Anna STEINMETZ

Moyenne standard (Pondérée par les dossiers) : 8.04 / 20	Moyenne pondérée (Pondérée par les questions uniquement) : 8.04 / 20

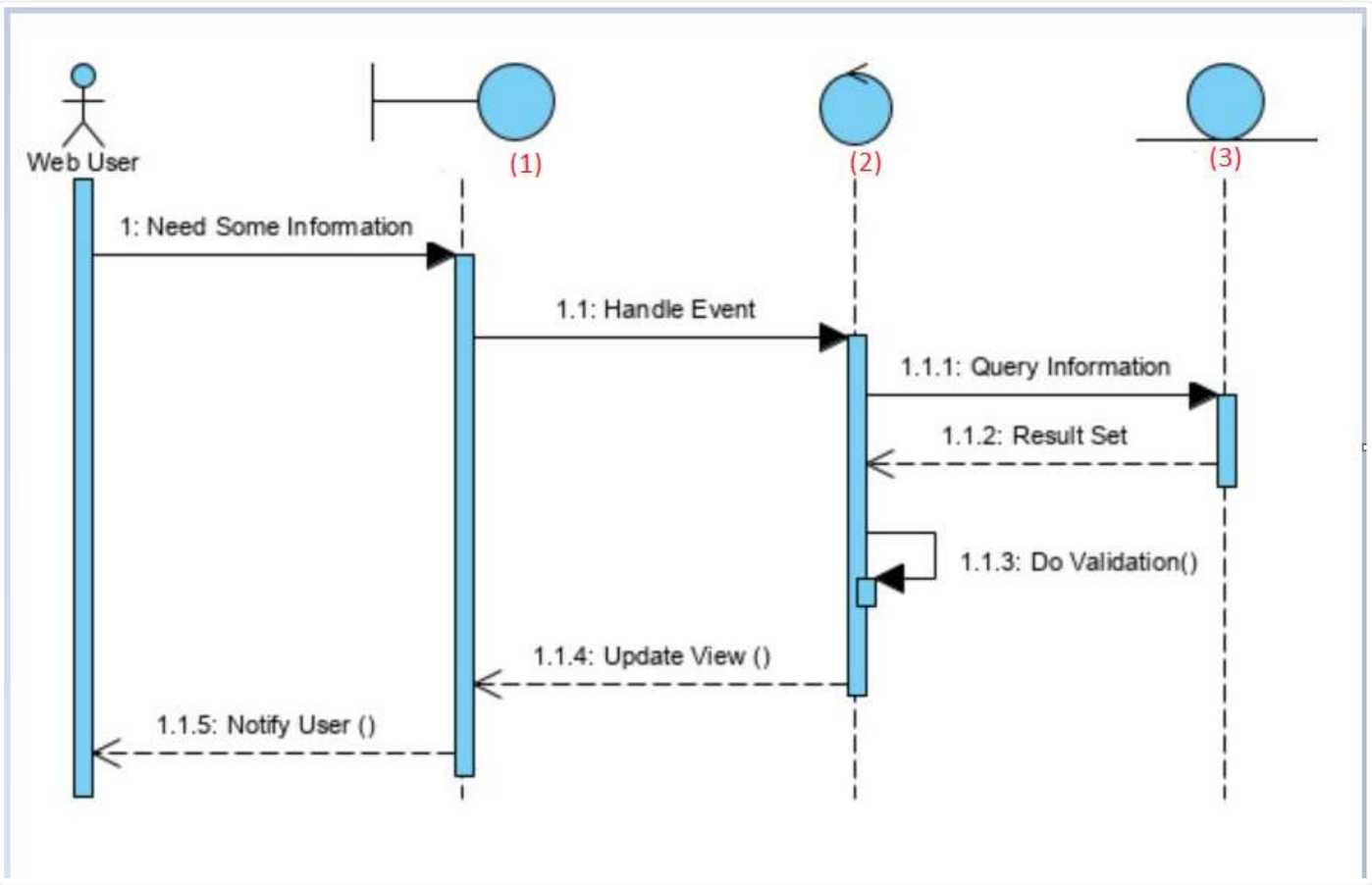
EN - 2022-2023_Architectures logicielles et modélisation UML_FISE_Initial

Note du dossier : 8.04 / 20

Question 1 **Association**

Compléter le diagramme ci-dessous

Complete the diagram below



Réponses correctes	0 discordance	1 point obtenu sur 1

Élément à associer	Réponse attendue	Réponse saisie	Réponse discordante
(1)	Vue *View*	Vue *View*	Non
(2)	Contrôleur *Controller*	Contrôleur *Controller*	Non
(3)	Modèle *Model*	Modèle *Model*	Non

Question 2 — **Question à réponses multiples**

Quels diagrammes ne sont pas des diagrammes UML (2 réponses)

Which diagrams are not UML diagrams (2 answers)

Réponses partiellement correctes	2 discordances	0.3714 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☐	☐	Non	Diagramme de cas d'utilisation *Use case diagram*
B	☐	☐	Non	Diagramme de séquence *Sequence diagram*
C	☐	☑	Oui (+1)	Diagramme des composants *Component diagram*
D	☐	☐	Non	Diagramme de classe *Class diagram*
E	☐	☐	Non	Diagramme d'activité *Activity diagram*
F	☑	☑	Non	Diagramme de Gantt *Gantt diagram*
G	☑	☐	Oui (+1)	Diagramme de pieuvre *Octopus diagram*

Question 3 **Question à réponse unique**

Identifiez la MAUVAISE proposition

Identify the WRONG statement

Réponses correctes	1 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☐	☐	Non	Le design pattern Modèle-Vue-Contrôleur (MVC) est un pattern architectural qui sépare les données (le modèle), l'interface homme-machine (la vue) et la logique de contrôle (le contrôleur). *The Model-View-Controller (MVC) design pattern is an architectural pattern that separates the data (the model), the man-machine interface (the view) and the control logic (the controller).*
B	☐	☐	Non	Le contrôleur : il gère l'interface entre le modèle et le client. Il va interpréter la requête de ce dernier pour lui envoyer la vue correspondante. Il effectue la synchronisation entre le modèle et les vues/ *The controller: it manages the interface between the model and the client. It interprets the client's query and sends it the corresponding view. It performs the synchronization between the model and the views.*
C	☐	☐	Non	La vue : elle représente l'interface utilisateur, ce avec quoi il interagit. Elle n'effectue aucun traitement, elle se contente d'afficher les données que lui fournit le modèle. Il peut tout à fait y avoir plusieurs vues qui présentent les données d'un même modèle. *The view: it represents the user interface, what he/she interacts with. It does not perform any processing, it simply displays the data provided by the model. There can be several views that present the data of the same model.*
D	☑	☑	Non	Le modèle : il représente les méthodes de l'application. *The model: it represents the methods of the application.*
E	☐	☐	Non	La synchronisation entre la vue et le modèle se passe avec le pattern Observer. Il permet de générer des événements lors d'une modification du modèle et d'indiquer à la vue qu'il faut se mettre à jour *The synchronization between the view and the model is done with the Observer pattern. It allows to generate events when the model is modified and to indicate to the view that it must be updated*

