



Rapport de Projet – Programmation Évènementielle

Gestion d'un jardin d'enfants

Par : Mme. Mariem Ennigrou, Mr. Kamel Khemissi

Réalisée par : Timoumi Bassem, Essonni Hayfa, Neffati Mohammed

Classe : 1 LNIG 4

13/12/2023

Principe d'application

Ce projet Gestion d'un jardin d'enfants, a pour objectif l'implémentation d'une application simple basée sur l'interfaçage graphique permettant au directeur la bonne gestion et le bon suivi de son jardin d'enfants.

Quels sujets seront abordés dans ce rapport ?

- Les bibliothèques utilisées.
- Les fonctions prédéfinies utilisées.
- La structure du projet.
- Implémentation.
- Références.

I. Les bibliothèques utilisées :

- **System.Drawing** : en VB.NET, elle est utilisée pour manipuler des images, créer des graphiques 2D, et effectuer des opérations de dessin. Elle offre des fonctionnalités pour charger, enregistrer et modifier des images, ainsi que pour créer des éléments graphiques sur des surfaces comme des formulaires.
- **System.Windows.Forms.VisualStylesVisualStyleElement** : est utilisé pour personnaliser l'apparence des contrôles graphiques en fonction du style visuel du système d'exploitation, assurant une intégration harmonieuse de l'interface utilisateur avec l'apparence générale de Windows. Il permet d'ajuster visuellement des éléments tels que les boutons en fonction de l'état actuel de l'application.
- **System.Net.Security** : est utilisé pour implémenter des protocoles de sécurité tels que SSL/TLS, assurant ainsi la confidentialité et l'intégrité des données échangées entre les clients et les serveurs lors de communications réseau, comme dans les connexions HTTPS. Il facilite la création de connexions sécurisées.
- **System.IO** : offre des classes et des fonctionnalités pour effectuer des opérations d'entrée/sortie (E/S), permettant la manipulation de fichiers et de flux de données. Il sert à lire, écrire, copier, déplacer et gérer des fichiers, des répertoires, et des flux de données, facilitant ainsi la gestion des opérations liées aux fichiers et aux données dans les applications .NET.

II. Les fonctions prédéfinies utilisées :

- **Me.Hide()** : masque temporairement la fenêtre actuelle dans une application Windows Forms.
- **Form.Show()** : rend visible la fenêtre masquée dans une application Windows Forms.
- **Textbox.Text** : permet de lire ou de définir le contenu texte actuel d'un contrôle TextBox.
- **MessageBox.Show** : affiche une boîte de dialogue modale avec un message dans une application Windows Forms.
- **File.Exists** : vérifie si un fichier existe à un emplacement spécifié, renvoyant une valeur booléenne.
- **File.ReadAllLines('nomfichier').ToList()** : lit toutes les lignes d'un fichier et les stocke dans une liste.
- **Lines.count** : renvoie le nombre de lignes dans le fichier.
- **Line.split** : divise la ligne en un tableau de sous-chaînes en utilisant le délimiteur par défaut.
- **.length** : renvoie la longueur d'une chaîne ou la taille d'un tableau.
- **RemoveAt()** : supprime l'élément à un index de la liste.

- **File.WriteAllLines** : est utilisé pour écrire un tableau de lignes dans un fichier spécifié.
- **.IsNullOrEmpty()** : renvoie True si la chaîne spécifiée est nulle ou vide, sinon False.
- **.Contains('')** : renvoie True si la chaîne '' est présente dans la chaîne, sinon False.
- **.ToString()** : convertit l'objet courant en sa représentation sous forme de chaîne de caractères.
- **MessageBoxButtons.OK** : est une valeur énumérée représentant le bouton "OK" dans une boîte de dialogue MessageBox.
- **MessageBoxIcon.Information** : est une valeur énumérée représentant une icône d'information dans une boîte de dialogue MessageBox.
- **MessageBoxIcon.Error** : est une valeur énumérée représentant une icône d'erreur dans une boîte de dialogue MessageBox.
- **SelectedItem** : représente l'élément actuellement sélectionné dans un contrôle ComboBox.
- **.SelectedIndex** : donne l'indice de l'élément sélectionné dans un contrôle ComboBox, ou -1 s'il n'y a pas d'élément sélectionné.

