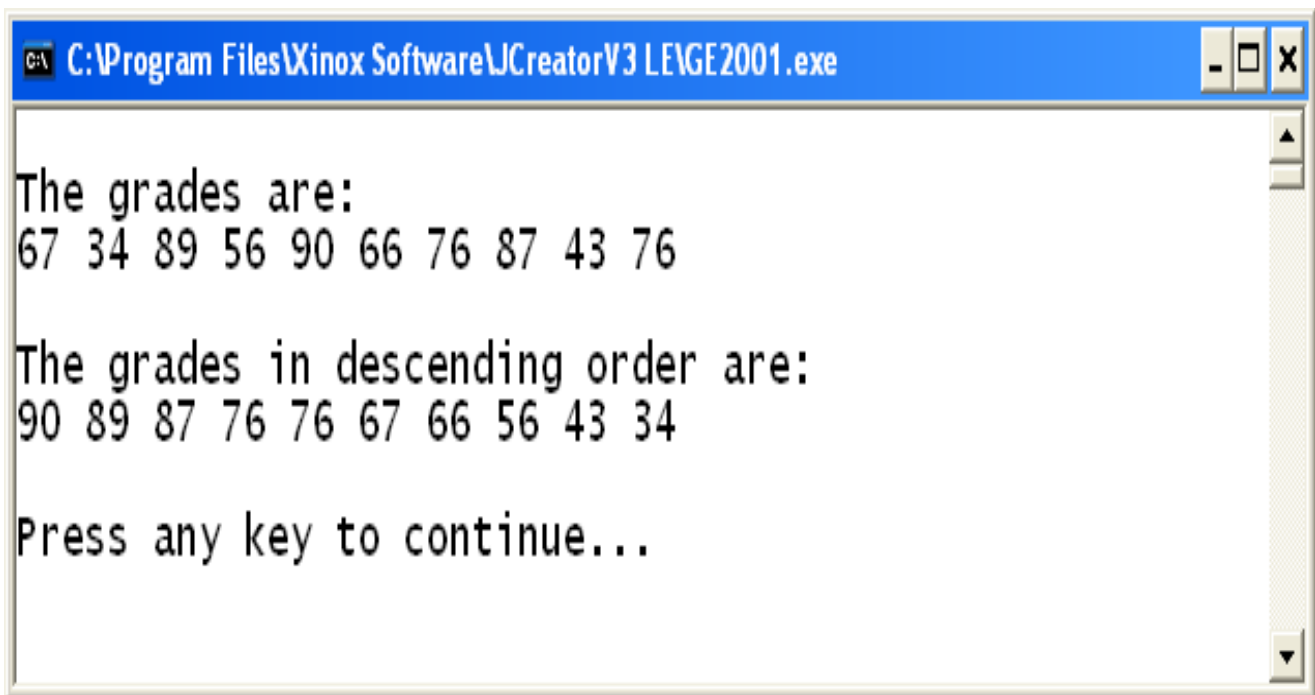
```

// array2.java
1. import javax.swing.*;
2. public class array2{
3.     public static void main(String[] args) {
4.         int grades[]= new int[10];
5.         int temp;
6.         String s;
7.         for(int i=0; i<grades.length; i++){
8.             s=JOptionPane.showInputDialog("Enter a grade:");
9.             grades[i]=Integer.parseInt(s);
10.        }
11.        System.out.println("\nThe grades are:");
12.        for(int i=0; i<grades.length; i++)
13.            System.out.print(grades[i]+" ");
14.        for(int i=1; i<grades.length; i++)
15.            for(int j=0; j<grades.length-1; j++)
16.                if(grades[j]<grades[j+1]){
17.                    temp=grades[j];
18.                    grades[j]=grades[j+1];
19.                    grades[j+1]=temp;

```

```
20. }
21. System.out.println("\n\nThe grades in descending order:");
22. for(int i=0; i<grades.length; i++)
23. System.out.print(grades[i]+" ");
24. System.out.println("\n");
25. System.exit(0);
26. }
27. }
```

في هذا البرنامج يقوم المستخدم بإدخال 10 درجات، حيث يقوم البرنامج بتخزين هذه الدرجات في المصفوفة grades ويطبعها بترتيبها الأصلي وبعد ذلك يقوم بترتيبها ترتيباً تنازلياً ويعيد طباعة محتويات المصفوفة لتكون الدرجات المخرجة مرتبة تنازلياً. والشكل التالي يبين مخرجات هذا البرنامج بعد أن يقوم المستخدم بإدخال الدرجات التالية عن طريق لوحة المفاتيح: 67، 34، 89، 56، 90، 66، 76، 87، 43، 76.



```
C:\Program Files\Xinox Software\JCreatorV3 LE\GE2001.exe

The grades are:
67 34 89 56 90 66 76 87 43 76

The grades in descending order are:
90 89 87 76 76 67 66 56 43 34

Press any key to continue...
```

الواجبات:**واجب رقم (1)**

اكتب برنامج لإدخال رقم الطالب، اسم الطالب، ومعدل الطالب وذلك لعشرة طلاب، حيث يقوم البرنامج بطباعة هذه المعلومات بالترتيب التصاعدي اعتماداً على رقم الطالب، وطباعتها مرة أخرى بالترتيب التنازلي اعتماداً على معدل الطالب.

واجب رقم (2)

اكتب برنامج لقراءة معدلات 10 طلاب وتخزين ما يكافئ هذه المعدلات من حروف في مصفوفة. ثم يقوم البرنامج بترتيب محتويات المصفوفة ترتيباً تنازلياً. اعتمد على ما يلي لتحويل المعدلات إلى أحرف:

- إذا كان المعدل أقل من أو يساوي 100 وأكبر من أو يساوي 90 يصبح A.
- إذا كان المعدل أقل من 90 وأكبر من أو يساوي 80 يصبح B.
- إذا كان المعدل أقل من 80 وأكبر من أو يساوي 70 يصبح C.
- إذا كان المعدل أقل من 70 وأكبر من أو يساوي 60 يصبح D.
- إذا كان المعدل أقل من 60 وأكبر من أو يساوي صفر يصبح F.
- إذا كان المعدل غير ذلك يقوم البرنامج بعرض رسالة خطأ. ويطلب إدخال معدل آخر.

