

TP2 de C++

Document de spécification et de conception

B3211 : Quentin « johnny » Guye, Louis Mer

Introduction

L'objectif de ce TP est de programmer un ensemble de capteurs sur un réseau routier. Chaque capteur envoie l'état du trafic sous la forme d'un code couleur : Vert, Jaune, Rouge ou Noir.

L'application doit être en mesure d'effectuer plusieurs fonctions à l'aide des informations fournies par les capteurs.

I - Spécification

Afin de réaliser l'application, nous avons décidé d'utiliser 3 objets différents : un représentant le réseau routier, auquel est rattaché un ensemble de capteurs, qui contiennent des données.

Lorsqu'un enregistrement est ajouté au programme, le capteur est créé si il n'existe pas encore et les données sont ensuite sauvegardées.

II - Conception

Le programme est composé de 3 objets.

Traffic

- Représente le réseau routier

Sensor

- Représente un capteur du réseau routier

TrafficData

- Représente un enregistrement des données du trafic routiers

A) Traffic

1. Attributs

La classe Traffic possède 3 attributs.

Sensor SensorsTab**

- Tableau contenant les capteurs du réseau.

int filled,

- Valeur de la dernière case non remplie du tableau SensorsTab.

int allocated,

- Valeur de la taille alloué au tableau SensorsTab.

2. Constructeurs

La classe Traffic contient 2 constructeurs.

Traffic()

- Sans paramètres, initialise l'objet à l'aide d'une valeur constante afin d'allouer la bonne taille au tableau SensorsTab et la bonne valeur à la variable allocated, et assigne à filled la valeur 0.

Traffic(int nbSensors)

- Prend en paramètre un entier représentant le nombre de capteur, il agira alors comme le premier constructeur en utilisant cette valeur à la place de la constante du programme.

3. Méthodes

Traffic possède 6 méthodes.

Sensor* getSensor(const int ID)

- Retourne le capteur dont l'identifiant est passé en paramètre.

bool addEvent(int numSensor, int year, int month, int day, int hour, int minute, char trafficValue, int dayWeek)

- Crée un nouvel objet TrafficData avec les valeurs passées en paramètre dans l'objet Sensor correspondant à *numSensor*. Si le capteur n'existe pas, il est alors créé au préalable.
- Retourne vrai si la création est un succès.

bool getStatsForASensor(const int numSensor)

- Affiche les statistiques du trafic pour le capteur dont l'identifiant est passé en paramètre.
- Retourne vrai si le capteur existe.

bool getStatsJamADAH(const int dayWeek)

- Affiche en pourcentage les embouteillages (« état Rouge ou Noir ») pour tous les capteurs, heure par heure, pour un jour de la semaine passé en paramètre (valeur entre 1 et 7).
- Retourne vrai si il y a des données pour ce jour dans le programme.

bool getStatsForADayWeek(const int dayWeek)

- Affiche les statistiques du trafic pour un jour de la semaine passé en paramètre.
- Retourne vrai si il y a des données pour ce jour dans le programme.

bool optimizedPath(const int dayWeek, const int hourStart, const int hourEnd, const int seg_count, int* tabSeg)

- Affiche le meilleur moment de départ pour un trajet sur le réseau routier ainsi que le temps de parcours estimé. Le jour de la semaine ainsi que la plage horaire(hourStart, hourEnd) sont passés en paramètre, ainsi que le nombre de segments à parcourir et les identifiants des capteurs assignés à ces segments.
- Le temps de parcours est calculé en minutes et suppose que les états Vert, Jaune, Rouge et Noir prennent respectivement 1, 2, 5 et 10 minutes à parcourir.
- Retourne vrai si l'heure de départ est bien inférieure à l'heure d'arrivée.

B) Sensor

1. Attributs

La classe Sensor possède 4 attributs.

int ID

- Identifiant du capteur, qui correspond à l'indice du tableau de capteur dans la classe Traffic.

int* filled

- Un tableau d'entier contenant les indices des dernières cases vides pour toutes les dimensions du tableau eventsPerDay.

int* allocated

- Un tableau d'entier contenant le nombre de cases allouées pour toutes les dimensions du tableau eventsPerDay.

TrafficData eventsPerDay**

- Un tableau contenant les objets TrafficData liés au capteur, séparé en plusieurs dimensions représentant les jours de la semaine.

2. Constructeurs

La classe Sensor contient 1 constructeur.

Sensor (int ID)

- Crée un capteur ayant l'identifiant passé en paramètre.
- Initialise à l'aide de constantes les tableaux filled, allocated et eventsPerDay.

3. Méthodes

Sensor possède 7 méthodes.

int getID() const

- Retourne l'identifiant du capteur.

TrafficData* getEventsPerDay() const**

- Retourne le tableau eventsPerDay.

bool addEvent(int year, int month, int day, int hour, int minute, char trafficValue, int dayWeek)

- Crée un nouvel objet TrafficData dans le tableau eventsPerDay. Il sera assigné à la dernière case libre de la dimension correspondante au jour de la semaine passé en paramètre.

int* getStats() const

- Retourne un tableau contenant les statistiques du trafic enregistré par le capteur.

double* getStatsJamADAH(int dayWeek) const

- Retourne un tableau contenant les pourcentages d'embouteillage (« état Rouge ou Noir ») enregistrés par le capteur, heure par heure, pour le jour de la semaine passé en paramètre.

double* getStatsForADayWeek(int dayWeek) const

- Retourne les statistiques du trafic enregistré par le capteur pour le jour de la semaine passé en paramètre.

char* optimizedPath(const int dayWeek, const int hourStart, const int hourEnd, const int seg_count, int* tabSeg)

- Retourne un tableau de caractères contenant l'état du trafic minutes par minutes.

C) TrafficData

1. Attributs

La classe Sensor possède 6 attributs.

int year

- l'année d'enregistrement de la donnée.

int month

- le mois d'enregistrement de la donnée.

int day

- Le jour d'enregistrement de la donnée.

int hour

- L'heure d'enregistrement de la donnée.

int minute

- Les minutes de l'enregistrement.

char trafficValue

- L'état du trafic de la donnée sous la forme d'un caractère : V,J,R ou N.

2. Constructeurs

La classe Sensor contient 1 constructeur:

TrafficData(int year, int month, int day, int hour, int minute, char trafficValue)

- Initialise tous les attributs avec les paramètres correspondants.

3. Méthodes

La classe TrafficData possède 6 méthodes.

int getYear()

- Retourne l'année de la donnée.

int getMonth()

-Retourne le mois de la donnée.

int getDay()

- Retourne le jour de la donnée.

int getHour()

- Retourne l'heure de la donnée.

int getMinute()

- Retourne les minutes de la donnée.

char getTrafficValue()

- Retourne l'état du trafic de la donnée.