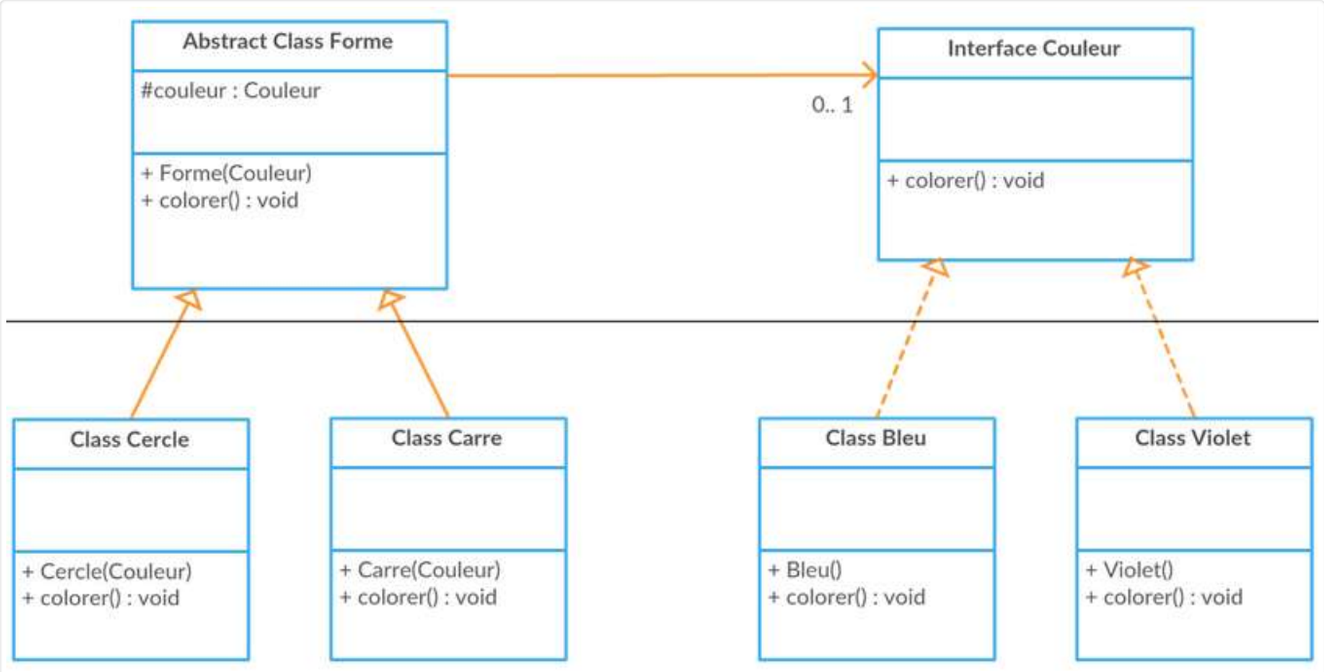
Question 4 **Question à réponse unique**

Observez le diagramme de classes suivant :

Quel Design Pattern implémente-t-il ? (1 réponse)

Look at the following class diagram:

Which Design Pattern does it implement? (1 answer)



Réponses correctes	1 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☐	☐	Non	OBSERVER
B	☐	☐	Non	BUILDER
C	☑	☑	Non	BRIDGE
D	☐	☐	Non	FACTORY
E	☐	☐	Non	STRATEGY

Question 5 **Question à réponse unique**

Quelle est la principale différence entre les modèles d'architectures logicielles MVC et MVVM ? (1 réponse)

What is the main difference between the MVC and MVVM software architecture models? (1 answer)