- **.Value** : dépend du contexte ; cela peut représenter la valeur d'une propriété dans un contrôle ou d'un objet.
- **.Title** : donne ou définit le titre de la boîte de dialogue d'ouverture de fichier.
- **.Filter** : définit ou récupère le filtre de fichiers, spécifiant les types de fichiers à afficher lors de la sélection d'un fichier.
- **.ShowDialog()** : est une méthode qui affiche la boîte de dialogue.
- **.OK** : vérifier si l'utilisateur a appuyé sur le bouton "OK" dans la boîte de dialogue.
- **.FileName** : renvoie le chemin complet du fichier sélectionné.
- **.ImageLocation** : Il renvoie ou définit l'emplacement de l'image affichée.
- **CreatGraphics()** : crée et renvoie un objet Graphics associé au contrôle ou à la surface spécifié.
- **Bitmap** : est une classe dans l'espace de noms System.Drawing utilisée pour représenter une image bitmap. Elle offre des fonctionnalités pour manipuler et afficher des images au format bitmap (une grille de pixels utilisée pour représenter graphiquement des images).

- **.FromImage()** : crée un objet Graphics à partir d'un objet Bitmap, permettant d'effectuer des opérations de dessin sur l'image bitmap.
- **.CopyFromScreen()** : est une méthode permettant de copier une région spécifique de l'écran dans un objet Bitmap, facilitant ainsi la capture d'écran.
- **Location.X** : donne la coordonnée horizontale (abscisse) du coin supérieur gauche d'un contrôle par rapport à son conteneur parent.
- **Location.Y** : donne la coordonnée verticale (ordonnée) du coin supérieur gauche d'un contrôle par rapport à son conteneur parent.
- **Size** : représente la taille d'un contrôle, encapsulant sa largeur et sa hauteur dans un objet Size.
- **Size.Height** : renvoie la hauteur du contrôle encapsulée dans l'objet Size.
- **Size.Width** : renvoie la largeur du contrôle encapsulée dans l'objet Size.
- **e** : un objet de type PaintEventArgs passé en paramètre lors de l'événement de dessin (Paint) d'un contrôle.
- **.DrawImage()** : une méthode de l'objet Graphics permettant de dessiner une image sur une surface graphique

III. La structure du projet :

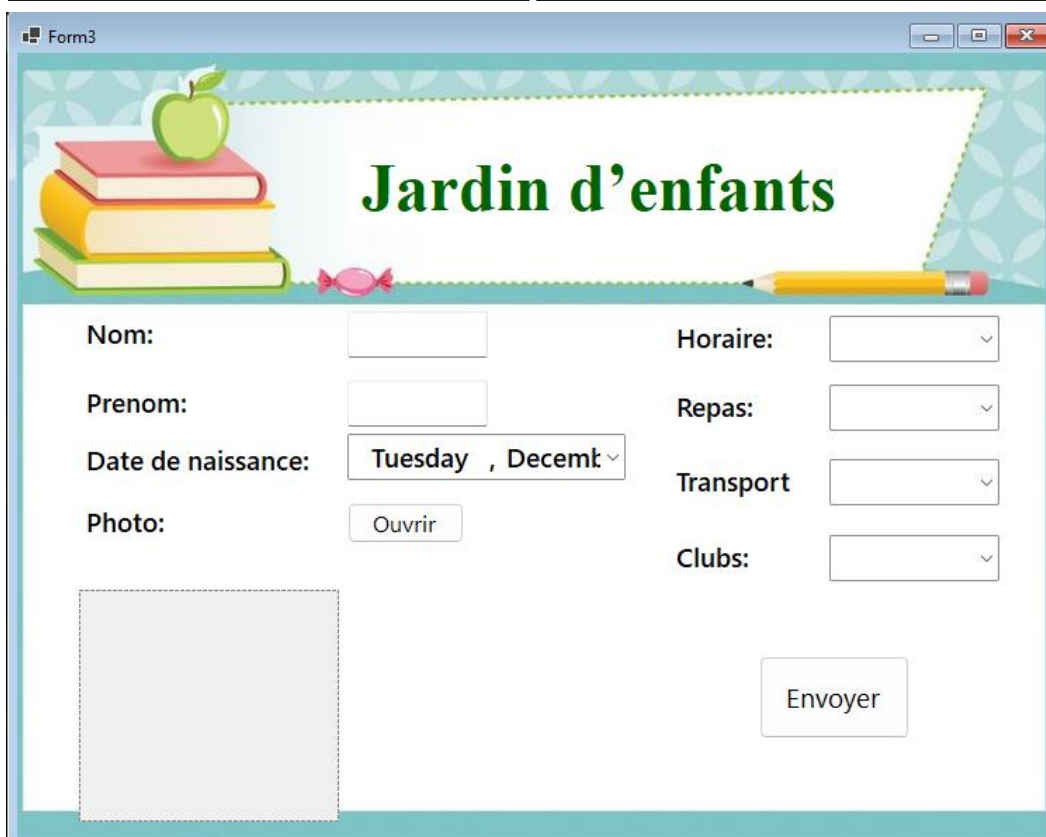
- Inscription :



