حل الدورة الأولى لعام 2022 هندسة طبية برمجة1\2

1-1- المتغيرات كانت حل عندما كان عددها قيل واصبحث متعبة عندما كثرت ممانطلب التعريف بالجمله أي الانتقال للمصفوفات وهذه بدورها تعرف نوعاً واحداً من المعطيات ضمن مصفوفة واحدة مما نطلب وجود السجلات التي تملك بيانات مختلفة وبرمجيات تطبق على هذه البيانات والتي اكتملت بصورة الأنماط التي أعطت افق في مجال الوراثة والتحميل الذاند وتعدد الأشكال ومعالجة الإستثناءات ...إلخ

1-2-تخزين المصفوفات والتجوال بها سهل ومريح وإنما الإضافه والحذف ضمنها متطلب للوقت نظرا لعملية الازاحة المطلوبه مما جعل الإنتقال إلى اللائحة الأحادية الارتباط **SLinkList** ولكن هذه عانت من عملية الحذف لضرورة معرفة العقدة السابقة للعقدة المرغوب جذفها مما جعل الإنتقال للوائح ثنائية الارتباط **DLL** لمعرفة العقدة السابقة والتالية بسهولة.

1-3-الصنق يعبر عنه بثلاث خلايا الأولى يكتب بها اسم الصنف بحط غامق والثانية يتم وضع أسماء وأنماط الحقول وتسبق باشارة السالب إن كانت خاصة و+ إن كانت عامه والثالثة يمتب المنهج الباقي بين الإشارتين « » واسم ونوع البارامترات له، يليها أسماء المناهج المستخدمة في الصنف مع أسماء وانماط المتغيرات بين قوسين داريين وإذا أعاد المنهج قيمة يذكر نمطها مابعد القوس الدائري الغالق للبارامترات ثم نقتطبين فوق بعض ومن حيث اشارتي + و – كما ذكر سابقاً.

1-1- String name = JOptionPane.showInputDialog("What is your name?");

1-2- يتم نداء المنهج showInputDialog من الكائن showInputDialog وعطانة المتغير المحرفي ?What is your name ليظهر علي نافذة الإدخال فوق صندوق الادخال وسيتم اسناد المدخل للمرجع name من النمط String.

- تم اشتقاق كائن input من الصنف Scanner من خلال منادات المعامل new والباني للصنف Scanner واعطانه System.in من أجل تأمين الادخال من لوحة المفاتيح . منادات المنهج nextInt() من نسخة الكائن input لتأمين الإدخال ونسبه للمتغي number1.

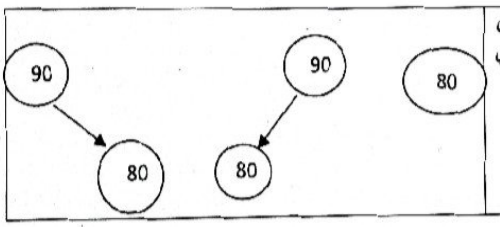
2-2-إنشاء كائن new Account(); حساب Account ونسبه للكائن myAccount.

قراءة سطر نصي من خلال المنهج nextLine(); الموجود في نسخة الكائن input ونسبة للمرجع theName مم النمط String.

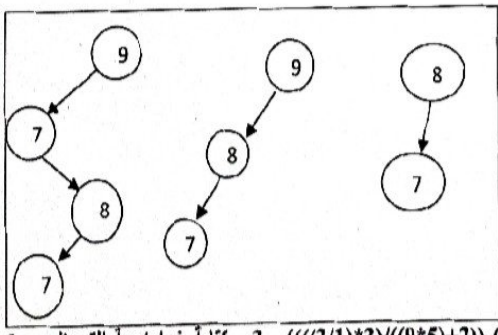
وضع المتغير (الأسم theName) في المرجع myAccount من خلال المنهج setName().

2-3-يتم رسم قوس ممتلئ باللون الافتراضي، نقطة البداية لها الاحداثيين centerX-counter*radius, centerY-counter*radius وبعرض counter*radius*2 وارتفاع counter * radius * 2 وينطلق من الدرجة 30 وتمسح نصف دائره باتجاه عقارب الساعة

III 3-1-حالة عقدة لاتملك أبناء وهنا يمكن أن تكون عقدة جذر عندها يتم وضع null في مؤشر root أي root=null أو عقدة يسارية للأب عندها نضع في الابن اليساري للاب null أي pare.LC=nul وغير ذلك عقدة يمينة للأب عندها نضع في الابن اليميني للاب null أي pare.RC=null



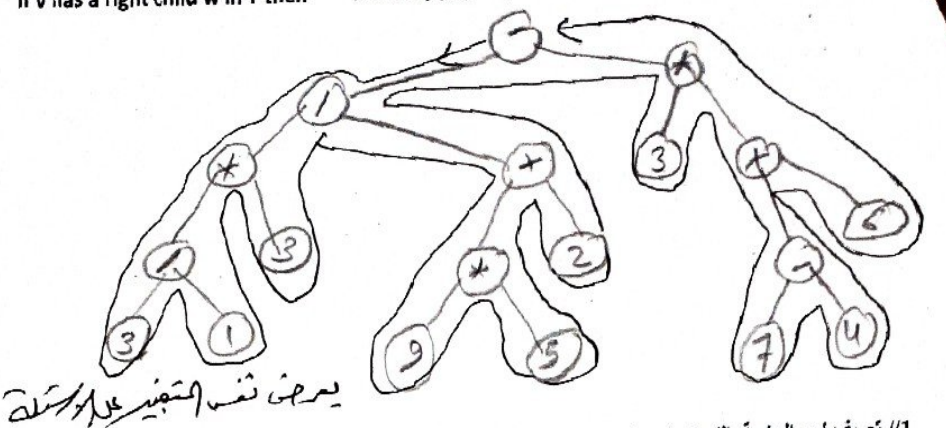
2-3-عقدة تملك ابن يساري فقط وهنا يمكن أن تكون عقدة جذر عندها يتم وضع Curr.LC في مؤشر root أي root= Curr.LC أو عقدة يسارية للأب عندها نضع في الابن اليساري للاب Curr.LC أي pare.LC= Curr.LC وغير ذلك عقدة يمينة للأب عندها نضع في الابن اليميني للاب Curr.LC أي pare.RC= Curr.LC



