|  |
| --- |
| **Nom et prénom : ………………………………………………………………………….**  **Note :………………………………………………………………………………………** |

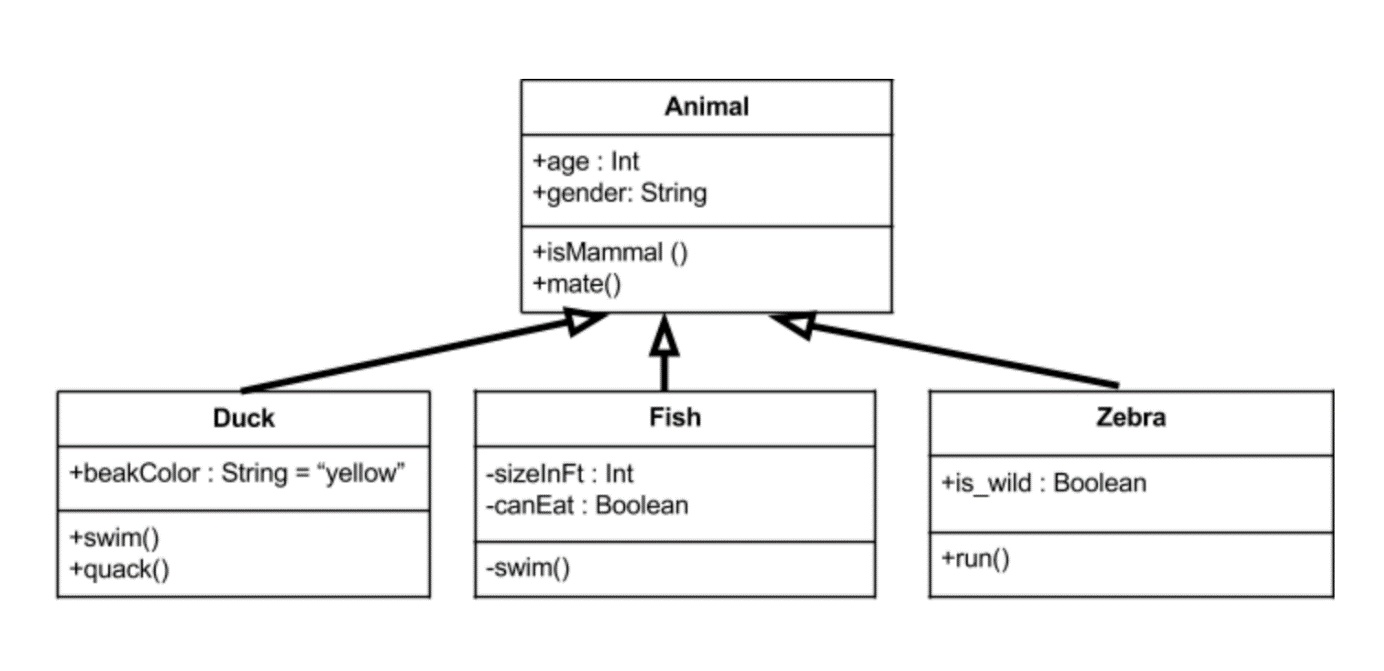
**QUIZ**

**Partie 1**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Vrai** | **Faux** |
| 1 | La classe exception permet de gérer les imprévues et erreurs d’exécution du programme |  |  |
| 2 | L'objet "this" est pour faire référence à une méthode de la classe |  |  |
| 3 | Une classe interface est un concept abstrait dans une phrase verbale, dont les fonctions sont à redéfinir chez les sous catégories de ce concept |  |  |
| 4 | L’héritage multiple en java est possible grâce à la notion de classe abstraite et la redéfinition des méthodes |  |  |
| *5* | Dans une phrase, l’héritage est une association possible entre deux mots de type « nom » dont il y a une relation de hiérarchisation de type « mère-fille » et les « petit fils » |  |  |
| 6 | Interface Graphique, est un dispositif de  « dialogue homme-machine », dans lequel les objets à manipuler sont dessinés sous forme de pictogrammes à l'écran |  |  |
| 7 | Les composants : Bouton, Champs de texte, Libellé… sont ajouter à l’objet de type « JFrame » |  |  |
| 8 | Le JDK : contient en plus du JVM, l’environnement de développement Java (JDK = JVM + autres) |  |  |
| 9 | Le langage Java a été crée par J. Gosling et P. Naughton à l’issue du projet « Green » de l’entreprise « Microsystem electronic», en 1991. |  |  |
| 10 | les structures de contrôles prévues en java sont uniquement :   * IF-ELSE * SWITCH-CASE * FOR * FOR-EACH * WHILE |  |  |
| 11 | Les entrées/sorties en java sont gérés sous forme de flux : « stream », entre le programme java et la source/destination des données. |  |  |
| 12 | « ActionListener » est une interface Java, qui réagit à un événement déclencher par le clavier. |  |  |
| 13 | « ActionListener » dispose d’une méthode parmi d’autre, appelé «  actionPerformed », que la classe qui implémenterait devrait la redéfinir. |  |  |
| 14 | JDBC est une série de classe prédéfinit, permettant de communiquer avec une base de donnée, pour effectuer l’ensemble des traitements |  |  |
| 15 | Pour libérer la connexion on appelle la méthode **Exist(-1);** |  |  |
| 16 | ***« try »*** : permet d’entourer le bloc de traitement utilisant une des méthodes identifiées comme génératrice d’exception. |  |  |
| 17 | ***« catch »*** : permet d’être à l’écoute pour qu’en cas de production de l’exception, on exécute le traitement prévu |  |  |
