|  |
| --- |
| **Nom :**  **Prénom :** |
| **Consignes :**   * **Travail en monôme**😉 * **Ecrivez votre réponse en dessous de chaque exercice. Si votre code est réparti sur plusieurs fichiers, écrivez le nom du fichier au-dessus de votre code.** * **Pensez à commenter votre code.** * **Une fois tous les exercices terminés, n’oubliez pas de remettre votre devoir.** * **Travail évalué. Des points faciles à prendre … Alors jouez le jeu de l’objet !** * **Pour ceux qui le souhaite, des rappels sur le Java seront donnés en cours.** * **Enjoy !** |

**Exercice 1 : First Class in Action !**

**Exercice 2 : Constructeurs**

**Exercice 3 : Héritage**

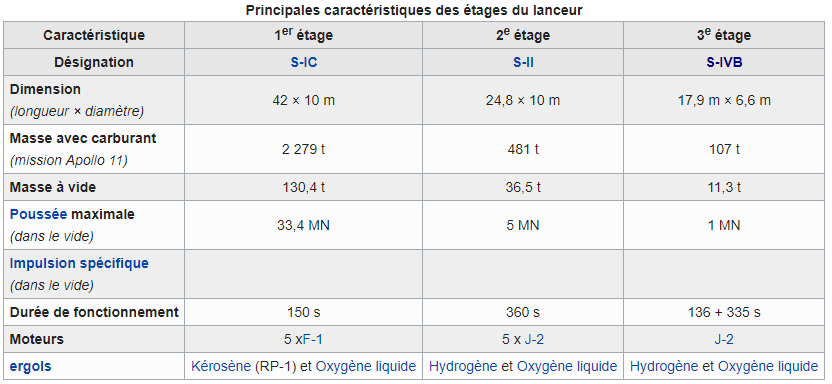
**Exercice 4 : Héritage et constructeur**

**Exercice 5 : Encapsulation par la pratique**

**Exercice 6 : Polymorphisme**

**Exercice 7 : Tableau polymorphe**

**Exercice 1 : First Class in Action !**

* **Objectif : Faire décoller des fusées !**
* Ecrire une classe « Fusee ».
  + Caractéristiques d’une fusée
    - Nom
    - Masse à vide (en kg)
    - Poussée (en N)
    - Quantité de carburant (en l en considérant 1l = 1kg)
    - Consommation de carburant (en l/s)
    - Position (altitude en m)
  + Méthode « decollage » retournant la position.
    - Calcul 🡺 (poussee\*duree\*duree\*100)/(masse+carburant)
* Créer une fusée « SATURN V ».
* Afficher toutes les caractéristiques de la fusée « SATURN V ».
* Faites décoller la Fusée « SATURN V ».

Exemple de fusée : SATURN V

**Exercice 2 : Constructeurs**

* Reprendre le programme de l’exercice 1.
  + Ecrire 3 constructeurs
    - Constructeur vide.
    - Constructeur prenant en paramètres toutes les caractéristiques.
    - Constructeur qui prend un objet de type « Fusee » en paramètre et initialise les variables à l’aide du constructeur précédent.
  + Ecrire une méthode pour afficher les caractéristiques d’une fusée.
  + Créer une fusée « SATURN V ».
  + Afficher toutes les caractéristiques de la fusée « SATURN V ».
  + Faites décoller la Fusée « SATURN V ».
  + Créer une Fusée « SATURN V2 » reprenant les mêmes caractéristiques que la « SATURN V ».
  + Afficher toutes les caractéristiques de la fusée « SATURN V2 ».

**Exercice 3 : Héritage**

* Ecrire une classe FCargo qui hérite de la classe « Fusee » et embarque un chargement.
* Ecrire une classe FAstronaute dérivée de la classe « Fusee » et embarque un équipage.
* Pour ces classes, ajouter 2 constructeurs.
  + Constructeur vide.
  + Constructeur prenant en paramètres toutes les caractéristiques.
* Créer une « SATURN V » avec 3 membres d’équipage.
* Créer une « SATURN V2 » embarquant un satellite de 3 t.
* Faites décoller les 2 fusées et afficher les caractéristiques.
* NB : ne pas oublier les méthodes pour charger de la marchandise ou pour embarquer un équipage.

**Exercice 4 : Héritage et constructeur**

* Ajouter des affichages d’appel dans les constructeurs des classes Fusee, FCargo et FAstronaute.
  + Par exemple :
    - System.out.println("Appel du constructeur FCargo())");
    - System.out.println("Appel du constructeur Fusee(nom, masse, …))");
* Que constatez-vous ?

**Exercice 5 : Encapsulation par la pratique**

* Trouver les erreurs.
* Corriger le code.
* Expliquer la différence entre « public », « protected » et « private ».

public class B extends A

{

public int pubb;

protected int prob;

private int prib;

public void test()

{

pria=1;

}

}

public class Encapsulation

{

public static void main(String []args)

{

A monA=new A();

B monB=new B();

monA.puba=1;

monA.proa=2;

monA.pria=3;

monB.puba=1;

monB.proa=2;

monB.pria=3;

monA.Affiche();

}

}

public class A

{

public int puba;

protected int proa;

private int pria;

public void Affiche()

{}

}

**Exercice 6 : boucle**

**Exercice 6 : Polymorphisme**

* Reprendre le programme de l’exercice 4.
* Spécialiser la méthode « decollage » dans les classes « FCargo » et « FAstronaute » en tenant compte des caractéristiques spéciales de chaque classe.
* Faites décoller les 2 fusées et affichez leur altitude respective.

**Exercice 7 : Tableau polymorphe**

* Créer un seul tableau contenant 5 « FCargo » et 5 « FAstronaute ».
* A l’aide d’une seule boucle « pour »
  + Faire décoller les « FCargo » et « FAstronaute ».
  + Afficher leur altitude.

*Pour aller plus loin, demandez la suite à votre professeur préféré !*