# Projet TConsole

Fournit des outils pour applications console.

Un diagramme UML de classes est fourni dans le même répertoire que le présent document.

## ConsoleApplication

Classe abstraite qui modélise une application console avec des pages accessibles via des menus, sur plusieurs niveaux. Gère un dictionnaire de Page en interne

## Page

Classe abstraite qui modélise une page de l’application avec un titre, sans menu

## MenuPage

Classe abstraite qui dérive de Page et ajoute un menu à la page.

La propriété statique **DefaultOptions** permet de définir des entrées de menu par défaut qui seront automatiquement ajoutées à toutes les pages créées à partir de cette classe.

## Menu

Modélise un menu. Gère en interne un dictionnaire pour les entrées de menus. Ce dictionnaire est consultable dans une version en lecture seule.

On peut ajouter des entrées au menu (classe option) au moyen des deux méthodes surchargées AddOption.

La méthode Prompt affiche le menu complet, et lance l’action correspondant au menu choisi par l’utilisateur.

## Option

Modélise une entrée de menu, avec un identifiant, un libellé et une action à exécuter lorsqu’on sélectionne cette entrée. L’identifiant est une chaîne de caractère qui est affichée avant le libellé et qui correspond à ce que l’utilisateur doit saisir pour sélectionner cette entrée.

## Input et Output

Classes statiques qui offrent quelques méthodes pour faciliter la validation de la saisie utilisateur, et l’affichage avec des couleurs et chaînes de formats.

La méthode générique Read demande la saisie d'une donnée du type T. Si la chaîne saisie ne peut pas être convertie dans ce type, un message d'erreur est affiché. On peut spécifier une valeur par défaut à afficher, avant la saisie de l’utilisateur. Ceci est utile par exemple lorsqu’on propose à l’utilisateur de modifier une valeur existante.

## ConsoleCard

Affiche des données sous forme de fiche comportant des paires libellé – valeur

La méthode statique générique **From** crée une fiche à partir d'un objet passé en paramètre. Chaque ligne de la fiche est composée du libellé et de la valeur d'une propriété de l'objet. Le libellé correspond à la valeur de l'attribut Display s’il est renseigné, ou au nom de la propriété sinon.  
La méthode retourne la fiche créée (objet de type ConsoleCard)

La méthode **Display** affiche la fiche avec ou sans les libellés. On peut également préciser un titre à afficher au-dessus de la fiche.

## ConsoleTable

Affiche les données d’une collection sous forme tabulaire.

Il y a deux façons de créer un tableau :

* En utilisant le constructeur auquel on passe la liste des colonnes du tableau. Il faut ensuite ajouter les lignes une par une en appelant la méthode AddRow
* En appelant la méthode From

La méthode **AddRow** permet d’ajouter une ligne à l’instance de tableau courante. Les colonnes du tableau doivent au préalable être définies.  
Les valeurs de la ligne sont passées sous forme d’un tableau d’objets dont le nombre d’éléments doit correspondre au nombre de colonnes du tableau.

La méthode statique générique **From** crée un tableau à partir d'une liste d'items. Si les items sont de type valeur ou string, le libellé de la colonne doit être spécifié. Sinon, les libellés sont déduits des propriétés du type de l'item (en reprenant leur attribut Display ou leur nom).  
La méthode retourne le tableau créé (objet de type ConsoleTable)

La méthode **Display** affiche le tableau selon le format spécifié. On peut afficher au-dessus, un libellé avec le nombre d’items.