| 18 | « RuntimeException » et « IOException » hérite de la classe Throwable |  |  |
| 19 | « FileNotFoundException » est une sous classe la classe « IOException » |  |  |
| 20 | « ArithmeticException » est une sous classe de la classe « IOException » |  |  |

**Partie 2 :**

**Soit le diagramme de classe ci-dessous :**



|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Vrai** | **Faux** |
| 21 | La classe « Animal » est une classe mère des classes filles « Duck », « Fish », « Zebra ». |  |  |
| 22 | La classe « Animal » devrait être une classe abstraite |  |  |
| 23 | La syntaxe pour implémenter la classe abstraite : abstract class Animal {…} |  |  |
| 24 | La classe « Animal » dispose d’un constructeur, les constructeurs des classes filles peuvent ne pas appeler le constructeur « Animal » |  |  |
| 25 | La classe « Zebra » devrait avoir des accesseurs uniquement |  |  |
| *26* | La méthode « isMammal() » devrait être redéfinit chez les classes filles |  |  |
| 27 | La méthode « swim() » devrait être redéfinit chez les deux classes « Duck » et « Fish » |  |  |
| 28 | beakColor = ‘yellow’ est un attribut statique |  |  |
| 29 | Pour manipuler l’attribut « age », la syntaxe Animal.age est correcte |  |  |
| 30 | La classe « Animal » est une classe interface, si il dispose des prototypes des méthodes « isMamal() », « mate() », « run() », « swim() » et « quack() » |  |  |

**Partie 3 :**

**Soit le code source ci-dessous :**

|  |
| --- |
| 1. **import** javax.swing.\*; 2. **public** **class** FirstSwingExample { 3. **public** **static** **void** main(String[] args) { 4. JFrame f=**new** JFrame(); //creating instance of JButton 6. JButton b=**new** JButton("click"); 7. b.setBounds (130,100,100, 40); 9. f.add(b);//adding button in Jpanel     f.setSize(400,500);  f.setLayout(**null**);  f.setVisible(**true**);  }  } |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Vrai** | **Faux** |
| 31 | Les commentaires indiqués sur ce code source sont incorrecte. |  |  |
| 32 | L’objet b est une instance de la classe « Button » un des composants de la classe des interfaces graphiques « AWT » |  |  |
| 33 | Le bouton « b » affichera un message « click » à l’utilisateur |  |  |
| 34 | Le JFrame f est de hauteur « 400 » et de largeur « 500 » |  |  |
| *35* | Le paramètre « null » de la méthode setLayout permet de ne pas utiliser un layout manager |  |  |
| *36* | Ce code source contient une seule erreur de compilation. |  |  |