Réponses incorrectes	0 point obtenu sur 1

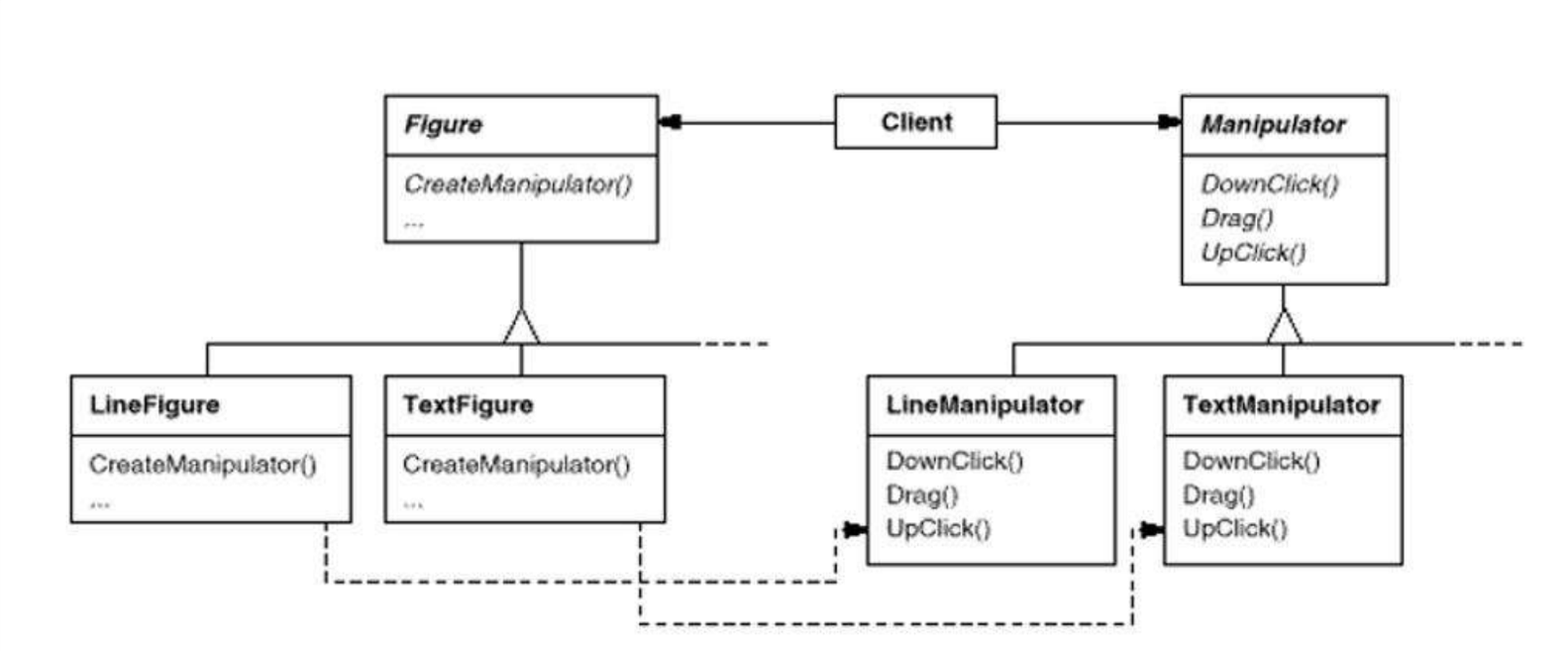
	Réponse attendue	Réponse saisie	Réponse discordante	
A	☑	☐	Oui (+1)	MVVM est utilisé pour *lier les données* de *manière bidirectionnelle* dans les vues. Il s'agit généralement d'une implémentation côté client. MVC est un moyen de séparer les problèmes *côté serveur.* *MVVM is used to link data bidirectionally in views. It is usually a client-side implementation. MVC is a way to separate server side issues.*
B	☐	☐	Non	MVVM est utilisé pour les applications développées sous .NET Core tandis que MVC est utilisé dans .NET framework. *MVVM is used for applications developed under .NET Core while MVC is used in .NET framework.*
C	☐	☐	Non	Il n'y a pas de différence. MVVM est une évolution de MVC. *There is no difference. MVVM is an evolution of MVC.*
D	☐	☐	Non	MVVM est un modèle d'architecture d'application multiplateformes. MVC est un modèle d'architecture dédié à Windows 10. *MVVM is a cross-platform application architecture model. MVC is an architecture model dedicated to Windows 10.*
E	☐	☑	Oui (+1)	MVC est une modèle d'architecture d'applications en mode console. MVVM est un modèle d'architecture d'applications en mode graphique. *MVC is a model of application architecture in console mode. MVVM is a model of application architecture in graphical mode.*

Question 6

Question à réponse unique

Observez le diagramme de classes suivant :

Look at the following class diagram:



Quel Design Pattern implémente-t-il ? (1 réponse)

Which Design Pattern does it implement? (1 answer)

Réponses incorrectes	0 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☐	☐	Non	OBSERVER
B	☐	☑	Oui (+1)	BUILDER
C	☐	☐	Non	BRIDGE
D	☑	☐	Oui (+1)	FACTORY
E	☐	☐	Non	STRATEGY

Question 7

Question à réponses multiples

Parmi les possibilités suivantes, quelles sont celles qui correspondent à des types de diagrammes UML ? (3 réponses)

Which of the following possibilities correspond to UML diagram types? (3 answers)

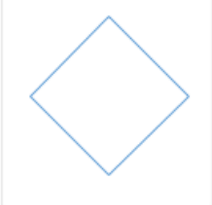
Réponses correctes	0 discordance	1 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☑	☑	Non	Séquences *Sequences*
B	☐	☐	Non	Cas critiques *Critical cases*
C	☑	☑	Non	Activités *Activities*
D	☑	☑	Non	Cas d'utilisation *Use cases*
E	☐	☐	Non	Cas d'erreurs *Error cases*

Question 8 — Association

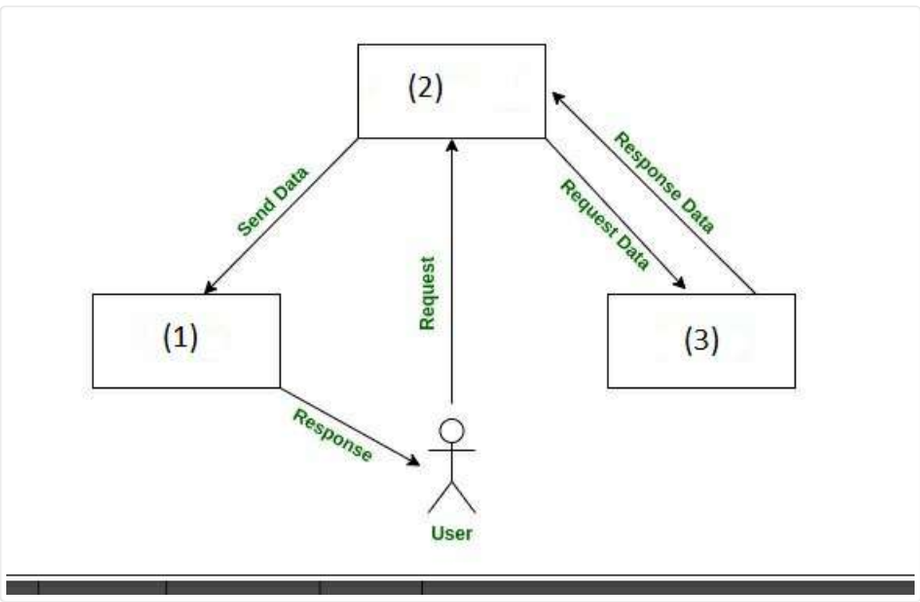
Dans un diagramme d'activité, quelle est la signification de chaque symbole ?
In an activity diagram, what is the meaning of each symbol?