التدريب الثالث

(طرق البحث في المصفوفات)

الأهداف:

- 1- التعرف على طريقة البحث الخطي (Linear Search) للبحث عن عنصر معين في المصفوفة.
- 2- التعرف على طريقة البحث الثنائي (Binary Search) للبحث عن عنصر معين في المصفوفة المرتبة.

التمارين:

تمرين رقم (1)

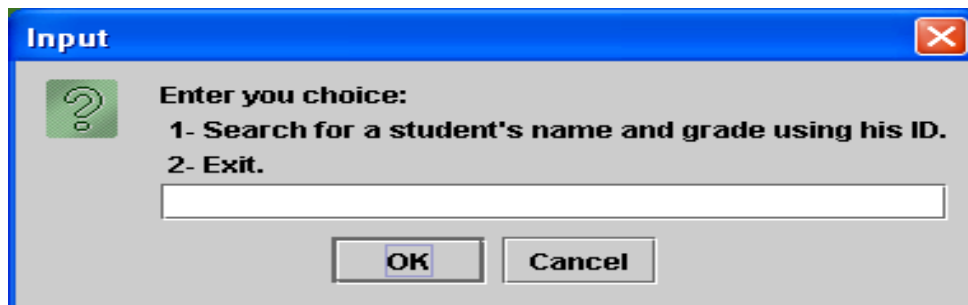
```
// array1.java
1. import javax.swing.*;
2. public class array1{
3.     public static void main(String[] args) {
4.         int grades[] = new int[10];
5.         String names[] = new String[10];
6.         int ids[]=new int[10];
7.         String name, s;
8.         int c, no;
9.         String out;
10.        for(int i=0; i<grades.length; i++){
11.            names[i]=JOptionPane.showInputDialog("Enter student's name:");
12.            s=JOptionPane.showInputDialog("Enter his grade:");
13.            grades[i]=Integer.parseInt(s);
14.            ids[i]=i+100;
15.        }
16.        do{
17.            out="There is no student with this ID";
18.            s=JOptionPane.showInputDialog("Enter you choice:\n 1- Search for a
                student's name and grade using his ID.\n 2- Exit.");
19.            c=Integer.parseInt(s);
20.            switch(c){
21.                case 1:
22.                    s=JOptionPane.showInputDialog("Enter the student's ID:");
23.                    no=Integer.parseInt(s);
```

```

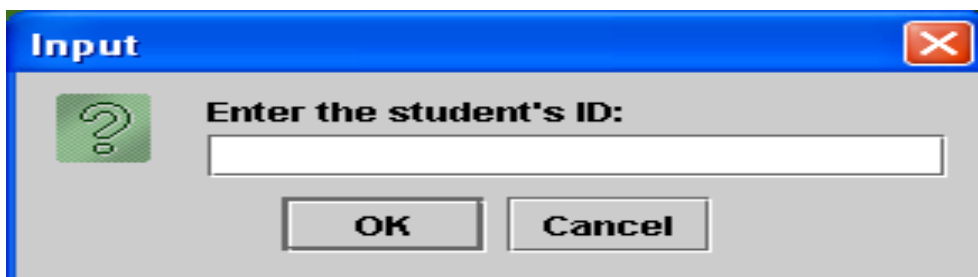
24.   for(int i=0; i<ids.length; i++)
25.       if(ids[i]==no)
26.           out="The student name is: " + names[i] + "\n\nThe student's grade is: " +
                grades[i];
27.       JOptionPane.showMessageDialog(null, out, "Results",
                JOptionPane.PLAIN_MESSAGE);
28.           break;
29.       case 2: System.exit(0);
30.   }
31. } while(c!=2);
32. System.exit(0);
33. }
34. }

```

في هذا البرنامج يتم إدخال أسماء، درجات وحساب أرقام لعشرة طلاب. بعد ذلك يقوم البرنامج بعرض قائمة اختيار يتم من خلالها اختيار عملية البحث عن اسم ودرجة طالب عن طريق إدخال رقم هذا الطالب. يتم البحث في هذا البرنامج باستخدام طريقة البحث الخطي. وتكون مخرجات البرنامج بعد إدخال أسماء ودرجات الطالب بالشكل التالي:



وعند إدخال الرقم 1 لتنفيذ الخيار الأول وهو البحث عن اسم ودرجة طالب باستخدام رقمه، يقوم البرنامج بطلب إدخال رقم الطالب المراد البحث عنه، وبالشكل التالي:



وعند إدخال الرقم 100، يقوم البرنامج بعرض اسم ودرجة الطالب صاحب هذا الرقم.

تمرين رقم (2)

```
// array2.java
1. import javax.swing.*;
2. public class array2{
3.     public static void main(String[] args) {
4.         int grades[] = new int[10];
5.         String names[] = new String[10];
6.         int ids[]=new int[10];
7.         String name, s;
8.         int c, no, low, high, middle;
9.         String out;
10.        for(int i=0; i<grades.length; i++){
11.            names[i]=JOptionPane.showInputDialog("Enter student's name:");
12.            s=JOptionPane.showInputDialog("Enter his grade:");
13.            grades[i]=Integer.parseInt(s);
14.            ids[i]=i+100;
15.        }
16.        do{
17.            high=grades.length-1;
18.            low=0;
19.            out="There is no student with this ID";
20.            s=JOptionPane.showInputDialog("Enter you choice:\n 1- Search for a
                student's name and grade using his ID.\n 2- Exit.");
21.            c=Integer.parseInt(s);
22.            switch(c){
23.                case 1:
24.                    s=JOptionPane.showInputDialog("Enter the student's ID:");
25.                    no=Integer.parseInt(s);
26.                    while(low <=high){
27.                        middle=(low+high)/2;
28.                        if(no==ids[middle]){
29.                            out="The student name is: " + names[middle] + "\nThe student's grade is:
                                    " + grades[middle];
30.                            break;
31.                        }
32.                        else if(no<ids[middle])
33.                            high=middle-1;
34.                        else
35.                            low=middle+1;
36.                    }
```

```
37. JOptionPane.showMessageDialog(null, out, "Results",
                                   JOptionPane.PLAIN_MESSAGE);
38. break;
39. case 2: System.exit(0);
40. }
41. } while(c!=2);
42. System.exit(0);
43. }
44. }
```

هذا البرنامج هو نفس البرنامج السابق لكن يتم البحث فيه باستخدام طريقة البحث الثنائي، لاحظ الأسطر (25 - 35).

