Laboratoire 05D – Estimation des coûts de construction d'une route ( sous-classes, interface graphique )

Nous ferons dans cet exercice un modèle simple servant à estimer les coûts rattachés à la construction d'une route au Québec, sujet ô combien d'actualité !

Utilisez le projet ProjetRoute présent dans LÉA. Il contient déjà une interface graphique FrameAutoroute qu’on utilisera pour tester nos classes modèles.

1. Dans ce projet, créez une nouvelle classe Route représentant une route à être construite. Les variables d'instance à utiliser sont les suivantes :

* une variable contenant le nombre de kilomètres que fera la route
* une variable contenant le nombre de ponts que contiendra la route à construire
* une variable contenant le nombre de passages fauniques installés dans cette route
* une variable représentant si les accotements de cette route seront asphaltés ou non ( boolean )

Qu’est-ce qu’un passage faunique ? Un passage faunique est un ensemble de hautes clôtures, de grilles et de passages sous la route empêchant les grands cervidés de se retrouver sur la route. On en retrouve entre autres sur l’autoroute 50 entre Lachute et Hawkesbury :



2. Comme à l’habitude, codez un constructeur permettant d’initialiser les variables à l’aide de paramètres et des méthodes d’accès ( get ) pour chacune des variables

3. Codez une méthode calculerCout qui retournera le coût estimé de construction de cette route. Le calcul se fera de la manière suivante, vous pouvez calculer en millions de dollars.

|  |  |
| --- | --- |
| Étapes | Coût ( en millions de $ ) |
| Route | 3,4 par kilomètre |
| Pont | 6 |
| Passage faunique | 0,7 |
| Asphaltage des accotements | 0,1 par kilomètre |

4. Réalisez maintenant une sous-classe Autoroute ( une Autoroute est une route ) . Un objet Autoroute aura une donnée supplémentaire, soit le nombre d’échangeurs qui seront présents sur cette autoroute.

De plus, une autoroute aura toujours ses accotements asphaltés. Codez un constructeur représentant cette situation.

5. Vous devez implémenter en surcharge la méthode calculerCout dans cette sous-classe car les coûts pour construire un pont ou pour assurer des accotements asphaltés est différent dû aux doubles voies de circulation. Voici les coûts de construction :

|  |  |
| --- | --- |
| Étapes | Coût ( en millions de $ ) |
| Autoroute ( chaussée ) | 6,8 par kilomètre |
| Pont | 6 |
| Passage faunique | 0,7 |
| Asphaltage des accotements | 0,2 par kilomètre |
| Échangeur | 7 |

6. Nous pouvons maintenant utiliser l’interface graphique pour tester notre modèle. L’interface est déjà construite mais il reste à coder des éléments. Suivez les explications en classe pour vous y retrouver…

AIDE – MÉMOIRE

pour chercher le contenu d’un champ texte : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pour écrire dans un champ texte : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pour transtyper une String en un int : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pour transtyper une String en double : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pour transtyper un type prédéfini ( int, double ) en String : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DecimalFormat : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SOLUTIONS

Classe Route

|  |
| --- |
| package projetroutes;  public class Route {   private double nbKilometres;  private int nbPonts;  private int nbPassagesFauniques;  private boolean accotementAsphalte;   public Route(double nbKilometres, int nbPonts, int nbPassagesFauniques, boolean accotementAsphalte) {  this.nbKilometres = nbKilometres;  this.nbPonts = nbPonts;  this.nbPassagesFauniques = nbPassagesFauniques;  this.accotementAsphalte = accotementAsphalte;  }    public double getNbKilometres() {  return nbKilometres;  }   public int getNbPonts() {  return nbPonts;  }   public int getNbPassagesFauniques() {  return nbPassagesFauniques;  }   public boolean isAccotementAsphalte() {  return accotementAsphalte;  }   public double calculerCout () {  double total = nbKilometres \* 3.4;  total += nbPonts \* 6;  total += nbPassagesFauniques \* 0.7;  if ( accotementAsphalte )  total += nbKilometres \* 0.1;  return total;  } } |

Classe Autoroute

|  |
| --- |
| package projetroutes;  public class Autoroute extends Route {  private int nbEchangeurs;   public Autoroute(double nbKilometres,int nbPonts, int nbPassagesFauniques,int nbEchangeurs ) {  super( nbKilometres, nbPonts, nbPassagesFauniques, true);  this.nbEchangeurs = nbEchangeurs;  }    public int getNbEchangeurs() {  return nbEchangeurs;  }     public double calculerCout () {  double total = getNbKilometres() \* 6.8;  total += getNbPonts() \* 6;  total += getNbPassagesFauniques() \* 0.7;  total += getNbKilometres() \* 0.2;  total += nbEchangeurs \* 7;  return total;  } } |