Réponses correctes	0 discordance	1 point obtenu sur 1

Élément à associer	Réponse attendue	Réponse saisie	Réponse discordante
	Début / Nœud initial *Start / Initial node*	Début / Nœud initial *Start / Initial node*	Non
	Action *Action*	Action *Action*	Non
	Décision *Decision*	Décision *Decision*	Non
	Commentaire *Comment*	Commentaire *Comment*	Non
	Nœud final *Final node*	Nœud final *Final node*	Non

Question 9 — Association

Dans le modèle d'architecture suivant,
Quel est le nom de chacun des 3 composants



Réponses incorrectes	3 discordances	0 point obtenu sur 1

Élément à associer	Réponse attendue	Réponse saisie	Réponse discordante
(1) Aucun fichier n'a été déposé	Vue *View*	Modèle-Vue *Model-View*	Oui (+1)
(2)	Contrôleur *Controller*	Vue *View*	Oui (+1)
(3)	Modèle *Model*	Vue-Modèle *View-Model*	Oui (+1)

Question 10 **Question à réponse unique**

Observez le code source suivant qui implémente un système de restitution des bagages d'un aéroport.
Une classe BaggageInfo fournit des informations sur les vols arrivés et sur les tapis roulants où les bagages de chaque vol peuvent être récupérés.

Look at the following source code that implements an airport baggage retrieval system.
A BaggageInfo class provides information about the arriving flights and the conveyor belts where the luggage of each flight can be retrieved.

```
using System;
using System.Collections.Generic;

public class BaggageInfo
{
   private int flightNo;
   private string origin;
   private int location;

   internal BaggageInfo(int flight, string from, int carousel)
   {
      this.flightNo = flight;
      this.origin = from;
      this.location = carousel;
   }

   public int FlightNumber {
      get { return this.flightNo; }
   }

   public string From {
      get { return this.origin; }
   }

   public int Carousel {
      get { return this.location; }
   }
}
```

Sur quel Design Pattern cette implémentation se base-t-elle ? (1 réponse)
On which Design Pattern is this implementation based? (1 answer)

Réponses correctes	1 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☑	☑	Non	OBSERVER
B	☐	☐	Non	BUILDER
C	☐	☐	Non	BRIDGE
D	☐	☐	Non	FACTORY
E	☐	☐	Non	STRATEGY

Question 11 — **Question à réponses multiples**

Parmi les propositions suivantes, lesquelles permettent de créer un processus et le lancer (2 réponses)

Which of the following can be used to create a process and launch it (2 answers)

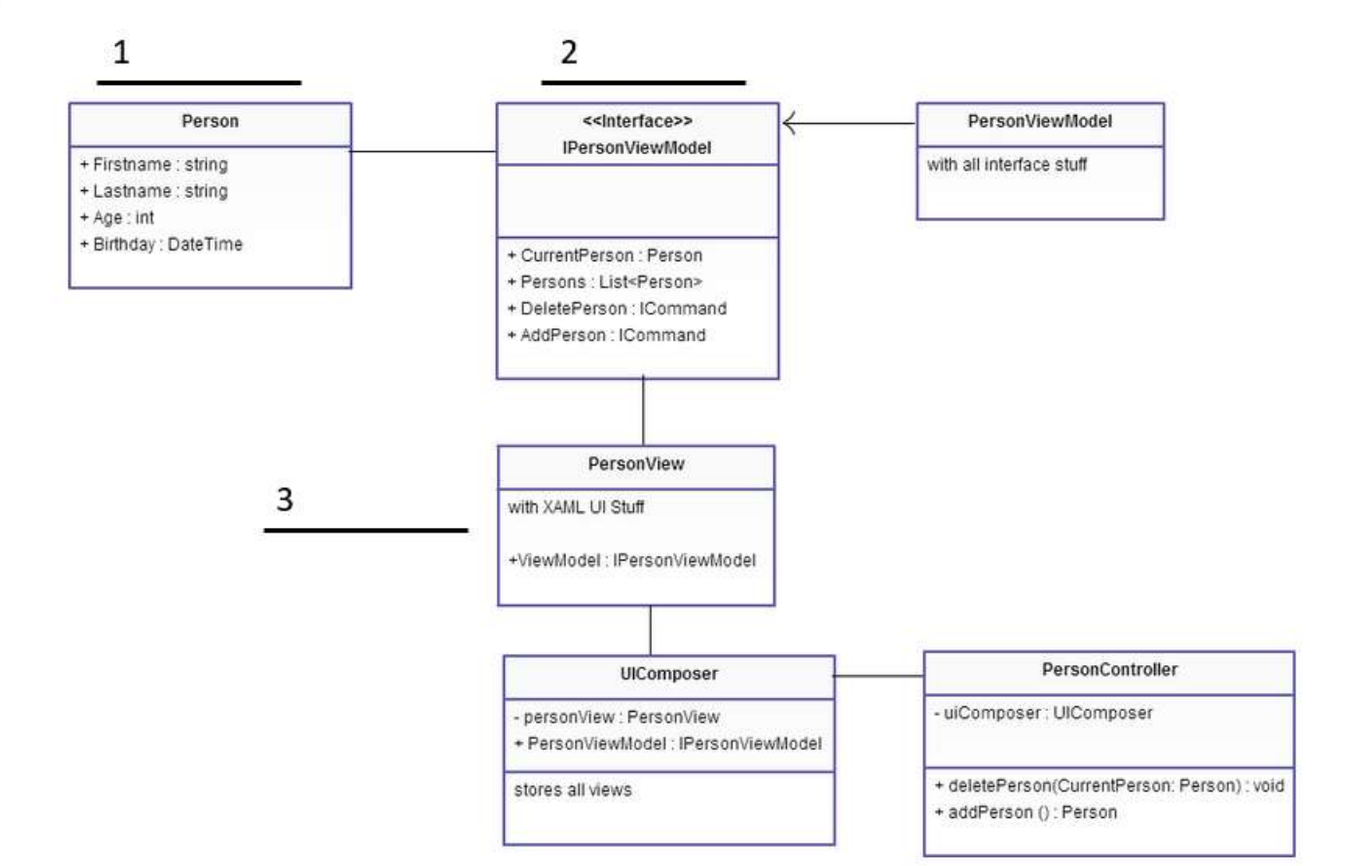
Réponses incorrectes	4 discordances	0 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☑	☐	Oui (+1)	Process processus = Process ("explorer.exe");
B	☑	☐	Oui (+1)	Process processus = new Process(); processus.StartInfo.FileName = "explorer.exe" ; processus.Start();
C	☐	☑	Oui (+1)	Process processus = Process.Start (); processus.StartInfo.FileName = "explorer.exe";
D	☐	☐	Non	Process processus = new Process().Start(); processus.StartInfo ("explorer.exe");
E	☐	☑	Oui (+1)	Process processus = new Process(); processus.StartInfo.FileName = "explorer.exe"; Process.Start();
F	☐	☐	Non	Process processus = Process.Start("explorer.exe");

Question 12 — **Question à réponse unique**

Parmi les combinaisons de mots-clés, quelle est la bonne combinaison pour compléter le diagramme ci-dessous ?