الواجبات:**واجب رقم (1)**

اكتب برنامج لقراءة 15 رقم وتخزينها في مصفوفة ، بحيث يقوم البرنامج بالبحث عن عدد مرات تكرار الرقم X والذي يقوم المستخدم بإدخاله عن طريق لوحة المفاتيح.

واجب رقم (2)

اكتب برنامج يقارن بين طريقتي البحث الخطي والبحث الثنائي وذلك من خلال طباعة أرقام المواقع للعناصر التي تم مقارنتها مع الرقم المراد البحث عنه.

التدريب الرابع

(المصفوفات ذات البعدين)

الأهداف:

- 1- التعرف على كيفية تعريف وحجز المواقع للمصفوفات ذات البعدين.
- 2- التعرف على كيفية إعطاء القيم الابتدائية للمصفوفات ذات البعدين.
- 3- التعرف على كيفية الوصول لصف معين أو عمود معين داخل المصفوفات ذات البعدين.
- 4- التعرف على كيفية طباعة محتويات المصفوفات ذات البعدين على شكل جداول.

التمارين:

تمرين رقم (1)

```
// array1.java
1. import javax.swing.*;
2. public class array1{
3.     public static void main(String[] args) {
4.         String o="";
5.         int sum=0;
6.         int a[][]={{2, 4, 5},
7.                    {9, 3, 6},
8.                    {8, 4, 7},
9.                    {6, 3, 1}};
10.        for(int i=0; i<a.length; i++){
11.            for(int j=0; j<a[i].length; j++)
12.                o=o+a[i][j]+" ";
13.            o=o+"\n";
14.        }
15.        JOptionPane.showMessageDialog(null, o, "Results",
16.            JOptionPane.PLAIN_MESSAGE);
17.        o="";
18.        for(int i=0; i<a.length; i++)
19.            sum=sum+a[i][2];
20.        o=o+"The sumation of the third column elements: "+sum;
21.        JOptionPane.showMessageDialog(null, o, "Results",
22.            JOptionPane.PLAIN_MESSAGE);
23.        o="";
```

```

19. sum=0;
20.for(int i=0; i<a[1].length; i++)
21. sum=sum+a[1][i];
22.o=o+"The sumation of the second row elements: "+sum;
23.JOptionPane.showMessageDialog(null, o, "Results",
    JOptionPane.PLAIN_MESSAGE);
24.System.exit(0);
25.}
26.}

```

في هذا البرنامج تم تعريف مصفوفة ذات بعدين وإعطائها قيماً ابتدائية (السطر 6). وفي الأسطر (7-12) تم طباعة محتويات المصفوفة على شكل جدول. ومن خلال الأسطر (14-17) تم حساب طباعة مجموع العناصر المخزنة في العمود الثالث للمصفوفة a. بينما في السطري (20-21) تم حساب وطباعة مجموع العناصر المخزنة في الصف الثاني للمصفوفة a.

تمرين رقم (2)

```

// array2.java
1. import javax.swing.*;
2. public class array2{
3.     public static void main(String[] args) {
4.         int mult[][]=new int[7][7];
5.         for(int i=0; i<mult.length; i++)
6.             for(int j=0; j<mult[i].length; j++)
7.                 mult[i][j]=(i+1)*(j+1);
8.         System.out.println();
9.         for(int i=0; i<mult.length; i++)
10.            System.out.print("\t"+(i+1));
11.        System.out.println();
12.        System.out.print("\t");
13.        for(int i=0; i<mult.length; i++)
14.            System.out.print("-----");
15.        System.out.println();
16.        for(int i=0; i<mult.length; i++){
17.            System.out.print(" "+(i+1)+"\t");
18.            for(int j=0; j<mult[i].length; j++)
19.                System.out.print(mult[i][j)+"\t");

```

```

20. System.out.println();
21. }
22. System.out.println();
23. System.exit(0);
24. }
25. }

```

في السطر رقم (4) من هذا البرنامج تم تعريف مصفوفة ذات بعدين (7 × 7) وتم حجز مواقع لها، ومن خلال جملة دوران تم تخزين جدول ضرب السبعة في هذه المصفوفة (الأسطر 5 - 7). حيث يقوم البرنامج بطباعة جدول الضرب هذا والمخزن في المصفوفة .mult. والشكل التالي يبين مخرجات هذا البرنامج.

	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7
2	2	4	6	8	10	12	14
3	3	6	9	12	15	18	21
4	4	8	12	16	20	24	28
5	5	10	15	20	25	30	35
6	6	12	18	24	30	36	42
7	7	14	21	28	35	42	49

Press any key to continue...

الواجبات:**واجب رقم (1)**

اكتب برنامج لتعريف مصفوفة ذات بعدين اسمها D2D ، بحيث يقوم البرنامج بإعطائها القيم الابتدائية التالية عند التعريف، الصف الأول يحتوي على الأرقام التالية: (12.3 ، 34.6 ، 67.1)، والصف الثاني يحتوي على الأرقام التالية: (56.6 ، 67.3 ، 87.7)، بينما الصف الثالث يحتوي على الأرقام التالية: (23.9 ، 48.4 ، 61.8). كما يقوم البرنامج بطباعة ما يلي:

- محتويات الصف الثاني.
- مجموع الأرقام المخزنة في الصف الثالث مع الأرقام المخزنة في العمود الأول.
- معدل كل صف من الصفوف الثلاث.
- معدل كل عمود من الأعمدة الثلاث.

واجب رقم (2)

لديك خمسة طلاب ولكل طالب منهم ثلاث درجات، اكتب برنامج لقراءة درجات الطلاب الخمسة وتخزينها في مصفوفة ذات بعدين. بحيث يحسب البرنامج مجموع درجات كل طالب من الطلاب الخمسة وخرن هذه المجاميع في مصفوفة أخرى (ذات بعد واحد) ومن ثم يرتبها ترتيباً تنازلياً. بحيث يطبع البرنامج ما يلي:

- أعلى درجة من درجات الطلاب.
- أعلى مجموع من مجاميع درجات الطلاب.
- محتويات المصفوفة ذات البعدين على شكل جدول.
- محتويات المصفوفة ذات البعد الواحد.

برمجة الحاسب (عملي)

الطرق

التدريب الخامس

(الطرق الخاصة بالعمليات الحسابية)

الأهداف:

- 1- أن يتمكن الطالب من التعامل مع صنف العمليات الحسابية (Math class).
- 2- أن يتمكن الطالب من التعامل مع الطريقة الخاصة بخلق الأرقام العشوائية.

