

Labo 1 - Taxi

Valentin Ricard and Ewan Mariaux

October 13, 2022

1 Informations

Nom du fichier source: taxi-v2.cpp

2 Code

```
1  //-----
2  // Fichier      : LaboTaxi.cpp
3  // Auteur(s)   :
4  //              v1 -> Philibert Alexandre, Villegas Castrillon Adrian
5  //              v2 -> Valentin Ricard, Ewan Mariaux
6  // Date        : 2021-10-02
7  // But          : Un programme qui calcule le prix d'une course en € en taxi
8  // Modifications :
9  // Remarque(s) : Le programme ne gère pas les saisies
10 //-----
11
12 #include <cstdlib>    // EXIT_SUCCESS
13 #include <iostream>   // cin et cout
14 #include <iomanip>    // setprecision
15 #include <limits>     // numeric_limits<streamsize>
16 #include <cmath>      // ceil(...) et max()
17
18 /// Affiche une date, en forçant la taille à 2, et en rajoutant des 0 si nécessaire.
19 #define FORMAT_DATE setw(2) << setfill('0')
20
21 /// Définir le format de nombres pour l'affichage, en forçant la taille à 6
22 #define FORMAT_NOMBRE setw(6) << setfill(' ')
23
24 /// Vider le buffer (on recoit un retour à la ligne)
25 #define VIDER_BUF cin.ignore(numeric_limits<streamsize>::max(), '\n')
26
27 /// Affiche et attend l'interaction de l'utilisateur pour arrêter le programme.
28 #define FINIR cout << "presser ENTREE pour quitter"; VIDER_BUF; return EXIT_SUCCESS
29
30 using namespace std;
31
32 // rajout des f pour préciser que ce sont des float car manquant sur le programme initial.
33 const float PRISE_EN_CHARGE = 5.00f;
34 const float SUPPLEMENT_PAR_BAGAGE = 2.50f;
35 const float TARIF_MINUTE_JOUR = 1.0f;
36 const float TARIF_MINUTE_NUIT = 1.6f;
37
38 const int HEURE_DEBUT_JOUR = 8;
39 const int MINUTE_DEBUT_JOUR = 0;
40 const int DEBUT_JOUR_EN_MINUTES = HEURE_DEBUT_JOUR * 60 + MINUTE_DEBUT_JOUR;
41
42 const int HEURE_FIN_JOUR = 20;
```

```

43  const int MINUTE_FIN_JOUR = 0;
44  const int FIN_JOUR_EN_MINUTES = HEURE_FIN_JOUR * 60 + MINUTE_FIN_JOUR;
45
46  int main() {
47
48      float distanceKm,
49            vitesseKmH,
50            supplementBagages,
51            prixCourseJournee, // prix des heures durant le jour
52            prixCourseNuit,    // prix des heures durant la nuit.
53            prixTotal;
54
55      int heureDepart,
56          minutesDepart,
57          tempsCourse,
58          nombreBagages; // Il n'est pas possible d'avoir 1/2 bagage, ils sont représentés
↪   par un entier
59
60
61      // On initialise cette variable, car la course peut ne pas passer par les chemins
↪   candides.
62      int tempsCourseMatin = 0,
63          tempsCourseTarifJournee,
64          tempsCourseSoir = 0,
65          tempsCourseTarifNuit,
66          departEnMinutes,
67          ecartArriveeSoir,
68          ecartDepartMatin;
69
70
71
72      // définit la précision d'affichage des nombres à virgules (exemple: 2.50)
73      cout << fixed << setprecision(2);
74
75      cout << "bonjour, ce programme calcule le prix d'une course en € en taxi"
76            << "voici les conditions" << endl
77            << "===== " << endl
78            << " - prise en charge      : " << FORMAT_NOMBRE << PRISE_EN_CHARGE << endl
79            << " - supp par bagage      : " << FORMAT_NOMBRE << SUPPLEMENT_PAR_BAGAGE << endl
80            << // Modification des constantes pour refléter le changement dans les besoins
81            << " - tarif/minute (jour) : " << FORMAT_NOMBRE << TARIF_MINUTE_JOUR << endl
82            << " - tarif/minute (nuit) : " << FORMAT_NOMBRE << TARIF_MINUTE NUIT << endl
83            << " - tarif jour          : "
84            << HEURE_DEBUT_JOUR << "h" << FORMAT_DATE << MINUTE_DEBUT_JOUR
85            << "-"
86            << HEURE_FIN_JOUR << "h" << FORMAT_DATE << MINUTE_FIN_JOUR
87            << endl;
88
89      cout << "votre commande" << endl
90            << "===== " << endl;
91
92      cout << " - nbre de bagage      [0-4]: ";
93
94      cin >> nombreBagages;
95      // Changement du vider de buffer explicite à la macro pour faciliter la maintenance
96      VIDER_BUF;
97      // Rajout d'une vérification
98      if (nombreBagages < 0 || nombreBagages > 4) {
99          cout << "Le nombre de bagages n'est pas valide." << endl;
100          FINIR;
101      }

