Algorithmique Programmation - NFA032

Exemple d'examen de juin 2013 - durée 2h30

Exercice 1 : Exception (6 points)

```
public class TestException {
	static String plusZ(String x) throws Exp {
		if (x == null) {
			throw new Exp();
		} else
			return "Z" + x;
	}

	public static void main(String[] args) {
		String s;
		String[] t = new String[10];
		t[0] = "A0";
		t[1] = "A1";
		t[2] = "A2";
		t[9] = "A9";
		System.out.println("1ERE BOUCLE ");
		try {
			for (int i = 0; i < t.length; i++) {
				s = t[i];
				System.out.println(plusZ(t[i]));
			}
		} catch (Exp e1) {
			System.out.println("Exception 1 ");
		}
		System.out.println("2EME BOUCLE ");
		for (int i = 0; i < t.length; i++) {
			s = t[i];
			try {
				System.out.println(plusZ(t[i]));
			} catch (Exp e2) {
				System.out.println("Exception 2 ");
			}
		}

	}
}
class Exp extends Exception {
}
```

QUESTIONS

1. Quels sont les messages affichés ?

2. Que se passe t-il si on remplace " `catch (Exp e1)`" par " `catch (Exception e1)`" ?

Exercice 2 : Objet et héritage (6 points)

La classe `Visite` représente une visite par
- son nom (une chaîne de caractères),
- sa distance en km (un réel),
- sa durée en heures (un réel),

```
class Visite {
	String nom ;
	double distance ;
	double duree ;

	public Visite (String nom, double distance, double duree){
		this.nom = nom;
		this.distance = distance;
		this.duree = duree;
		}
	public void affiche(){
		Terminal.ecrireStringln("nom : " + this.nom) ;
		Terminal.ecrireString ( "distance = "+ this.distance) ;
		Terminal.ecrireStringln ( " duree = "+ this.duree) ;
	}
	public double cout(){
		return 0.1 * this.distance;
	}
}
```

Un visite avec nuit possède les mêmes caractéristiques qu'un trajet avec en plus :
- le nombre de nuits (un int)

La classe `VisiteAvecNuit` est une classe fille de la classe `Visite ;` qui possède la variable d'instance supplémentaire :

- `nombreDeNuit(int)`.

1. Ecrire la classe `VisiteAvecNuit`. Elle doit comporter un constructeur à 4 paramètres.

2. Pour une `VisiteAvecNuit`, le mode de calcul du coût est le suivant :

cout = 0.1 * distance + 30* `nombreDeNuit`

Redéfinir la méthode `cout()` dans la classe `VisiteAvecNuit`.

3. La classe `TestVisites` permet de tester les deux classes :

```
public class TestVisits{
	public static void main (String args[] ) {
		String nom ="DURANT JEAN" ;
		Visite a = new Visite (nom, 160.0, 6.0);
		VisiteAvecNuit b = new VisiteAvecNuit (nom, 160.0, 56.0, 2);
		a.affiche();
		b.affiche();
		System.out.println("cout :" +a.cout());
		System.out.println("cout:" +b.cout());
		}
}
```

Quelles sont les lignes affichées ? **Justifier chaque réponse**.

Exercice 3 : Dessin d'objet (4 points)

```
public class Produit{
    int reference, quantite;
    double prix;
    char categorie;
    Produit(int r, double p, char c){
        reference = r;
        quantite = 0;
        prix = p;
        categorie = c;
    }
    void arriveeDepart(int n){
        quantite = quantite + n;
    }
    public static void main(String[] a){
        Produit prod1, prod2;
        Produit[] t = new Produit[2];
        prod1 = new Produit(10023,5.23,'A');
        t[0] = prod1;
        t[1] = new Produit(10047,6.60,'A');
        prod1.arriveeDepart(50);
    }
}
```

Représentez au moyen d'un petit dessin les objets et tableaux existant à la fin de l'exécution du programme donné ci-dessus. Chaque objet sera représenté par un rectangle et chaque référence par une flèche.
Attention : on ne vous demande pas un diagramme de classe. C'est chaque objet créé et chaque tableau qui doit être représenté.

Exercice 4 : Objet et Liste chaînée (4 points)

Un Colis est constitué de gâteaux.

1. Ecrire `poids`() une méthode d'instance de la classe `GateauListe` qui retourne le poids du gâteau.

2. Ecrire `poids`() une méthode d'instance de la classe `ListeGateau` qui retourne le poids total du colis.

3. Ecrire dans la classe `TestListe` une méthode main qui teste les fonctionnalités de la classe `ListeGateau.`

La classe `Gateau` est déjà définie :

```
public class Gateau {
	int num;
	String descriptif;
	double poids;
	public Gateau(int num, String descriptif, double poids) {
		this.num = num;
		this.descriptif = descriptif;
		this.poids = poids;
	}
	public double getPoids() {
		return poids;
	}
}
```

La classe `GateauListe` est définie par :

```
public class GateauListe {
	private Gateau g;
	private GateauListe suivant;

	public GateauListe(Gateau g, GateauListe suivant) {
		this.g = g;
		this.suivant = suivant;
	}
	public GateauListe(Gateau g) {
		this.g = g;
		this.suivant = null;
	}
	public Gateau getGateau() {
		return g;
	}
	public void setValeur(Gateau g) {
		this.g = g;
	}
	public GateauListe getSuivant() {
		return suivant;
	}
	public void setSuivant(GateauListe suivant) {
		this.suivant = suivant;
	}
}
```

La classe `ListeGateau` est définie par :

```
public class ListeGateau{
	GateauListe premier;

	public GateauListe getPremier() {
		return premier;
	}
	public void ajouterAudebut(Gateau g){
		GateauListe ancienPremier=premier;
		premier=new GateauListe(g,ancienPremier);
	}
	// etc...
}
```