التمارين:

تمرين رقم (1)

```
// Math_Methods_1.java
1. import javax.swing.JOptionPane;
2. public class Math_Methods_1 {
3.     public static void main(String[] args) {
4.         double y,x,z;
5.         x=Integer.parseInt(
        JOptionPane.showInputDialog("Enter value for x"));
6.         z=Integer.parseInt(
        JOptionPane.showInputDialog("Enter value for z"));
7.         y=4 * Math.sqrt(5*x) + 3* Math.pow(z,2);
8.         JOptionPane.showMessageDialog(null,"y = "+y);
9.     }
10. }
```

يقوم هذا البرنامج بحل المعادلة التالية:

$$y = 4\sqrt{5x} + 3z^2$$

وذلك باستخدام الطرق الموجودة في صنف العمليات الحسابية (Math class).

تمرين رقم (2)

```
// Math_Methods_2.java
1. public class Math_Methods_2{
2.     public static void main( String args[]){
3.         int die ,coin ;
4.         char coinc;
5.         System.out.println(" The result is " );
6.         for (int i = 1;i<=10;i++){
7.             die = 1+(int) (Math.random()*6);
8.             coin = 1+(int) (Math.random()*2);
9.             if (coin==1)
10.                coinc='F';//F Face
11.            else
12.                coinc='T';//T Tail
13.            System.out.println(" " + i + "\t ( "+ die + " , "
                + coinc + " )");
14.        }// end for loop
15.    }// end of main
16. }// end of class RollDie
```

هذا البرنامج يقوم بعرض نتائج رمي قطعة نقد وحجر نرد 10 مرات في نفس الوقت.

الواجبات:

واجب رقم (1)

اكتب برنامج يقوم بما يلي:

- قراءة 3 متغيرات x, y, z من نوع int.
- طباعة اسم وقيمة المتغير الذي يحتوي على أكبر قيمة.
- طباعة اسم وقيمة المتغير الذي يحتوي على أصغر قيمة.
- حل المعادلة التالية:

$$z = 2 * x^2 + 4 * y^3$$

استخدم الطرق التالية: Math.pow(), Math.max(), Math.min()

واجب رقم (2)

اكتب برنامج يقوم بما يلي:

- طباعة نتائج عملية رمي قطعة نقد 10 مرات .
- طباعة عدد مرات ظهور الصورة .
- طباعة عدد مرات ظهور الكتابة .

واجب رقم (3)

اكتب برنامج لتقريب الأعداد الحقيقية :

- قراءة عدد حقيقي double.
- قراءة عدد المنازل المراد تقريب العدد لها.
- طباعة العدد قبل وبعد عملية التقريب.

التدريب السادس

(تعريف الطرق واستدعائها)

الأهداف:

- 1- التعرف على كيفية تعريف الطرق بجميع اشكالها.
- 2- التعرف على كيفية استدعاء الطرق بجميع اشكالها.

التمارين:

تمرين رقم (1)

```
// Method1.java
1. import javax.swing.JOptionPane;
2. public class Method1{
3.     public static void main(String[] args) {
4.         int num1,num2,num3,sumOfnNmbers,maxValue;
5.         num1=read("input the first number");
6.         num2=read("input the second number");
7.         num3=read("input the third number");
8.         sumOfnNmbers=sum(num1,num2,num3);
9.         maxValue=max(num1,num2,num3);
10.        print ("the maximum number is ",maxValue);
11.        print ("the sum of numbers is ",sumOfnNmbers);
12.    } //end of main
13.
14.    // a method to read an integer value
15.    static int read(String m){
16.        int n = Integer.parseInt(JOptionPane.showInputDialog(m));
17.        return n;
18.    } // end of read
19.
20.    // this method is to calculate the sum of three
21.    // integer numbers
22.    static int sum(int n1,int n2,int n3){
23.        int s;
24.        s=n1+n2+n3;
```

```
25. return s;
26. } //end of sum
27.
28. // this method returns the max of 3 integer numbers
29. // using Math Class
30. static int max(int n1,int n2,int n3){
31. int max;
32. max=Math.max(Math.max(n1,n2),n3);
33. return max;
34. } //end of max
35.
36. // this methos prints out an integer value with a
37. // message
38. static void print(String m,int n){
39. JOptionPane.showMessageDialog(null ,m+" "+n);
40. } //end of print
41. }
```

هذا البرنامج يحتوي على تعريف واستخدام للطرق التي تقوم بما يلي:

- أ - قراءة ثلاثة أعداد من خلال طريقة تقوم بعملية القراءة ().read()
- ب - إيجاد المجموع للأعداد المدخلة وذلك باستخدام طريقة ().sum()
- ج - إيجاد أكبر عدد من بين هذه الأعداد باستخدام الطريقة ().max()
- د - طباعة النتائج باستخدام الطريقة ().print()

الواجبات:

واجب رقم (1)

اكتب برنامج لقراءة أسماء 10 طلاب وكذلك قراءة 5 درجات لكل طالب. ويعرّف الطرق التالية:

- طريقة لقراءة الاسم.
- طريقة واحدة لقراءة الدرجات وإيجاد المجموع لهذه الدرجات ثم تقوم هذه الطريقة بإرجاع المجموع.
- طريقة لإيجاد المتوسط الحسابي (استخدم المجموع المرجع من الطريقة السابقة).
- طريقة لطباعة اسم الطالب والمتوسط الحسابي .

واجب رقم (2)

اكتب برنامج لقراءة 10 أعداد صحيحة، ويعرّف الطرق التالية:

- طريقة لعملية القراءة.
- طريقة لفحص العدد فيما إذا كان موجب أم سالب بحيث تقوم هذه الطريقة بعملية فحص العدد وتعيد true أو false، استخدم هذه الطريقة لإيجاد عدد الأعداد الموجبة وعدد الأعداد السالبة.
- طريقة لفحص فيما إذا كان العدد زوجي أم فردي بحيث تقوم هذه الطريقة بعملية الفحص وتعيد true أو false، استخدم هذه الطريقة لإيجاد مجموع الأعداد الزوجية ومجموع الأعداد الفردية.
- طريقة لطباعة النتائج.

التدريب السابع

(استخدام الطرق مع المصفوفات)

الأهداف:

- 1- التعرف على كيفية استخدام الطرق مع المصفوفات.
- 2- التعرف على التمرير باستخدام القيمة والتمرير باستخدام العنوان.
- 3- التعرف على متغيرات الصنف ومجال استخدامها.