Which of the following keyword combinations is the correct combination to complete the diagram below?



Réponses incorrectes	0 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☐	☐	Non	1-Model, 2-Controler, 3-View
B	☐	☐	Non	1-Model, 2-Controler, 3-ViewModel
C	☐	☐	Non	1-ModelView, 2-Controler, 3-View
D	☐	☑	Oui (+1)	1-Controler, 2-ModelView, 3-View
E	☑	☐	Oui (+1)	1-Model, 2-ViewModel, 3-View

Question 13 — **Question à réponse unique**

Comment pouvez-vous caractériser l'utilisation du Design Pattern SINGLETON ?

How would you characterize the use of the SINGLETON Design Pattern?

Réponses correctes	1 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☑	☑	Non	C'est un patron de conception dont l'objectif est de restreindre l'instanciation d'une classe à un seul objet. *It is a design pattern whose objective is to restrict the instantiation of a class to a single object.*
B	☐	☐	Non	C'est un patron de conception qui permet d'instancier des objets qui seront utilisés une et une seule fois tout au long de l'exécution de l'application. *It is a design pattern that allows instantiating objects that will be used once and only once throughout the execution of the application.*
C	☐	☐	Non	C'est un patron de conception de type comportemental grâce auquel un et seul algorithme peut être sélectionnés à la volée au cours du temps d'exécution de l'application. *It is a behavioral design pattern through which one and only one algorithm can be selected on the fly during the execution time of the application.*
D	☐	☐	Non	C'est un patron de conception qui est utilisé pour envoyer un et un seul signal à des modules jouant le rôle d'observateurs et qui, en cas de notification, effectuent une action donnée. *It is a design pattern that is used to send one and only one signal to modules playing the role of observers and that, upon notification, perform a given action.*
E	☐	☐	Non	C'est un patron de conception qui permet de découpler une et une seule interface d'une classe et de son implémentation. *It is a design pattern that allows decoupling one and only one interface from a class and its implementation.*

Question 14 — **Question à réponse ouverte et courte**

Compléter la ligne de code pour passer les paramètres au processus. On suppose que ce programme lance un processus « copier un fichier d'un répertoire à un autre » **(case sensitive !)**.

Complete the code to pass the parameters to the process. We suppose that this program launches a process "copy a file from a directory to another" ***(case sensitive!)****.*

```
Process processus = null;
String parametres = "-s toto.txt -d titi.txt" ;

// Créer un processus
…
processus.StartInfo.FileName = "copy.exe";

processus.StartInfo.                    = parametres;

// Lancer le processus
```

Réponse incorrecte	0 point obtenu sur 1

FileName

Réponse attendue

Arguments

Question 15

Question à réponses multiples

Parmi ces propositions, lesquelles sont **fausses**? (2 réponses)

*Which of these statements are **false**? (2 answers)*

Réponses incorrectes	4 discordances	0 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☑	☐	Oui (+1)	Lorsque le « garbage collector » a effectué sa première opération de la génération 0, il promeut les objets accessibles à la génération 2 directement. *When the garbage collector has performed its first generation 0 operation, it promotes the accessible objects directly to generation 2.*
B	☐	☑	Oui (+1)	Le « garbage collector » procède à une opération collection lorsque la génération 0 est complète *The garbage collector performs a collection operation when generation 0 is complete*
C	☑	☐	Oui (+1)	Le « garbage collector » du runtime stocke de nouveaux objets dans la génération 2 *The runtime garbage collector stores new objects in generation 2*
D	☐	☐	Non	Le « garbage collector » du runtime stocke de nouveaux objets dans la génération 0 *The runtime garbage collector stores new objects in generation 0*
E	☐	☑	Oui (+1)	Les objets créés à un stade précoce de la vie de l'application et qui survivent aux opérations garbage collection sont promus et stockés dans les générations 1 et 2 *Objects created at an early stage of the application's life that survive garbage collection operations are promoted and stored in generations 1 and 2*

Question 16

Question à réponse unique

Les patrons de conception ont été formulés dans le début des années 90 pour poser les bases et les principes de la réutilisation de logiciels dans le cadre de la programmation orientée objet.
On parle souvent de « Design Pattern -Elements of Reusable Object-Oriented Software ».
Bien qu'elle soit toujours utilisée comme une référence, quel est le patron de conception qui n'a pas été abordé dans cette formulation ?

Design patterns were formulated in the early 90's to lay the foundations and principles of software reuse in the context of object-oriented programming.
They are often referred to as "Design Pattern -Elements of Reusable Object-Oriented Software".
Though it is still used as a reference, what design pattern has not been addressed in this formulation?

Réponses incorrectes	0 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☑	☐	Oui (+1)	MVC
B	☐	☐	Non	OBSERVER
C	☐	☑	Oui (+1)	FACTORY
D	☐	☐	Non	BRIDGE
E	☐	☐	Non	SINGLETON

Question 17 — Question à réponse unique

Dans le code suivant,

In the code below:

```
namespace Room
{
    internal sealed class Room
    {
        private static Room _instance = null;
        private Room()
        {

        }

        public static Room Instantiate()
        {

            if (_instance == null)
            {

                _instance = new Room();

            }
            return _instance;
        }
    }
}
```

Sur quel modèle de conception la classe Room est-elle implémentée?

On which design pattern is the Room class implemented?

Réponses correctes	1 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☐	☐	Non	PROTOTYPE
B	☑	☑	Non	SINGLETON
C	☐	☐	Non	DECORATOR
D	☐	☐	Non	BUILDER
E	☐	☐	Non	ADAPTATER

Question 18 — Question à réponse unique

Le formalisme UML permet de décrire la multiplicité d'une relation. Parmi les propositions de description de multiplicité ci-dessous, laquelle **est fausse** ? (1 réponse)

*UML formalism is used to describe the multiplicity of a relationship. Among the following proposals for describing multiplicity, which one **is false**? (1 answer)*

Réponses correctes	1 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☑	☑	Non	∝ signifie que le nombre de relations est indéfini ∝ *means that the number of relations is undefined*
B	☐	☐	Non	* signifie que le nombre de relations est plusieurs * *means that the number of relations is several*
C	☐	☐	Non	**n** signifie que le nombre de relations est exactement n **n** *means that the number of relations is exactly n*
D	☐	☐	Non	**n..m** signifie que le nombre de relations est compris entre n et m **n..m** *means that the number of relations is between n and m*
E	☐	☐	Non	**1** signifie qu'il y a qu'une seule relation. **1** *means that there is only one relation.*

Question 19 — **Question à réponse unique**

Concernant un diagramme de classe
Identifiez la MAUVAISE proposition.

