

Travail pratique 1 : Javascript et POO

Note : travail doit se faire individuellement ou en une équipe de deux maximums

Énoncé :

L'objectif de ce travail est de permettre à l'étudiant de travailler avec le JavaScript pour faire de la POO. Comme objectifs secondaires on vise l'apprentissage et la mise en pratique des fonctions de base du DOM, d'événements simples et d'affichage dynamique d'une page Web en utilisant JavaScript.

Travail à faire

Dans un fichier que vous nommerez classeEtudiant_tp1.js vous devez écrire une classe Etudiant qui possède les attributs (propriétés) suivants:

- id (une chaîne)
- nom (une chaîne)
- prenom (une chaîne)
- codePermanent (une chaîne)
- surForm (un booléen)
- noteTP1 (un nombre entre 0 et 20), noteTP2 (un nombre entre 0 et 20)
- noteExamIntra, noteExamFinal (deux nombres entre 0 et 100)
- noteTravaux (un nombre - note intermédiaire calculée par votre programme en additionnant les noteTP1, noteTP2)
- noteExamens (un nombre - note intermédiaire calculée par votre programme: $\text{noteExamIntra} \times 0.2 + \text{noteExamFinal} \times 0.4$)
- noteGlobale (un nombre - note calculée par votre programme si le seuil de 23.5 aux examens est atteint en additionnant noteTravaux et noteExamens; si le seuil n'est pas atteint seulement la note des examens est utilisée comme noteFinale)
- noteLettre (une chaîne calculée par votre programme en fonction de la valeur de noteFinale).
 - A+ si la noteFinale est ≥ 89.5
 - A si la noteFinale est ≥ 84.5
 - A- si la noteFinale est ≥ 79.5
 - B+ si la noteFinale est ≥ 76.5
 - B si la noteFinale est ≥ 72.5

- B- si la noteFinale est ≥ 69.5
- C+ si la noteFinale est ≥ 64.5
- C si la noteFinale est ≥ 59.5
- C- si la noteFinale est ≥ 56.5
- D+ si la noteFinale est ≥ 53.5
- D si la noteFinale est ≥ 49.5
- E sinon.

Ensuite vous ajoutez un constructeur de l'objet Etudiant, qui reçoit en paramètre l'identifiant (id) de l'étudiant. Le constructeur place l'identifiant dans l'attribut id et initialise les nom, prenom, codePermanent et noteLettre avec une chaîne vide, le booléen surForm à false et tous les autres attributs à 0. Ajoutez les méthodes suivantes:

- **construireForm()**: qui permet de construire un formulaire pour afficher les attributs de l'étudiant (le formulaire est retourné dans une chaîne de caractères). N'utilisez pas de balise <form>, elle sera insérée directement en html. Tous les attributs à saisir sont dans des <input type="text"...> alors que les valeurs calculées sont dans des . Placez un <fieldset> autour des balises de ce formulaire afin d'encadrer l'étudiant (utilisez le id comme légende au <fieldset>). Finalement la méthode met true dans le booléen surForm.
- **formDejaAffiche()**: qui retourne la valeur du booléen surForm. Sera utile pour déterminer si le formulaire a déjà été construit pour cet objet.
- **validerForm()**: qui permet de vérifier les valeurs du formulaire. Cette méthode retourne true si les champs du formulaire sont valides. Pour valider les champs du formulaire vous devez vérifier seulement les nombres qui doivent être dans l'intervalle respectif mentionné ci-dessus.
- **lireForm()**: qui permet de récupérer les valeurs des attributs de l'étudiant (et de les mettre dans l'objet)

- **calculerNotes()**: qui permet de calculer les notes intermédiaires, la note finale et la noteLettre de l'étudiant
- **afficherNotes()**: qui permet de placer les notes intermédiaires, la note finale et la noteLettre directement sur le formulaire (avec innerHTML directement dans les). Attention affichez seulement une décimale après le point (utilisez toFixed).

Ajoutez la propriété (attribut) de classe instances qui permet de comptabiliser le nombre d'instances créées. Ajoutez aussi la méthode de classe nbInstances qui permet de retourner la valeur de la propriété de classe instances. Modifiez le constructeur pour incrémenter la propriété de classe instances lors de l'instanciation de l'objet.

Pour tester votre classe Etudiant vous devez obligatoirement utiliser le fichier suivant [tp1_test.html](#)