التمارين:

تمرين رقم (1)

```
// arrays_with_methods.java
1. import javax.swing.JOptionPane;
2. public class arrays_with_methods {
3.     static double avg=0;
4.     static double sum=0;
5.     public static void main(String[] args) {
6.         String Employee[][]=new String[10][3];
7.         for (int i=0;i<Employee.length;i++){
8.             Employee[i][0]= readString("input name for
                Employee # "+ i);
9.             Employee[i][1]= readString("input job for Employee
                # " + i);
10.            Employee[i][2]= readSalary(i);
11.            sum=totalSalary(Employee);
12.            avg=avgSalary(sum,Employee.length);
13.        }
14.    }
15.    static String readString(String m){
16.        String s=JOptionPane.showInputDialog(m);
17.        return s;
18.    }
19.
20.    static String readSalary(int n){
```

```

21. double sal;
22. String s = JOptionPane.showInputDialog("Input salary
    for Employee # "+n);
23. sal= Double.parseDouble(s);
24. return ""+sal;
25. }
26.
27. static double totalSalary(String[][] emp) {
28. double sum1=0;
29. for (int i=0;i<emp.length;i++)
30. sum1+= Double.parseDouble(emp[i][2]);
31. return sum1;
32. }
33.
34. static double avgSalary(double sum1 ,int n) {
35. double avg=(double)(sum/n);
36. return avg;
37. }
38. static boolean checkSalary(double sal ) {
39. if (sal >= avg)
40. { return true;}
41. else
42. { return false;}
43. }
44. }

```

يقوم هذا البرنامج بما يلي:

قراءة أسماء، وظائف، ورواتب 10 موظفين وتخزينها في مصفوفة employee، طباعة اسم ووظيفة وراتب كل موظف وكذلك وضع إشارة (+) بجانب الراتب إذا كان أكبر أو يساوي معدل الرواتب وإشارة (-) إذا كان الراتب اقل من المعدل. وطباعة المجموع الكلي للرواتب وكذلك المعدل.

ويحتوي هذا البرنامج على الطرق التالية:

1. الطريقة readString لقراءة قيمة من نوع String حيث تقوم بإرجاع هذه القيمة.
2. الطريقة readSalary لقراءة قيمة من نوع double حيث تقوم بإرجاع هذه القيمة (يجب أن تكون القيمة المدخلة أكبر من صفر).
3. الطريقة totalSalary تقوم بحساب مجموع الرواتب لجميع الموظفين.

4. الطريقة avgSalary تقوم بحساب معدل الرواتب لجميع الموظفين.
5. الطريقة checkSalary تقوم بعمل فحص لراتب موظف معين إذا كان أكبر أو يساوي المعدل تعيد true وإلا تعيد false.
6. الطريقة printEmployee لطباعة عناصر المصفوفة.

الواجبات:**واجب رقم (1)**

اضف الطرق التالية إلى التمرين (1):

1. إضافة طريقة `maxSalary` تقوم بإرجاع رقم الصف الذي يحتوي على أكبر راتب.
2. إضافة طريقة `minSalary` تقوم بإرجاع رقم الصف الذي يحتوي على أقل راتب.
3. إضافة طريقة `getName` تعيد اسم الموظف في صف معين.
4. إضافة طريقة `getSalary` تعيد راتب الموظف في صف معين.
5. إضافة طريقة `getJob` تعيد وظيفة الموظف في صف معين.
6. إضافة طريقة `increaseSalary` تقوم بزيادة 10% لكل موظف راتبه أقل من متوسط الرواتب

استخدم هذه الطرق لما يلي:

1. طباعة اسم ووظيفة وراتب الموظف صاحب أكبر راتب.
2. طباعة اسم ووظيفة وراتب الموظف صاحب أقل راتب.

التدريب الثامن

(الاستدعاء الذاتي والتحميل الزائد للطرق)

الأهداف:

- 1- التعرف على كيفية التعامل مع الاستدعاء الذاتي للطرق.
- 2- التعرف على كيفية التعامل مع التحميل الزائد للطرق.

التمارين:

تمرين رقم (1)

```
// Rec-1.java
1. public class Rec-1{
2. public static void main(String args[])
3. {
4.     rec(5);
5. }
6. static void rec(int i)
7. {
8.     if (i!=0)
9.     {
10.         rec(--i);
11.         System.out.println("i = "+i);
12.     }
13. }
14. }
```

```
// Rec-2.java
1. public class Rec-2{
2. public static void main(String args[])
3. {
4.     rec(5);
5. }
6. static void rec(int i)
7. {
```



```

8.     if (i!=0)
9.     {
10.        System.out.println("i = "+i);
11.        rec(--i);
12.    }
13. }
14. }

```

```

// Rec-3.java
1.  public class Rec-3{
2.  public static void main(String args[])
3.  {
4.      rec(5);
5.  }
6.  static void rec(int i)
7.  {
8.      if (i!=0)
9.      {
10.         System.out.println("i = "+i);
11.         rec(i--);
12.     }
13. }
14. }

```

لاحظ الاختلافات بين البرامج الثلاثة أعلاه، علماً بأنها جميعاً تحتوي على طريقة استدعاء ذاتي تسمى .rec()

تمرين رقم (2)

```

// Overloading.java
1.  public class Overloading{
2.      public static void main(String args[])
3.      {
4.          double num1, num2;
5.          int num3;
6.          overload();
7.          overload(7, 4);
8.          num1=overload(3.5, 8);
9.          System.out.println(num1);
10.         num2=overload(7, 5.2);

```

```
11.      System.out.println(num2);
12.      num3=overload(3, 5, 9);
13.      System.out.println(num3);
14.  }
15.
16.  static void overload()
17.  {
18.      int a=9, b=5;
19.      System.out.println("No return, No
                           parameters:");
20.  }
21.
22.  static void overload(int a, int b)
23.  {
24.      System.out.println("int return, Two int
                           parameters:"+(a+b));
25.  }
26.  }
27.
28.  static double overload(double a, int b)
29.  {
30.      System.out.print("double return, double &
                           int parameters: ");
31.      return a+b;
32.  }
33.
34.  static double overload(int a, double b)
35.  {
36.      System.out.print("double return, int &
                           double parameters: ");
37.      return a+b;
38.  }
39.
40.  static int overload(int a, int b, int c)
41.  {
42.      System.out.print("int return, Three int
                           parameters: ");
43.      return a+b+c;
44.  }
45.  }
46. }
```

يحتوي هذا البرنامج على خمسة أشكال للطريقة (overload)، وذلك تطبيقاً للتحميل الزائد للطرق. لاحظ الطرق الخمسة ووضّح الفروقات فيما بينهم.