Jardin d'enfants

Inscrire un nouvel enfant

Authentifier d'un enfant



Jardin d'enfants

Nom:

Prenom:

Date de naissance:

Photo:

Horaire:

Repas:

Transport:

Clubs:

- Authentification :



The screenshot shows a software window titled 'Form8'. The header has a decorative border with a stack of books and a green apple on the left, and a yellow pencil on the right. The title 'Jardin d'enfants' is centered in a green, stylized font. Below the header, there are two input fields: 'Nom:' and 'Prenom:'. A 'Verifier' button is located at the bottom right of the form area.

Nom:

Prenom:

Verifier

- Le paiement, Mise à jour, diplôme :



The screenshot shows a software window titled 'Form5'. It has the same decorative header as Form8, with the title 'Jardin d'enfants' in green. The main area contains three menu items, each with a corresponding button to its right: 'Mise à jour un fiche d'enfant' with a 'Mise a jour' button, 'Suivi de paiement' with a 'Paiment' button, and 'Une attestation de diplôme' with a 'Consulter' button.

Mise à jour un fiche d'enfant

Suivi de paiement

Une attestation de diplôme

Form9



Jardin d'enfants

Nom:

Prenom:

Date de naissance:

Photo:

Horaire:

Repas:

Transport:

Clubs:

Form7



Jardin d'enfants


Nom:

Prenom:

Form10

Diplome

Congratulations



Afficher

Imprimer

vous avez terminé la maternelle au [jardin d'enfant]
et recevez ce certificat en reconnaissance de vos réalisations

signé par : Timoumi Bassem, Essonni Hayfa, Neffati Mohammed

IV. Implémentation :

- Mise à jour des fiches (suppression):

```
Sub A1
    If test = True Then
        If File.Exists("SavedInformation.txt") = True Then
            Dim trouve As Boolean = False
            Dim lines As List(Of String) = File.ReadAllLines("SavedInformation.txt").ToList()
            For i As Integer = 0 To lines.Count - 1
                Dim parties As String() = lines(i).Split("*", StringSplitOptions.RemoveEmptyEntries)

                If parties.Length = 8 AndAlso parties(0) = nom AndAlso parties(1) = prenom Then
                    trouve = True
                    lines.RemoveAt(i)
                    Exit For
                End If
            Next

            If trouve Then
                Try
                    File.WriteAllLines("SavedInformation.txt", lines)
                    Dim Form9 As New Form9()

                    Me.Hide()

                    Form9.Show()
                Catch ex As Exception
                    MessageBox.Show($"Error: {ex.Message}", "Error")
                End Try
            Else
                MessageBox.Show("L'enfant n'est pas inscrit", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information)
            End If
        Else
            MessageBox.Show("Le fichier n'existe pas.", "Erreur")
        End If
    End If
End Sub
```

Ce segment de code permet de vérifier l'existence d'un enfant dans le système en parcourant les lignes d'un fichier. Il prend chaque ligne, la divise en parties, sachant que chaque enfant a des informations telles que le nom, le prénom, la photo, le type d'abonnement (journée complète ou demi-journée), la présence de repas, de transport et les clubs choisis. En se basant sur le nom et le prénom (parties 0 et 1 de la ligne), l'algorithme vérifie si l'enfant est trouvé. Si c'est le cas, il le supprime du fichier, actualise ce dernier, puis affiche une nouvelle interface.

