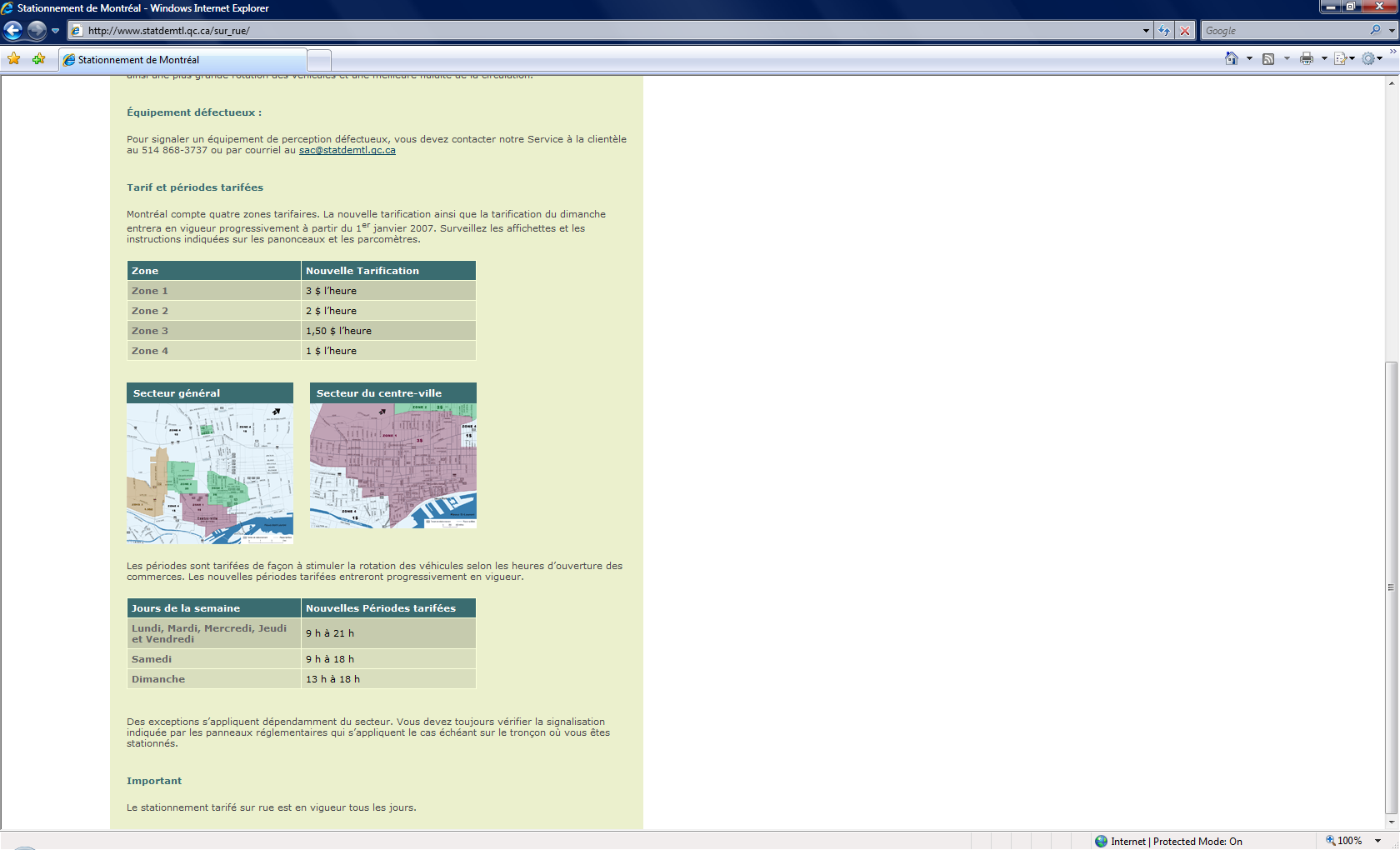
**TRAVAIL PRATIQUE 1**

**BORNE INFORMATISÉE *PAYEZ-ET-PARTEZ***

Ce travail aura pour but de modéliser une application s’apparentant aux bornes informatisées de stationnement installées dans le quartier ( sur la rue Ontario, face au cégep notamment ). Vous devez donc réaliser une application orientée objet à l’aide des renseignements inclus dans cet énoncé. Vous pouvez également vous référer au site web de la société Stationnement de Montréal : <http://www.statdemtl.qc.ca/>

**Mode de fonctionnement des bornes informatisées de stationnement**

**1. Heures de fonctionnement de ces bornes :**



Votre système devra donc tenir compte de ces heures. En dehors des périodes tarifées, votre système devra indiquer qu’elle n’attend aucun paiement de la part des automobilistes.