Regarding a class diagram
Identify the WRONG proposal.

Réponses correctes	1 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☑	☑	Non	Ce type de diagramme fait apparaître les valeurs possibles des attributs *This type of diagram shows the possible values of the attributes*
B	☐	☐	Non	Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que leurs relations *The class diagram is a schema used in software engineering to present the classes and interfaces of systems and their relationships*
C	☐	☐	Non	Ce type de diagramme fait partie de la partie statique d' UML, ne s'intéressant pas aux aspects temporels et dynamiques *This type of diagram is part of the static part of the UML, not dealing with temporal and dynamic aspects*
D	☐	☐	Non	Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets *A class describes the responsibilities, behavior and type of a set of objects*
E	☐	☐	Non	Une classe peut comporter des instances d’autres classes *A class can have instances of other classes*

Question 20 — **Question à réponse unique**

Concernant les inconvénients du modèle MVC.
Identifiez la proposition qui n'est pas un inconvénient

Regarding the drawbacks of the MVC model.
Identify the proposition that is not a drawback

Réponses correctes	1 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☐	☐	Non	Augmentation de la complexité lors de l’implantation *Increased complexity during implementation*
B	☐	☐	Non	Éventuel cloisonnement des développeurs *Possible compartmentalization of developers*
C	☐	☐	Non	Architecture complexe pour des petits projets *Complex architecture for smaller projects*
D	☐	☐	Non	Le nombre important de fichiers représente une charge non négligeable dans un projet *The large number of files represents a significant burden in a project*
E	☑	☑	Non	Possibilité de réutilisation de code dans d’autres applications *Ability to reuse code in other applications*

Question 21 — **Question à réponses multiples**

Quels sont les 2 modes de fonctionnement d'un serveur dans une architecture Client-Serveur?
What are the 2 modes of operation of a server in a Client-Server architecture?

Réponses partiellement correctes	2 discordances	0.2 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☑	☐	Oui (+1)	Mode itératif *Iterative mode*
B	☑	☑	Non	Mode Concurrent *Concurrent mode*
C	☐	☑	Oui (+1)	Mode Partagé *Shared mode*
D	☐	☐	Non	Mode singleton *Singleton mode*
E	☐	☐	Non	Mode MVC *MVC Mode*

Question 22 — **Question à réponse unique**

Design Pattern - Quel Design Pattern ne fait pas parti des DP comportementaux ?
Which of the following is not a behavioral design pattern?

Réponses incorrectes	0 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☐	☐	Non	État *State*
B	☐	☐	Non	Stratégie *Strategy*
C	☐	☑	Oui (+1)	Patron de méthode *Template method*
D	☐	☐	Non	Visiteur *Visitor*
E	☑	☐	Oui (+1)	Façade *Facade*

Question 23

Question à réponses multiples

Type anonyme : Quelle syntaxe est juste (1 réponse)?

Anonymous type: Which syntax is correct (1 answer)?

Réponses partiellement correctes	2 discordances	0.3 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☑	☐	Oui (+1)	var v = new { Title = "Hello", Age = 24 };
B	☐	☐	Non	var v = new { Title = "Hello" ; Age = 24 };
C	☐	☑	Oui (+1)	var v = new (Title = "Hello", Age = 24);
D	☐	☐	Non	var v = new (Title = "Hello" ; Age = 24);
E	☐	☐	Non	var v = new [Title = "Hello" ; Age = 24];
F	☐	☐	Non	var v = new [Title = "Hello" , Age = 24];

Question 24

Question à réponse ouverte et courte

Quelle instruction a été masquée (case blanche) ?

Which instruction has been masked (white box)?

```
public sealed class Singleton
{
    private Singleton() { }

    private        Singleton _instance;

    public static Singleton GetInstance()
    {
        if (_instance == null)
        {
            _instance = new Singleton();
        }
        return _instance;
    }

    public void MyFunction()
    {
        //My code
    }
}
```

Réponse correcte	1 point obtenu sur 1

static

Réponse attendue

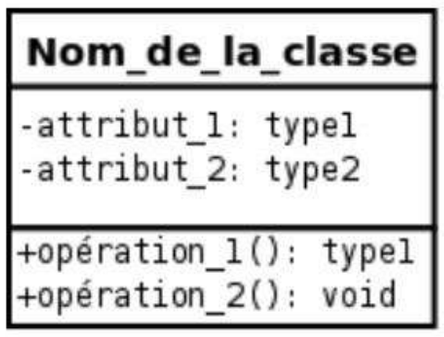
static

Question 25

Question à réponse unique

Dans le diagramme suivant, quelle est la signification de l'opérateur **"+"**

In the following diagram, what is the meaning of the "+" operator



Réponses incorrectes	0 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☐	☑	Oui (+1)	privé *private*
B	☑	☐	Oui (+1)	public *public*
C	☐	☐	Non	protégé *protected*
D	☐	☐	Non	fonction *function*
E	☐	☐	Non	fonction virtuelle *virtual function*

Question 26

Question à réponses multiples

Dans la phase de conception d'un logiciel, quels sont les apports d'un diagramme de séquences ? (1 réponse)

In the design phase of a software application, what are the contributions of a sequence diagram? (1 answer)

Réponses partiellement correctes	2 discordances	0.2 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☑	☐	Oui (+1)	Il représente l'ordonnancement des messages que s'échangent les différents objets tout au long de leurs interactions au cours du temps. *It represents the ordering of the messages that the different objects exchange throughout their interactions over time.*
B	☐	☐	Non	Il présente les différentes séquences de tests à réaliser pour décrire la stratégie de tests du logiciel. *It presents the different test sequences to be performed to describe the software testing strategy.*
C	☐	☐	Non	Il représente les interactions des utilisateurs pour dérouler un cas d'utilisation du logiciel. *It represents the interactions of users to unfold a use case of the software.*
D	☐	☑	Oui (+1)	Il représente les déclenchements d'événements en fonction des états du système pour aider à la modélisation des comportements parallélisables. *It represents event triggers according to system states to help model parallelizable behavior.*
E	☐	☐	Non	Il représente l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux. *It represents the use of the physical infrastructure by the system and how the system components are distributed and how they relate to each other.*

Question 27 **Question à réponse unique**

En programmation informatique, quelle affirmation ci-dessous décrit-elle le modèle de conception proposé par le design pattern STRATEGY ?