الواجبات:**واجب رقم (1)**

اكتب برنامج يحتوي على طريقة استدعاء ذاتي (Recursive Method) اسمها $\text{powerXY}()$ تستقبل معاملين من نوع int ، بحيث تقوم هذه الطريقة بإرجاع ناتج العدد مرفوعاً للأس (قوة)، ومثال على ذلك:

$$\text{powerXY}(5, 4) = 5^4 = 5 \times 5 \times 5 \times 5$$

واجب رقم (2)

اكتب برنامج يحتوي على ثلاث طرق لها نفس الاسم $\text{search}()$ لكل واحدة من هذه الطرق معامل واحد فقط، وتقوم هذه الطرق بالبحث عن عنصر معين داخل مصفوفة وأرجاع هل هذا العنصر موجود في المصفوفة ام لا، بحيث يتم تحديد الطريقة المراد استخدامها والمصفوفة المراد البحث فيها من خلال نوع المعامل الخاص بهذه الطريقة. يجب تعريف ثلاث مصفوفات int و char و String على مستوى الصنف. بحيث يتم إدخال بيانات 10 طلاب في المصفوفات الثلاثة من داخل الطريقة $\text{main}()$ ، وتتكون هذه البيانات من رقم الطالب، درجة الطالب بالحروف، واسم الطالب.

التدريب التاسع

(الطرق الخاصة بالسلاسل الرمزية)

الأهداف:

1- التعرف على الطرق الخاصة بالسلاسل الرمزية والموجودة مع الصنف Math.

التمارين:

تمرين رقم (1)

```
// stringExample.java
```

```
1. public class stringExample{
2.     public static void main(String args[]){
3.         String command="this is an example";
4.         String source = "wombat";
5.         String digits = "0123456789";
6.         String sub, tail;
7.         int pos, pos2;
8.         char c;

9.         System.out.println("\nWithout ignoring case:");
10.        if ( command.equals("quit"))
11.            System.out.println("\t\"command\" is equal to
                \"quit\");
12.        else
13.            System.out.println("\t\"command\" is NOT equal to
                \"quit\");
14.
15.        System.out.println("\nWith ignoring case:");

16.        if ( command.equalsIgnoreCase("quit"))
17.            System.out.println("\t\"command\" is equal to
                \"quit\");
18.        else
19.            System.out.println("\t\"command\" is NOT equal
                to \"quit\");
```

```

20. System.out.println();
21. if ( command.compareTo("dog")>0)
22.     System.out.println("\"command\" is after
                          \"dog\"\n");
23. // This code will be executed if String word is
24. // alphabetically after "dog"
25.
26. System.out.println("The result of:
                      \"A\".compareToIgnoreCase(\"a\")
                      is " +
                      "A".compareToIgnoreCase("a") +
                      "\n");

27. System.out.println("The result of:
                      \"A\".compareTo(\"a\") is " +
                      "A".compareToIgnoreCase("a") +
                      "\n");

28. pos = command.indexOf("example");
29. System.out.println("The position of \"example\" in
                      \"\" + command+"\" is " + pos +
                      "\n");

30. pos2 = command.indexOf( "example", pos );
31. System.out.println("The position of the second
                      occurrence of \"example\" in
                      \"\" + command+"\" is "+ pos2 +
                      "\n");

32. c = digits.charAt(5);
33. System.out.println("The character at position 5 of
                      \"\" + command+"\" is "+c+"\n");

34. sub = source.substring( 1, 3 );
35. System.out.println("The substring from
                      \"\"+source+"\" which started at
                      position 1 of the first 3
                      characters is " + "\"" + sub +
                      "\n");

36. tail = source.substring( 3 );

```

```
37. System.out.println("The last 3 characters of  
    \""+source+"\" is " + "\"" +  
    tail + "\"\n");  
38. }  
39. }
```

هذا المثال يبين كيفية استخدام بعض الطرق الخاصة في التعامل مع السلاسل الرمزية.

الواجبات:**واجب رقم (1)**

اكتب برنامج يطلب إدخال جملة نصية، ويقوم بتنفيذ ما يلي:

- 1- طباعة عدد حروف الجملة.
- 2- طباعة الجملة كاملة بعد تحويل جميع حروفها إلى حروف كبيرة.
- 3- طباعة الجملة بعد استبدال كل فراغ بالرمز \$.
- 4- طباعة عدد الكلمات في هذه الجملة.
- 5- طباعة الجملة بشكل معكوس.

برمجة الحاسب (عملي)

الأصناف والكائنات

التدريب العاشر (تعريف الأصناف وإنشاء الكائنات)

الأهداف:

- 1- التعرف على كيفية تعريف الأصناف.
- 2- التعرف على كيفية إنشاء الكائنات والوصول لمحتوياتها.

التمارين:

تمرين رقم (1)

```
1. class Car {
2. String manufacturer;
3. String model;
4. int year;
5. int passengers;
6. float cost;

7. // calculate the sale price of a car based on its cost
8. public double CalculateSalePrice() {
9. return cost * 1.5;
10. }

11. // a public constructor
12. public Car(String madeBy, String name, int yr, int pass,
13. float cst) {
14. manufacturer = madeBy;
15. model = name;
16. year = yr;
17. passengers = pass;
18. cost = cst;
19. }
```

```

20.// create and return a string with the basic details
21.// about this particular car

22.public String GetStats() {
23. return new String(year + " " + manufacturer + " " + model);
24.}
25.}

```

الصنف التالي يعرف كائن من نوع هذا الصنف ليتم استخدام مكوناته:

```

1. public class Ford {
2. public static void main(String args[]) {
3. Car myCar = new Car("Ford", "Grand", 1997, 4, 9000);
4. String str = myCar.GetStats();
5. System.out.println(str);
6. System.out.println(myCar.CalculateSalePrice());
7. }
8. }

```

عند تنفيذ هذا المثال على جهاز الحاسب ضع الصنفين في ملف واحد وسمه `Ford.java`.