Comment trouver l’heure du système ? Utiliser un GregorianCalendar 🡪 Voir notes de cours en classe

**2. Tarifs des bornes automatisées en fonction de la zone où elles sont situées :**

Ville-Marie : 4,00$

Pour vous simplifier la tâche, votre modèle de Borne devra représenter le fait qu’un objet Borne se situera toujours dans la zone Ville-Marie; le tarif sera donc de 4$ de l’heure. Il faut noter **qu’un automobiliste ne peut stationner sa voiture pour plus de deux heures. De plus, la machine ne remet pas la monnaie.**

**3. Marche à suivre pour utiliser la borne de stationnement :**

* L’usager doit entrer le code correspondant à l’espace de stationnement choisi ( A009, B678, etc… )
* Si le code est valide ( **une lettre suivie de trois chiffres** ) :
  + *Une transaction est créée*
* l’usager a ensuite deux options :
  + *Paiement comptant* :
    - Il entre une pièce de monnaie ( 0.25 cents, 1$ ou 2$ ) 🡪 la borne affiche le nombre de minutes disponibles avec cette pièce / l'heure de fin du stationnement
    - Il entre d’autres pièces jusqu’à ce qu’il clique sur OK ou que le maximum soit atteint
  + *Paiement par carte de crédit :*
    - Il entre sa carte de crédit ( numéro et date d’expiration sous la forme mm/aa ) 🡪 on vérifie si la carte n’est pas expirée.
    - Si elle ne l’est pas, l’usager utilise les touches 25cents et MAX pour choisir le temps désiré 🡪 le temps en minutes correspondant au montant est affiché / l'heure de fin du stationnement est affichée
    - Il peut continuer jusqu’à ce qu’il clique sur OK.
* Lorsque l’usager clique sur OK, on doit produire un coupon confirmant la transaction. Ce coupon devra comme comprendre comme informations :
  + Le type de paiement ( crédit ou comptant )
  + Le coût de la transaction
  + Les heures de début et de fin du stationnement (sous forme de  GregorianCalendar)
  + L’espace de stationnement choisi

**4. Ajout : Méthode de contrôles**

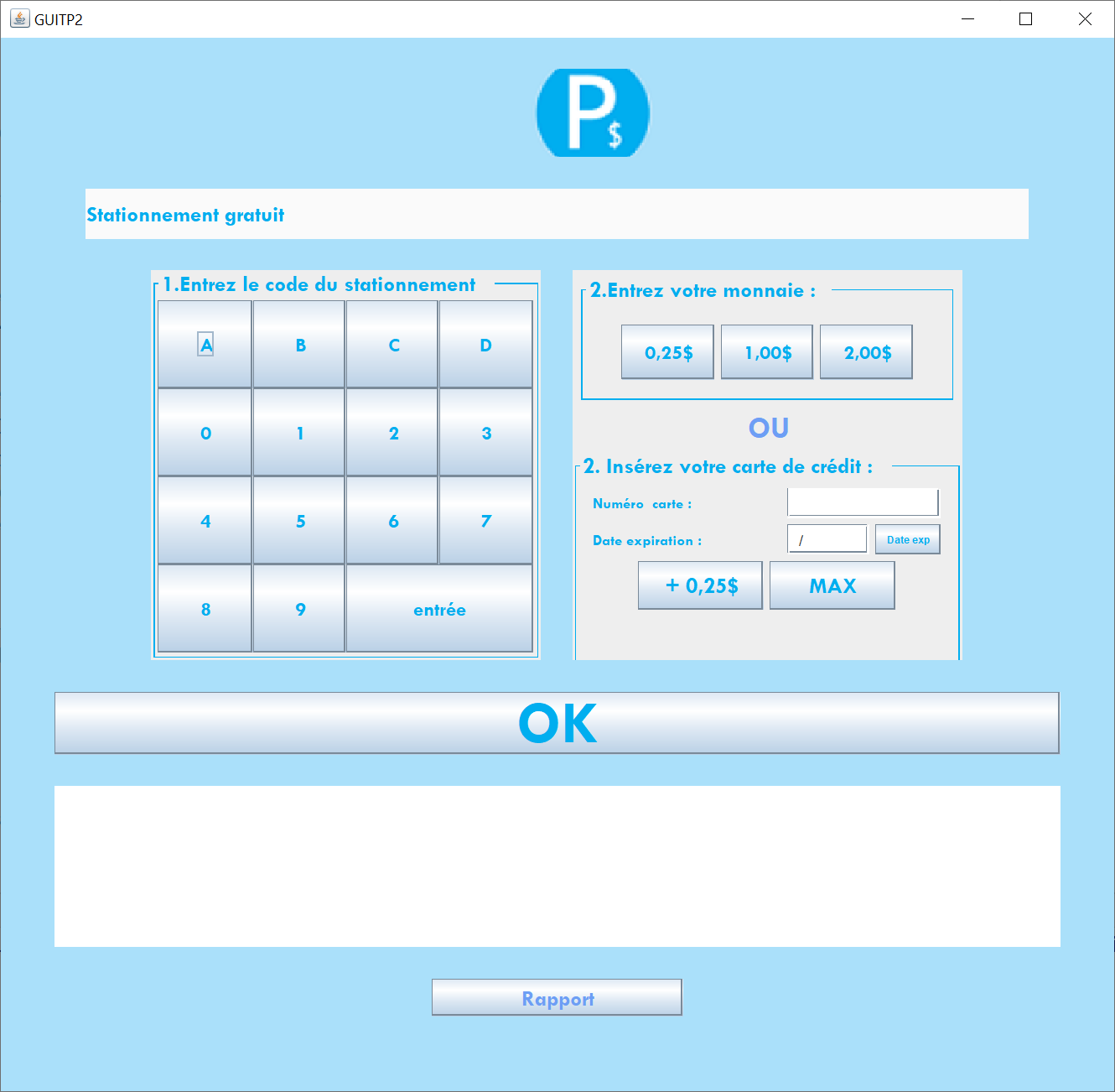
Nous ajouterons un module de contrôle afin de représenter une action utilisée uniquement par un agent de stationnement :

