CLASSE :

Program :

* Start.LancementQuizz() ;

Start :

* Void LancementQuiz : lance le programme du quiz ;
  + - User user = new User() ;
    - Identification() ;
    - Choix()
* Void Identification : user donne son username, s’il n’existe pas le crée dans la base de donné et lance la méthode user.CreationUser(username). S’il existe lance la méthode user.Connection(username)
* Void Choix() : demande ce que veut faire l’utilisateur : voir ses quizz , User.ShowQuizz(). Crée un quizz si user.isCreateQuizz = true, User.CreateQuizz(). Voir ses données User.ShowData()

user :

list<quizz>

* Void Creation(string username) user : crée un nouveau user demande les paramètre user et les rajoute dans la bdd
* Void Connection (string username) : définie les donnée de l’user comme étant égale a celui existant dans la base de donnée avec l’username « username »
* Void showQuizz() : montre la liste de quizz que l’utilisateur possède sous forme de liste et demande à l’utilisateur s’il veut voir un quizz en particulier Get l’id et quizz.showThisQuiz(id). S’il ne veut pas on retourne à Start.choix()
* Void CreateQuizz : (instancie un quizz) ajoute un quizz, demande le nom du quiz et la catégorie, quizz.selectCatégorie(), demande le nombre de question qu’il veut (max n), demande la question, la reponse \* n . et ajoute dans la base de donnée. Start.choix()
* showData() : = ToString user : affiche toutes les données du user et remplace le mdp par des \*\*\*\*, Start.choix().

Quiz :

List<theme> : égale a toute les isntanciation de quizz

Public quizz : titre theme level , solution, list<theme>

Contructeur ={ this list theme = get all list theme bdd]}

List<solution>

* Int SelectCatégorie : affiche les catégories disponible sous forme de tableau getAllTheme , et demande à l’utilisateur la catégorie de son quiz, recupere l’id du theme et le retourne.
* Void showThisQuiz(User user, int idQuiz) : montre les question et repose du quizz et rappelle la méthode user.showQuizz() ;

Solution :

Id

Nom