تمرين رقم (2)

```

1. class Point {
2. // Variable declarations
3. private int x;
4. private int y;

5. // Method declarations
6. public Point() {
7. x = 0;
8. y = 0;
9. }

10. public Point(int xValue, int yValue) {

```

```
11. x = xValue;
12. y = yValue;
13. }

14. public Point(Point p) {
15. x = p.x();
16. y = p.y();
17. }

18. public int x() {
19. return x;
20. }

21. public int y() {
22. return y;
23. }

24. public void xSet(int xValue) {
25. x = xValue;
26. }

27. public void ySet(int yValue) {
28. y = yValue;
29. }

30. public Point add(Point p) {
31. return new Point(x+p.x(), y+p.y());
32. }

33. public Point add(int i,int j) {
34. return new Point(x+i,y+j);
35. }
36.}
```

الصنف التالي يعرف 3 كائنات من نوع هذا الصنف ليتم استخدام مكوناتهم:

```
1. public class Test{
2. public static void main(String args[]){
3. Point p1=new Point();
4. Point p2=new Point(5, 7);
5. Point p3=new Point(p2);
```

```
6. System.out.println("Before changing x and y in all
    points:");
7. System.out.println("\tIn p1, x=" + p1.x() + " and y=" +
    p1.y());
8. System.out.println("\tIn p2, x=" + p2.x() + " and y=" +
    p2.y());
9. System.out.println("\tIn p3, x=" + p3.x() + " and y=" +
    p3.y());
10. p1.xSet(9);
11. p1.ySet(2);
12. p2.xSet(4);
13. p2.ySet(8);
14. p3.xSet(10);
15. p3.ySet(22);
16. System.out.println("After changing x and y for all
    points:");
17. System.out.println("\tIn p1, x=" + p1.x() + " and y=" +
    p1.y());
18. System.out.println("\tIn p2, x=" + p2.x() + " and y=" +
    p2.y());
19. System.out.println("\tIn p3, x=" + p3.x() + " and y=" +
    p3.y());
20. p1=p1.add(2, 4);
21. p2=p2.add(p3);
22. p3=p3.add(p1);
23. System.out.println("After adding new points for all
    points:");
24. System.out.println("\tIn p1, x=" + p1.x() + " and y=" +
    p1.y());
25. System.out.println("\tIn p2, x=" + p2.x() + " and y=" +
    p2.y());
26. System.out.println("\tIn p3, x=" + p3.x() + " and y=" +
    p3.y());
27. }
28. }
```

عند تنفيذ هذا المثال على جهاز الحاسب ضع الصنفين في ملف واحد وسمه `.Test.java`.

الواجبات:

واجب رقم (1)

أنشئ صنف مسمى HugeInteger لتمثيل عدد صحيح كبير جداً وذلك باستخدام مصفوفة مكونه من أربعين موقع كل موقع يحتوي على خانة واحدة من خانات هذا الرقم. ويحتوي هذا الصنف على الطرق التالية:

- الطريقة inputHugeInteger لإدخال الرقم الصحيح الكبير جداً إلى المصفوفة.
- الطريقة outputHugeInteger لطباعة الرقم الصحيح الكبير جداً على الشاشة.
- الطريقة addHugeInteger لجمع عددين صحيحين كبيرين جداً.
- الطريقة subtractHugeInteger لطرح عددين صحيحين كبيرين جداً.
- الطريقة isEqualTo للسؤال فيما إذا كانا عددين صحيحين كبيرين جداً متساويين أم لا، بحيث ترجع true إذا كانا متساويين وترجع false إذا كانا غير متساويين.
- الطريقة isGreaterThan للمقارنة بين العددين الصحيحين وتحديد هل الأول أكبر من الثاني أم لا، بحيث ترجع true إذا كان العدد الصحيح الأول أكبر من العدد الصحيح الثاني وترجع false إذا كان غير ذلك.

اكتب برنامج لتعريف كائنات من نوع هذا الصنف وتطبيق جميع الطرق السابقة عليهم.

التدريب الحادي عشر

(مفهوم الوراثة والوصول للطرق الموروثة)

الأهداف:

- 1- التعرف على كيفية اشتقاق الاصناف.
- 2- التعرف على فائدة استخدام المؤشر this.

التمارين:

تمرين رقم (1)

```
// Test.java
```

```
1. class SimpleRecord {
2.   String firstName;
3.   String lastName;
4.   // Default constructor
5.   public SimpleRecord() {
6.     firstName = "";
7.     lastName = "";
8.   } // Constructor...
9.   public SimpleRecord(String firstName, String lastName){
10.    this.firstName = firstName;
11.    this.lastName = lastName;
12.  }
13. // List the elements of the record...
14. public void list() {
15.   System.out.println("\tFirst Name: " + firstName);
16.   System.out.println("\tLast Name: " + lastName);
17.   System.out.println();
18. }
19. }
```

```
20. class AddressRecord extends SimpleRecord {
21. String address;

22. // Default constructor
23. public AddressRecord() {
24. firstName = "";
25. lastName = "";
26. address = "";
27. }

28. // Constructor...
29. public AddressRecord(String firstName, String lastName,
30. String address) {
31. this.firstName = firstName;
32. this.lastName = lastName;
33. this.address = address;
34. }

35. public void printDetails(){
36. System.out.println("\tFirst Name: " + firstName);
37. System.out.println("\tLast Name: " + lastName);
38. System.out.println("\tAddress: " + address);
39. System.out.println();
40. }
41. }

42. public class Test{
43. public static void main(String args[]){
44. AddressRecord record;
45. record = new AddressRecord("Thomas", "Jefferson",
46. "Monticello");
47. System.out.println("The First and the Last Name
48. ONLY:");
49. record.list();
50. System.out.println("The First Name, the Last Name, and
51. the Address:");
52. record.printDetails();
53. }
54. }
```