* Rapport : vérifier la somme d’argent ( monnaie + crédit ) contenue dans la borne informatisée à un instant donné

**5. Modalités du travail :**

* Il est bien important de saisir qu’il s’agit d’un travail de programmation orientée objet; il faut donc retrouver plusieurs classes et les manipuler.
* Les classes qui devront être utilisées sont les suivantes :
  + Piece : représentant une pièce de monnaie ( 25 cents, 1$ ou 2$ )
    - *Quelle(s) variable(s) d’instance représentera(ont) son état ?*
    - *A-t-elle des comportements pouvant être représentés par des méthodes ?*
  + CarteCredit : représentant une carte de crédit servant à acquitter le stationnement
    - *Quelle(s) variable(s) d’instance représentera(ont) son état ?*
    - *A-t-elle des comportements pouvant être représentés par des méthodes ?*
    - *Comment simuler le solde de la carte de crédit ? ( vu qu'on ne sait pas encore comment monter une base de données de cartes de crédit )*
  + Transaction : représentant une Transaction à la borne
    - *Quelle(s) variable(s) d’instance représentera(ont) son état ?*
    - *A-t-elle des comportements pouvant être représentés par des méthodes ?*
  + Borne : représentant la borne de stationnement ( principale classe )
    - *Quelle(s) variable(s) d’instance représentera(ont) son état ?*
    - *A-t-elle des comportements pouvant être représentés par des méthodes ? OUI, PLUSIEURS…*

Pour l’instant, vous ne vous concentrez que sur le modèle. **Vous obtiendrez plus tard le GUI ( interface graphique ) qui se servira de votre modèle**. Voici un GUI possible :



Donc, rappelez-vous que votre modèle, constitué de plusieurs classes, **ne doit rien afficher pour le moment.**

**À Venir : Codage de l’interface graphique**

|  |  |
| --- | --- |
|  |  |
| **Explications modèle ( ce document )** | **24 Avril** |
| **Explications GUI graphique** | **4 Mai** |
| **Remise finale** | **8 Mai** |
|  |  |

**note**

**Pour simuler une carted ecredit on peutfaire un sub string avec les deux premier chiffre de la carte moins les deux dernier chiffre**

**Utilise random avec la classe math methode static math.random**

**Il existe aussi des objet random rndom r = new random**

**La borne est-ce qu’elle est fonctionnelle en fonction de l’heure avec un bool**

**L’argent totale de la machine**

**Si je veuc un methode pour le recu**

**On s’en fou si une place déjà occupe ou non**

**Methode qui Verifie sur le code est valide ou pas code de stationnement en paramattre**

**Accept l’argent qui prend en paramettre piece Accepterpiece(Piece p)**

**Valider carte de credit, date d’expiration validerCartedecredit( carte) on check l’expiration de la carte dans la borne et non dans carte de credit**

**Avec gc les mois commence a 0 alors faire +1**