```

```

102
103     cout << " - distance [km]    [0-500]: ";
104     cin >> distanceKm;
105     VIDER_BUF;
106     if (distanceKm < 0 || distanceKm > 500) {
107         cout << "La distance n'est pas valide." << endl;
108         FINIR;
109     }
110
111     cout << " - vitesse [km/h] [30-120]: ";
112
113     cin >> vitesseKmH;
114     VIDER_BUF;
115     if (vitesseKmH < 30 || vitesseKmH > 120) {
116         cout << "La vitesse n'est pas valide." << endl;
117         FINIR;
118     }
119
120     cout << " - depart           [hh:mm]: ";
121     cin >> heureDepart;
122     cin.ignore(numeric_limits<streamsize>::max(), ':');
123     cin >> minutesDepart;
124
125     if (heureDepart < 0 || heureDepart > 23
126         || minutesDepart < 0 || minutesDepart > 60) {
127         cout << "L'heure de départ n'est pas valide." << endl;
128         FINIR;
129     }
130
131     VIDER_BUF;
132
133     // Calculs
134     departEnMinutes = heureDepart * 60 + minutesDepart;
135     // Séparation du temps de course avec le prix de la course pour les opérations
136     ↪     suivantes
137     tempsCourse = (int) ceil(distanceKm / vitesseKmH * 60);
138
139     ecartDepartMatin = DEBUT_JOUR_EN_MINUTES - departEnMinutes;
140     if (ecartDepartMatin > 0) {
141         // On utilise max pour ne pas dépasser le temps de course
142         tempsCourseMatin = min(tempsCourse, ecartDepartMatin);
143     }
144
145     ecartArriveeSoir = (departEnMinutes + tempsCourse) - FIN_JOUR_EN_MINUTES;
146     if (ecartArriveeSoir > 0) {
147         tempsCourseSoir = min(tempsCourse - tempsCourseMatin, ecartArriveeSoir);
148     }
149
150     tempsCourseTarifNuit = tempsCourseMatin + tempsCourseSoir;
151     tempsCourseTarifJournee = tempsCourse - tempsCourseTarifNuit;
152
153
154     prixCourseJournee = (float) tempsCourseTarifJournee * TARIF_MINUTE_JOUR;
155     prixCourseNuit = (float) tempsCourseTarifNuit * TARIF_MINUTE_NUIT;
156
157     // Modification du supplément bagage pou enlever un warning de compilation
158     supplementBagages = SUPPLEMENT_PAR_BAGAGE * (float) nombreBagages;
159
160     prixTotal = PRISE_EN_CHARGE + supplementBagages + prixCourseJournee + prixCourseNuit;
161

```

```

162     cout << endl
163     << "votre ticket" << endl
164     << "=====" << endl
165     << " - prise en charge   : " << FORMAT_NOMBRE << PRISE_EN_CHARGE << endl
166     << " - supp bagages     : " << FORMAT_NOMBRE << supplementBagages << endl
167     << " - temps course" << endl
168
169     << FORMAT_NOMBRE << tempsCourseTarifJournee << "' @ 1.00   : " << FORMAT_NOMBRE <<
↪ prixCourseJournee << endl
170     << FORMAT_NOMBRE << tempsCourseTarifNuit << "' @ 1.60   : " << FORMAT_NOMBRE <<
↪ prixCourseNuit << endl
171
172     << "-----" << endl
173     << "TOTAL : " << prixTotal << endl;
174
175     FINIR;
176 }

```