تمرين رقم (2)

```
// TestReserchAssistant.java
```

```
1. class Student {
2.     protected long id;
3.     protected String name;
4.     protected double gpa;

5.     public Student(long id, String name, double gpa) {
6.         this.id = id;
7.         this.name = name;
8.         this.gpa = gpa;
9.     }

10.    public Student() {
11.        this(999999, "No name", 0.0);
12.    }

13.    public void changeGPA(double newGPA) {
14.        gpa = newGPA;
15.    }

16.    public double getGPA() {
17.        return gpa;
18.    }

19.    public void print() {
20.        System.out.print(id+"\t"+name+ "\t"+gpa);
21.    }
22. }

23. class ResearchAssistant extends Student {
24.     private int workLoad; // in hours

25.     ResearchAssistant(long id, String name, double gpa, int workLoad){
26.         this.id = id;
27.         this.name = name;
28.         this.gpa = gpa;
29.         this.workLoad = workLoad;
```

```
30. }  
  
31. ResearchAssistant() {  
32. id = 999999;  
33. name = "No name";  
34. gpa = 0.0;  
35. workLoad = 0;  
36. }  
37.  
    public void print() {  
38. super.print();  
39. System.out.print("\t" + workLoad);  
40. }  
41. }  
  
42. class TestReserchAssistant {  
43. public static void main (String[] args) {  
44. ResearchAssistant s1;  
45. s1 = new ResearchAssistant();  
46. s1.print();  
47. ResearchAssistant s2;  
48. s2 = new ResearchAssistant(991234, "Ali Elayyan", 3.45, 15);  
49. s2.changeGPA(3.75);  
50. System.out.println();  
51. s2.print();  
52. System.out.println();  
53. }  
54. }
```

الواجبات:**واجب رقم (1)**

أنشئ صنف لتمثيل التاريخ بحيث يسمى Date وله الخصائص التالية:

- يتم إخراج التاريخ بأحد الأشكال التالية:

MM/DD/YYYY

April 01, 2004

DDD YYYY

- استخدم التحميل الزائد للبيانات لإنشاء كائنات تحمل قيماً ابتدائية للتاريخ حسب أشكال التاريخ السابقة.

- ثم أنشئ صنف اسمه BirthDate مشتق من الصنف Date بحيث يحتوى على بعض الطرق

(Methods) الخاصة به، مثل: طريقة لمقارنة التاريخ الحالي بتاريخ مدخل لتحديد هل هذا التاريخ

تاريخ الميلاد أم لا.

ثم اكتب برنامج يستخدم كائنات من نوع هذا الصنف.

التدريب الثاني عشر (استبدال الطرق الموروثة)

الأهداف:

- 1- التعرف على كيفية استبدال تعريف الطرق الموروثة.
- 2- التعرف على فائدة استخدام المؤشر super.

التمارين:

تمرين رقم (1)

// Test.java

```
1. class Employee {
2.     private String name;
3.     private double salary;
4.     public Employee(String n, double s) {
5.         name = n;
6.         salary = s;
7.     }

8.     public String toString() {
9.         String str="Name: " + name + ", Salary=" + salary;
10.        return str;
11.    }
12. }

13. class Manager extends Employee {
14.     private String department;
15.     public Manager(String n, double s, String d) {
16.         name = n;
17.         salary = s;
18.         department = d;
19.     }

20.    public String toString() {
```

```
21. return super.toString() + ", Department=" + department;
22. }
23. }

24. public class Test {
25. public static void main(String[] args) {
26. Employee e = new Employee("Ibrahim", 65000);
27. Manager m = new Manager("Abdullah", 85000, "Engineering");
28. System.out.println(e);
29. System.out.println(m);
30. }
31. }
```

تمرين رقم (2)

```
// Test.java

1. class Car {
2. private int year;
3. private float originalPrice;

4. // calculate the sale price of a car based on its
5. // cost
6. public double CalculateSalePrice() {
7. double salePrice;
8. if (year > 1994)
9. salePrice = originalPrice * 0.75;
10. else if (year > 1990)
11. salePrice = originalPrice * 0.50;
12. else
13. salePrice = originalPrice * 0.25;
14. return salePrice;
15. }

16. // a public constructor
17. public Car(int year, float originalPrice) {
18. this.year = year;
19. this.originalPrice = originalPrice;
20. }
21. }
```

```
21. class ClassicCar extends Car {
22. // calculate the sale price of a car based on its
23. // cost
24. public double CalculateSalePrice() {
25. return 10000;
26. }

27. // a public constructor
28. public ClassicCar(int year, float originalPrice) {
29. super(year, originalPrice);
30. }
31. }

32. public class Test{
33. public static void main(String args[]){
34. ClassicCar myClassic = new ClassicCar(1920, 1400);
35. double classicPrice =
36. myClassic.CalculateSalePrice();
37. System.out.println(classicPrice);
38. Car myCar = new Car(1990, 12000);
39. double price = myCar.CalculateSalePrice();
40. System.out.println(price);
41. }
42. }
```

المناقشة:

مناقشة مشروع المقرر

يتم مناقشة المشروع الذي تم تحديده للطلاب في الاسبوع السادس من الفصل الدراسي والذي يغطي جميع مفردات المقرر. بحيث تتم المناقشة لكل مشروع وبشكل منفصل وخلال الأسابيع: الثالث عشر، الرابع عشر، والخامس عشر.

المراجع

References

1. Deitel and Deitel, Java: How to Program, 3rd Edition, Prentice Hall, 2001
2. Patrick Naughton and Michael Morrison, the Java Handbook, McGraw-Hill, 1996
3. Bruce Eckel, Thinking in Java (2nd Edition), 2001
4. م. فادي حجار، لغة البرمجة JAVA 2، دار شعاع للنشر والعلوم، 2001
5. كن أرنولد، لغة برمجة جافا، مركز التعريب والبرمجة، 2001

المحتويات

الصفحة

1	الوحدة الأولى: المصفوفات
1	التدريب الأول: المصفوفات ذات البعد الواحد
5	التدريب الثاني: ترتيب المصفوفات
9	التدريب الثالث: طرق البحث في المصفوفات
14	التدريب الرابع: المصفوفات ذات البعدين
18	الوحدة الثانية: الطرق
18	التدريب الخامس: الطرق الخاصة بالعمليات الحسابية
21	التدريب السادس: تعريف الطرق واستدعائها
24	التدريب السابع: استخدام الطرق مع المصفوفات
28	التدريب الثامن: الاستدعاء الذاتي والتحميل الزائد للطرق
33	التدريب التاسع: الطرق الخاصة بالسلاسل الرمزية
37	الوحدة الثالثة: الأصناف والكائنات
37	التدريب العاشر: تعريف الأصناف وإنشاء الكائنات
42	التدريب الحادي عشر: مفهوم الوراثة والوصول للطرق الموروثة
47	التدريب الثاني عشر: استبدال الطرق الموروثة
51	المراجع