In computer programming, which statement below describes the design pattern proposed by the STRATEGY design pattern?

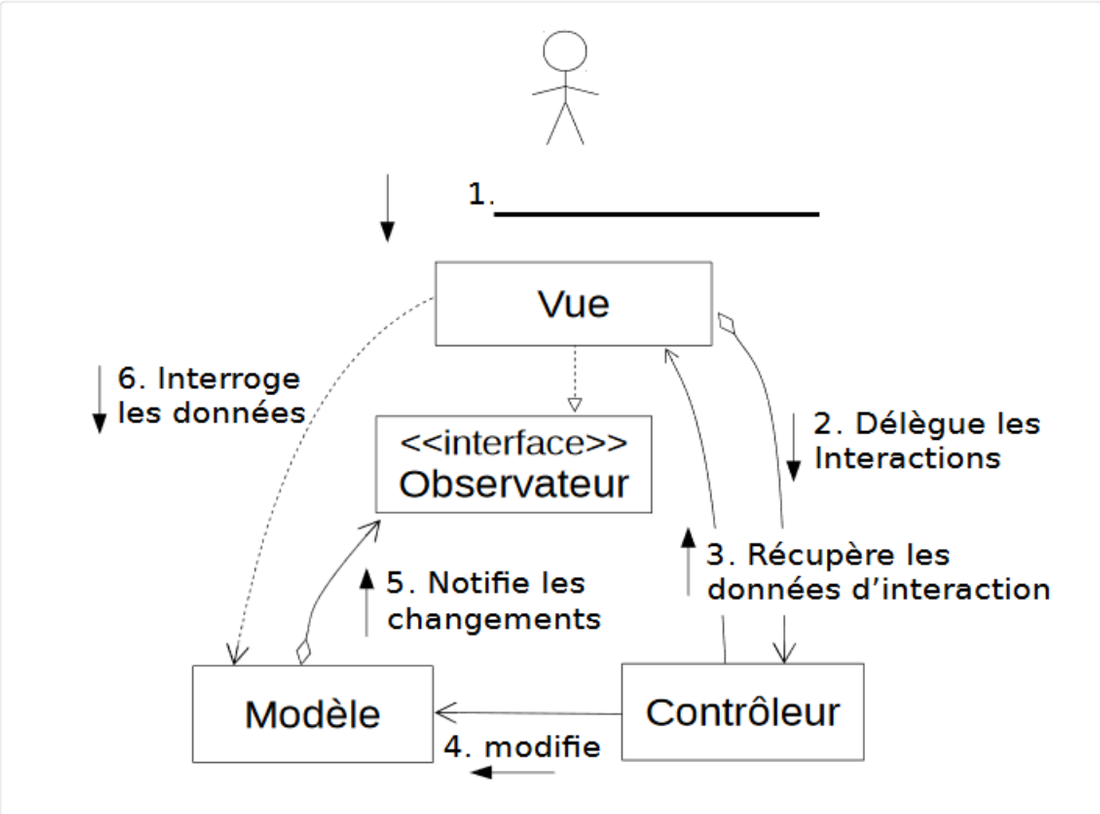
Réponses incorrectes	0 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☐	☑	Oui (+1)	Modèle de conception logiciel qui permet de créer une variété d'objets complexes à partir d'un objet source. L'objet source peut consister en une variété de parties contribuant individuellement à la création de chaque objet complet grâce à un ensemble d'appels à une interface commune de la classe abstraite décrivant l'objet source. *A software design pattern that allows a variety of complex objects to be created from a source object. The source object can consist of a variety of parts that individually contribute to the creation of each complete object through a set of calls to a common interface of the abstract class describing the source object.*
B	☑	☐	Oui (+1)	Modèle de conception de logiciel comportemental qui permet de sélectionner un algorithme lors de l'exécution. Au lieu d'implémenter directement un algorithme unique, le code reçoit des instructions d'exécution indiquant les algorithmes à utiliser dans une famille d'algorithmes. *A behavioral software design pattern that allows an algorithm to be selected at runtime. Instead of directly implementing a single algorithm, the code is given runtime instructions indicating which algorithms to use in a family of algorithms.*
C	☐	☐	Non	Modèle de conception qui permet de créer une et une seule instance d'un objet donnée utilisé dans une application logicielle. *A design pattern that allows you to create one and only one instance of a given object used in a software application.*
D	☐	☐	Non	Modèle de conception de l'architecture d'une application qui permet de répartir et regrouper les objets en trois grandes familles : les données à afficher, la présentation des interfaces graphiques et les traitements concernant les actions effectuées par les utilisateurs. *A design pattern for the architecture of an application that allows objects to be divided and grouped into three main families: the data to be displayed, the presentation of graphical interfaces, and the processing of actions performed by users.*
E	☐	☐	Non	Modèle de conception structurel qui permet de diviser un ensemble de classes étroitement liées en deux hiérarchies distinctes (abstraction et implémentation) pouvant être développées indépendamment l'une de l'autre. *A structural design pattern that allows a set of closely related classes to be divided into two distinct hierarchies (abstraction and implementation) that can be developed independently of each other.*

Question 28 — **Question à réponse unique**

Observez le diagramme de communication du modèle d'architecture d'application MVC ci-dessous.

Look at the communication diagram of the MVC application architecture model below.