Si l'enfant n'est pas trouvé, il affiche un message informant que l'enfant n'est pas inscrit. En cas d'erreur liée au fichier, un message d'erreur est également affiché. Ce processus permet de gérer efficacement les inscriptions et les modifications dans le système du jardin d'enfants.

```
Private Function IsValidName(name As String) As Boolean
    Return Not String.IsNullOrEmpty(name) AndAlso
        name.Length >= 3 AndAlso
        name.Length <= 20 AndAlso
        name.All(Function(c) Char.IsLetter(c)) AndAlso
        Not name.Contains(" ")
End Function
```

Cette fonction vise à vérifier la validité d'un nom ou d'un prénom. Elle retourne vrai si et seulement si la chaîne de caractères n'est pas vide, a une longueur comprise entre 3 et 20 caractères, est composée uniquement de lettres alphabétiques, et ne contient pas d'espaces. Ainsi, elle offre une validation rigoureuse pour s'assurer que le nom ou prénom respecte les critères spécifiés.

• Le montant :

```

If File.Exists("SavedInformation.txt") = True Then
    Dim trouve As Boolean = False
    Dim lignes As String() = File.ReadAllLines("SavedInformation.txt")
    For Each ligne As String In lignes
        parties = ligne.Split(";", StringSplitOptions.RemoveEmptyEntries)

        If parties.Length = 8 AndAlso parties(0) = nom AndAlso parties(1) = prenom Then
            trouve = True
            Exit For
        End If
    Next

    If trouve Then
        If parties(4) = "Journée complète" Then
            price = price + 20
        ElseIf parties(4) = "Demi-journée" Then
            price = price + 10
        End If

        If parties(5) = "Avec repas" Then
            price = price + 10
        ElseIf parties(5) = "Sans repas" Then
            price = price + 0
        End If

        If parties(6) = "Avec transport" Then
            price = price + 5
        ElseIf parties(6) = "Sans transport" Then
            price = price + 0
        End If

        If parties(7) = "Sport" Then
            price = price + 15
        ElseIf parties(7) = "Musique" Then
            price = price + 10
        ElseIf parties(7) = "Coran" Then
            price = price + 5
        ElseIf parties(7) = "Beaux Art" Then
            price = price + 10
        End If

        Label4.Text = price.ToString()
    Else
        MessageBox.Show("L'enfant n'est pas inscrit", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information)
    End If
Else
    MessageBox.Show("Le fichier n'existe pas.", "Erreur")
End If
End If

```

Cette partie du code effectue une recherche dans le fichier "SavedInformation.txt" pour trouver les informations associées à un enfant spécifique, identifié par son nom et son prénom. Si l'enfant est trouvé, les détails de son abonnement sont analysés, et un montant total est calculé en fonction des différentes

options sélectionnées, telles que le type d'abonnement (journée complète ou demi-journée), la présence de repas, de transport et les clubs auxquels l'enfant participe. Les montants associés à chaque option sont cumulés pour déterminer le coût total. Ce coût est ensuite affiché dans l'interface utilisateur, permettant au directeur du jardin d'enfants de visualiser rapidement le montant dû par chaque enfant en fonction de ses choix d'abonnement. Si l'enfant n'est pas trouvé dans le fichier, un message informatif est affiché, indiquant que l'enfant n'est pas inscrit. Des messages d'erreur sont également prévus en cas d'absence du fichier spécifié.

• Le paiement :

```
Dim price As Integer
If Label4.Text <> "" Then
    price = Label4.Text
End If

If File.Exists("payment.txt") = True Then
    Dim trouve As Boolean = False
    Dim lignes As String() = File.ReadAllLines("payment.txt")
    For Each ligne As String In lignes
        Dim parties As String() = ligne.Split("c", StringSplitOptions.RemoveEmptyEntries)

        If parties.Length = 3 AndAlso parties(0) = nom AndAlso parties(1) = prenom Then
            trouve = True
            Exit For
        End If
    Next

    If trouve Then
        MessageBox.Show("L'enfant déjà payé.", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information)
    Else
        If Label4.Text <> "" Then
            Try
                Dim fileName As String = "payment.txt"

                Using writer As New StreamWriter(fileName, True)
                    writer.WriteLine($"{nom}*{prenom}*{price}")
                End Using

                MessageBox.Show("Information enregistrer au fichier.", "Succès", MessageBoxButtons.OK, MessageBoxIcon.Information)
            Catch ex As Exception
                MessageBox.Show($"Error saving information to file: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
            End Try
        Else
            MessageBox.Show("Avoir le prix !", "Erreur", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End If
    End If
    Label4.Text = ""
Else
    MessageBox.Show("Le fichier n'existe pas.", "Erreur")
End If
```