**Partie 4 :**

|  |
| --- |
| 1. **import** java.sql.\*; 2. **class** MysqlCon{ 3. **public** **static** **void** main(String args[]){ 4. **try**{ 5. Class.forName("com.mysql.jdbc.Driver"); 6. Connexion conx=DriverManager.getConnection( 7. "jdbc:mysql://localhost:3306/sonoo","root","root"); 8. Statement stmt=con.createstatement();   ResultSet rs=stmt.executeQuery("select \* from emp");  **while**(rs.next())  System.out.println(rs.getInt(1)+"  "+rs.getString(2)+"  "+rs.getString(3));  con.close();  }**catch**(Exception e){ System.out.println(e);}  }  } |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Vrai** | **Faux** |
| 37 | Ce code source est correcte. Et il permet de récupérer les données des trois champs de la table « emp » |  |  |
| 38 | Ce code source contient deux erreurs de compilation uniquement |  |  |
| 39 | La table emp est crée au niveau de la base « sonoo » au niveau du serveur de base de donnée Mysql. Les champs de cette table est de type String |  |  |
| 40 | Le mot de passe de root est « root » et le serveur qui héberge la base de données, écoute les requêtes sur le port 3306. |  |  |

**Partie 4 :**

**Soit le code source ci-dessous :**

|  |
| --- |
| class OuterClass {  int x = 10;  private class InnerClass {  int y = 5;  }  }  public class Main {  public static void main(String[] args) {  OuterClass myOuter = new OuterClass();  OuterClass.InnerClass myInner = myOuter.new InnerClass();  System.out.println(myInner.y + myOuter.x);  }  } |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Vrai** | **Faux** |
| 41 | La classe « InnerClass » est une classe interne privée. Cette declaration est impossible pour ce type de classe |  |  |
| 42 | Si vous ne voulez pas que des objets extérieurs accèdent à la classe interne, déclarez la classe comme privée |  |  |
| 43 | L’objet « myInner » ne va pas s’instancier |  |  |
| 44 | Le code en haut va provoquer une erreur d’exécution et pas de compilation |  |  |
| 45 | Le code en haut va provoquer 3 erreurs de compilation |  |  |
| 46 | Une « ArrayList » est une classe de Java.util, il permet un stockage des éléments même dupliquer. C’est une classe qui hérite de la classe Interface « List » |  |  |
| 47 | Pour ajouter dans une ArrayList on utilise la méthode « get() » et pour récupérer un element de l’ArrayList on utilise la méthode « add() » |  |  |
| 48 | Pour parcourir un « ArrayList » on utilise généralement la boucle « while » plus que la boucle « For » |  |  |

**Partie 5 :**

**Soit le code source ci-dessous :**

|  |
| --- |
| public class Box<T> {  private T t;  public void add(T t) {  this.t = t;  }  public T get() {  return t;  }  public static void main(String[] args) {  Box<Integer> integerBox = new Box<Integer>();  Box<String> stringBox = new Box<String>();    integerBox.add(new Integer(10));  stringBox.add(new String("Hello World"));  System.out.printf("Integer Value :%d\n\n", integerBox.get());  System.out.printf("String Value :%s\n", stringBox.get());  }  } |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Vrai** | **Faux** |
| 49 | La classe Box est une classe générique parcequ’il possède des méthodes add et get générique |  |  |
| 50 | Les deux objets « integerBox » et « stringBox » sont de type la classe générique « Box » dont le paramètre générique est respectivement de type Entier et Chaine de caractère |  |  |