III 2- رسم الشجرة المعبره عن ((((3/1)*3)/((9*5)+2))-((3*(7-4))+(6)) **3-** كتابة خوارزمية التجوال inorder traversal وتطبيقها على الشجره السابقة. - كتابة خوارزمية التجوال inorder traversal وتطبيقها على الشجره السابقة.

```
Algorithm inorder(T,v):  // LPR
  if v has a left child u in T then      inorder(T,u)    {recursively traverse left subtree)
   perform the "visit" action for node v
```

if v has a right child w in T then inorder(T,w) {recursively traverse right subtree}



يعرض نقص المتبقي على الأسئلة

1// تعريف اسم الحزمة التي نعمل بها.

2// تعريف مصفوفة ألوان خاصة باسم[]colors ككائن من الصنف الأساس Color وعناصرها من عناصر المصفوفة الأم Color ويضاف اليهما عنصران معرفان بشكل خاص VIOLET,INDIGO.

3// تعريف للمنهج الباني ()DR3 للصنف DR3 والوارث للصنف JPanel والاستفادة من المنهج setBackground(Color.WHITE); من أجل وضع الخلفية بيضاء.

4// اسناد القيمة الصحيحة للمتغير الصحيح shapesColor والتي تحسب من قيمة الازاحة 2 مضافة للقيمة التي يعيدها المنهج nextInt والتي تحدد قيمها من 0 إلى 6 والموجود ضمن الصنف العشوائي randomNumbers.

5// إضافة اللوحة panel على التطبيق application من خلال المنهج add.

6// تفعيل حالة التطبيق application من خلال المنهج setVisible على القيمة true كي يصبح ظاهر.

سيتم طباعة الترويسة التالية، ثم سطر جديد.

```
"figNum Color      x1    y1     w     h    shapType re=1 OR ov=2
```

أول قيمه تعاد هي في السطر randomNumbers.nextInt(2)=1 يضاف لها 1 فتصبح 2 وتسند لـ shapesType

ثاني قيمه تعاد هي في السطر randomNumbers.nextInt(195)=87 وتسند لـ x1

القيمة الثالثة التي تعاد هي في السطر randomNumbers.nextInt(195)=60 وتسند لـ y1

القيمة الرابعة التي تعاد هي في السطر randomNumbers.nextInt(190)=184 وتسند لـ w

القيمة الخامسة التي تعاد هي في السطر randomNumbers.nextInt(190)=115 وتسند لـ h

القيمة السادسة التي تعاد هي في السطر randomNumbers.nextInt(7)=6 سيتم لإضافة 2 عليها لتصبح 8 وتسند لـ shapesColor=8

سيتم بناء السلسلة المحرفية s حيث يسند لها الفراغ أولاً ويدمج به colorsn[shapesColor]=colorsn[8] بالعودة للمصفوفة colorsn سنجة قيمة السلسلة RED التي ستضاف للفراغ سيتم اعتماد اللون من خلال المنهج setColor من الكائن g وفق البارمترات colors[shapesColor]=colors[8] وبالنظر للمصفوفة colors سنجد Color.RED أي اللون الأحمر .

الأن تعليمه الخرج من خلال المنهج System.out.printf ويتم ترتيب الخرج 6 مسافات للعدد الصحيح 6d% والموافقة للعدد counter=2 الحلقة تنازلية وبالهامش الافتراضي اليميني،

يليها التنسيق 8s- ثمانية مواضع لسلسلة محرفية s= RED وهنا السلسلة RED والهامش يساري نظراً لوجود إشارة - .

يليها التنسيق 5d 5 مسافات للعدد الصحيح x1 وسملك القيمة 87 .

يتكرر ذلك اربع مرات لمتغيرا بالقيم المحسوبه سابقاً وهي y1=60, w=184, h=115 من اليسار لليمين.

يليها التنسيق 15d أي 15 مسافات للعدد الصحيح shapesType .

ستحصل على السطر التالي مابعد سطر الترويسة.

```
figNum Color      x1    y1     w     h    shapType re=1 OR ov=2
     2 RED        87    60   184   115           2
```

سيتم اختبار if(shapesType==1) سيتم رسم مستطيل زالا سيتم رسم قطع وهي حالتنا وهنا 2 هذا يعني سيتم تنفيذ g.fillOval(x1, y1, w, h); وهو عبارة عن قطع المماس اليساري والعلوي يتقاطعان عند 87,60 وعرضه 184 وارتفاعه 115. يكتفي من الطالب حتى هذه النقطه .

- سبب كتابة //values of nextInt 1,87,60,184,115,6,0,51,121,45,93,7 (علامه واحدة فقط).

هو عدم قلب ورقة الأسئلة 12 مرة على الأقل لتتبع البرنامج أو أن يقوم الطالب بكتابتها.

انتهى الحل