Ce segment de code vise à vérifier si un enfant a déjà été payé en examinant le contenu du fichier "payment.txt", qui stocke les informations sur les enfants ayant effectué des paiements. Il parcourt les lignes du fichier pour rechercher l'enfant spécifié par son nom et prénom.

Si l'enfant est trouvé, un message informatif est affiché indiquant que l'enfant a déjà été payé. Dans le cas contraire, le code vérifie si le montant total à payer (affiché dans l'interface utilisateur sous "Label4") n'est pas vide.

Si c'est le cas, les informations de paiement sont enregistrées dans le fichier, indiquant le nom, prénom et montant payé. Un message de confirmation est alors affiché. Si le fichier "payment.txt" n'existe pas, un message d'erreur approprié est affiché.

Cette logique permet au directeur du jardin d'enfant de suivre les paiements des parents et d'enregistrer ces informations de manière ordonnée pour une gestion efficace des finances.

• Inscription de l'enfant :

```

If IsValidName(prenom) = False Then
    test = False
    MessageBox.Show("Veuillez saisir un prenom valide (alphabétique, sans espaces, maximum 20 lettres et minimum 3).", "Invalid Input", MessageBoxButtons.OK, MessageBoxIcon.Error)
End If

If TextBoxFilePath.Text = "" Then
    test = False
    MessageBox.Show("Veuillez choisir une photo.", "Invalid photo", MessageBoxButtons.OK, MessageBoxIcon.Error)
End If

Dim selectedCategory1 As String
Dim selectedCategory2 As String
Dim selectedCategory3 As String
Dim selectedCategory4 As String
If ComboBox1.SelectedIndex <> -1 And ComboBox2.SelectedIndex <> -1 And ComboBox3.SelectedIndex <> -1 And ComboBox4.SelectedIndex <> -1 Then

    selectedCategory1 = ComboBox1.SelectedItem.ToString()
    selectedCategory2 = ComboBox2.SelectedItem.ToString()
    selectedCategory3 = ComboBox3.SelectedItem.ToString()
    selectedCategory4 = ComboBox4.SelectedItem.ToString()

Else
    test = False
    MessageBox.Show("Veuillez sélectionner une catégorie dans la ComboBox.", "Invalid Selection", MessageBoxButtons.OK, MessageBoxIcon.Error)
End If

Dim SelectedDate As DateTime

    MessageBox.Show("Veuillez sélectionner une catégorie dans la ComboBox.", "Invalid Selection", MessageBoxButtons.OK, MessageBoxIcon.Error)
End If
Dim SelectedDate As DateTime

SelectedDate = DateTimePicker1.Value
Dim formattedDate As String = SelectedDate.ToString("yyyy-MM-dd")
If test = True Then
    Try
        Dim fileName As String = "SavedInformation.txt"

        Using writer As New StreamWriter(fileName, True)
            writer.WriteLine($"{nom}*{prenom}*{formattedDate}*{selectedFilePath}*{selectedCategory1}*{selectedCategory2}*{selectedCategory3}*{selectedCategory4}")
        End Using

        MessageBox.Show("Information saved to file.", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information)
    Catch ex As Exception
        MessageBox.Show($"Error saving information to file: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End If
Sub

```

Ce bloc de code gère la sauvegarde des informations relatives à un enfant dans un fichier nommé « SavedInformation », après avoir vérifié la validité de chaque entrée. Il commence par extraire les données saisies, telles que le nom, le prénom, la date de naissance, le chemin d'accès à la photo, et les différentes catégories d'abonnement sélectionnées via des contrôles de saisie appropriés. Ensuite, il vérifie la validité de ces informations 'isValidName'. De plus, il vérifie si une photo a été sélectionnée et si toutes les catégories

d'abonnement ont été choisies dans les ComboBox. Si toutes ces vérifications réussissent, les données sont enregistrées dans le fichier "SavedInformation.txt" sous un format structuré. En cas d'erreur, un message d'erreur approprié est affiché pour informer l'utilisateur du problème.

```
Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click

    Dim openFileDialog As New OpenFileDialog()

    openFileDialog.Title = "Select a Photo"
    openFileDialog.Filter = "Image Files|*.jpg;*.jpeg;*.png|All Files|*.*"

    If openFileDialog.ShowDialog() = DialogResult.OK Then

        selectedFilePath = openFileDialog.FileName

        TextBoxFilePath.Text = selectedFilePath
        PictureBox1.ImageLocation = selectedFilePath
    End If

End Sub
```

Ce code met en place une fonctionnalité de sélection de photo via une boîte de dialogue OpenFileDialog. Il configure les filtres pour n'accepter que les fichiers image (.jpg, .jpeg, .png) et récupère le chemin du fichier sélectionné. Ce chemin est ensuite affiché dans un champ de texte et l'image associée est chargée dans un PictureBox, offrant une interface utilisateur conviviale pour choisir une photo dans l'application de gestion du jardin d'enfants.

- **Imprimer une attestation de diplôme pour chaque enfant :**

```
Private Sub PrintPreviewDialog1_Click(sender As Object, e As EventArgs) Handles PrintPreviewDialog1.Click
    g = CreateGraphics()
    bmp = New Bitmap(Size.Width, Size.Height, g)
    mg = Graphics.FromImage(bmp)
    mg.CopyFromScreen(Location.X, Location.Y, 0, 0, Size)
    PrintPreviewDialog2.ShowDialog()
End Sub

Private Sub PrintDocument1_PrintPage(sender As Object, e As Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
    e.Graphics.DrawImage(bmp, 0, 0)
End Sub
```

Ce code est associé à l'impression d'une prévisualisation dans une application de gestion de jardin d'enfants. Lorsque l'utilisateur clique sur une boîte de dialogue de prévisualisation d'impression (PrintPreviewDialog1), le programme capture une capture d'écran de la fenêtre principale de l'application. Cette capture d'écran est stockée dans un objet Bitmap (bmp). Lorsque l'utilisateur imprime, l'image stockée dans bmp est dessinée sur la page d'impression (PrintDocument1_PrintPage). La boîte de dialogue d'impression (PrintPreviewDialog2) s'ouvre ensuite pour permettre à l'utilisateur de visualiser et d'imprimer la capture d'écran de la fenêtre principale.

● Authentification :

```

Line 41
If test = True Then

    If File.Exists("SavedInformation.txt") = True Then
        Dim trouve As Boolean = False
        Dim lignes As String() = File.ReadAllLines("SavedInformation.txt")
        For Each ligne As String In lignes
            Dim parties As String() = ligne.Split("**c", StringSplitOptions.RemoveEmptyEntries)

            If parties.Length = 8 AndAlso parties(0) = nom AndAlso parties(1) = prenom Then
                trouve = True
                Exit For
            End If
        Next

        If trouve Then
            MessageBox.Show("L'enfant déjà inscrit.", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information)
        Else
            MessageBox.Show("L'enfant n'est pas inscrit", "Information", MessageBoxButtons.OK, MessageBoxIcon.Information)
        End If
    Else
        MessageBox.Show("Le fichier n'existe pas.", "Erreur")
    End If
End If
Sub
```

Ce segment de code permet de vérifier l'existence d'un enfant dans le système en parcourant les lignes d'un fichier. Il prend chaque ligne, la divise en parties, sachant que chaque enfant a des informations telles que le nom, le prénom, la photo, le type d'abonnement (journée complète ou demi-journée), la présence de repas, de transport et les clubs choisis. En se basant sur le nom et le prénom (parties 0 et 1 de la ligne), l'algorithme vérifie si l'enfant est trouvé. Si l'enfant n'est pas trouvé, il affiche un message informant que l'enfant n'est pas inscrit. En cas d'erreur liée au fichier, un message d'erreur est également affiché.

Références

Visual Studio

IDE le plus complet pour les développeurs .NET et C++ sur Windows. Entièrement rempli d'un bon ensemble d'outils et de fonctionnalités permettant d'élever et d'améliorer chaque étape du développement de logiciels.