Pouvez-vous compléter le schéma ci-dessus ? (1 réponse)

Réponses incorrectes	0 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☐	☐	Non	1) Affiche les résultats des traitements *1) Displays the results of the treatments*
B	☐	☐	Non	1) Observe les changements apportés aux données *1) Observes the changes to the data*
C	☑	☐	Oui (+1)	1) Interagit avec les composants graphiques *1) Interacts with graphic components*
D	☐	☐	Non	1) Monitore les modifications apportées sur les données *1) Monitors the changes made on the data*
E	☐	☑	Oui (+1)	1) Définit la volumétrie des données *1) Defines the volume of data*

Question 29 — **Question à réponses multiples**

Comment arrêter un processus « p » (2 bonnes réponses)

How to stop a "p" process (2 correct answers)

Réponses partiellement correctes	2 discordances	0.2 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☑	☐	Oui (+1)	p.Close();
B	☑	☑	Non	p.Kill();
C	☐	☐	Non	p.Finalise();
D	☐	☑	Oui (+1)	p.Stop();
E	☐	☐	Non	p.End();

Question 30 — **Question à réponse unique**

Les 23 Design Pattern décrits dans la littérature informatique et notamment dans l'ouvrage de Erich Gamma, sont classés en trois catégories principales. Quelles sont-elles ? (1 réponse)

The 23 Design Patterns described in the computer science literature, and in particular in Erich Gamma's book, are classified into three main categories. What are they? (1 answer)

Réponses incorrectes	0 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☑	☐	Oui (+1)	Création, Structurel, Comportement *Creation, Structure, Behavior*
B	☐	☑	Oui (+1)	Création, Algorithme, Comportement *Creation, Algorithm, Behavior*
C	☐	☐	Non	Initialisation, Algorithme, Graphique *Initialization, Algorithm, Graph*
D	☐	☐	Non	Graphique, Structurel, Comportement *Graphical, Structure, Behavior*
E	☐	☐	Non	Algorithme, Base de données, Création *Algorithm, Database, Creation*

Question 31 — **Question à réponse ouverte et courte**

Quelle est l'instruction manquante (carré blanc)?

What is the missing instruction (white square)?

```
public class EHClass
{
    void ReadFile(int index)
    {
        // To run this code, substitute a valid path from your local machine
        string path = @"c:\users\public\test.txt";
        System.IO.StreamReader file = new System.IO.StreamReader(path);
        char[] buffer = new char[10];
        try
        {
            file.ReadBlock(buffer, index, buffer.Length);
        }
        catch (System.IO.IOException e)
        {
            Console.WriteLine("Error reading from {0}. Message = {1}", path, e.Message);
        }
        [redacted]
        {
            if (file != null)
            {
                file.Close();
            }
        }

    }
}
```

Réponse incorrecte	0 point obtenu sur 1

else

Réponse attendue

finally

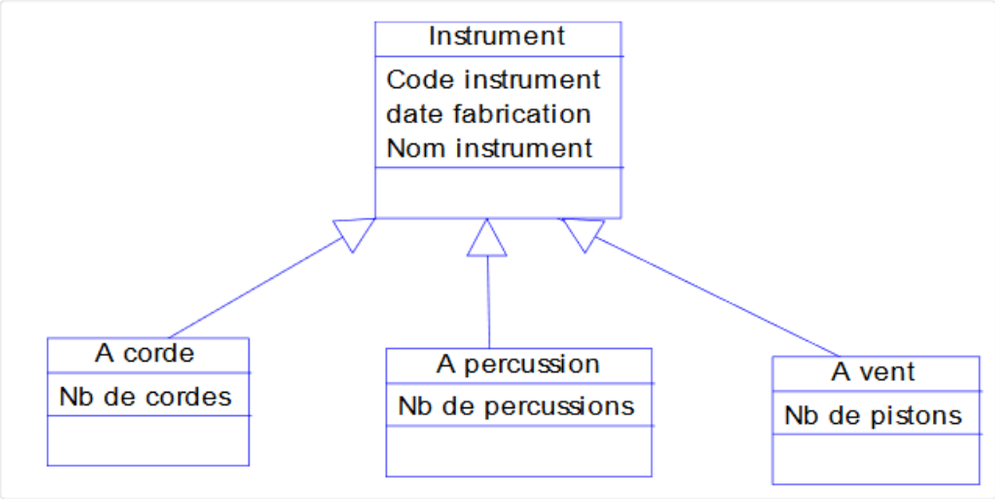
Question 32 — Question à réponse unique

Observez le diagramme suivant :

Quel type d'association ce diagramme traduit-il ? (1 réponse)

Look at the following diagram:

What type of association does this diagram reflect? (1 answer)



Réponses incorrectes	0 point obtenu sur 1

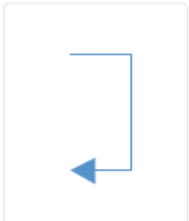
	Réponse attendue	Réponse saisie	Réponse discordante	
A	☑	☐	Oui (+1)	Héritage *Inheritance*
B	☐	☐	Non	Agrégation *Aggregation*
C	☐	☐	Non	Composition *Composition*
D	☐	☐	Non	Héritage multiple *Multiple inheritance*
E	☐	☑	Oui (+1)	Dépendance *Dependence*

Question 33 — Association

Dans un diagramme de séquence, quelle est la signification des symboles suivants ?

In a sequence diagram, what is the meaning of the following symbols?

Réponses incorrectes	2 discordances	0.2 point obtenu sur 1

Élément à associer	Réponse attendue	Réponse saisie	Réponse discordante
	Acteur *Actor*	Acteur *Actor*	Non
	Message de retour *Return message*	Message de retour *Return message*	Non
	Message automatique *Automatic message*	Message automatique *Automatic message*	Non
	Message asynchrone *Asynchronous message*	Message synchrone *Synchronous message*	Oui (+1)
	Message synchrone *Synchronous message*	Message asynchrone *Asynchronous message*	Oui (+1)

Question 34 — **Question à réponse unique**

Design Pattern – Les types de Design Patterns / De qui parlons nous ?

Ce patron est un patron de conception de la famille des patrons comportementaux. Il est utilisé pour envoyer un signal à des modules. En cas de notification, les modules effectuent alors l'action adéquate en fonction des informations qui parviennent depuis les autres modules Ce patron permet de limiter le couplage entre les modules aux seuls signaux émis.

Which design pattern are we talking about?

This pattern is a design pattern of the behavioral pattern family. It is used to send a signal to modules. In case of notification, the modules perform the appropriate action according to the information received from the other modules. This pattern allows to limit the coupling between the modules to the signals sent.

Réponses incorrectes	0 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☑	☐	Oui (+1)	Observateur *Observer*
B	☐	☐	Non	MVC *MVC*
C	☐	☑	Oui (+1)	Stratégie *Strategy*
D	☐	☐	Non	Adaptateur *Adapter*
E	☐	☐	Non	Décorateur *Decorator*

Question 35 — **Question à réponses multiples**

Quelle(s) technologie(s) n'est (ne sont) pas basée(s) sur une architecture client-serveur ?

Which technology(ies) is (are) not based on a client-server architecture?

Réponses incorrectes	3 discordances	0 point obtenu sur 1

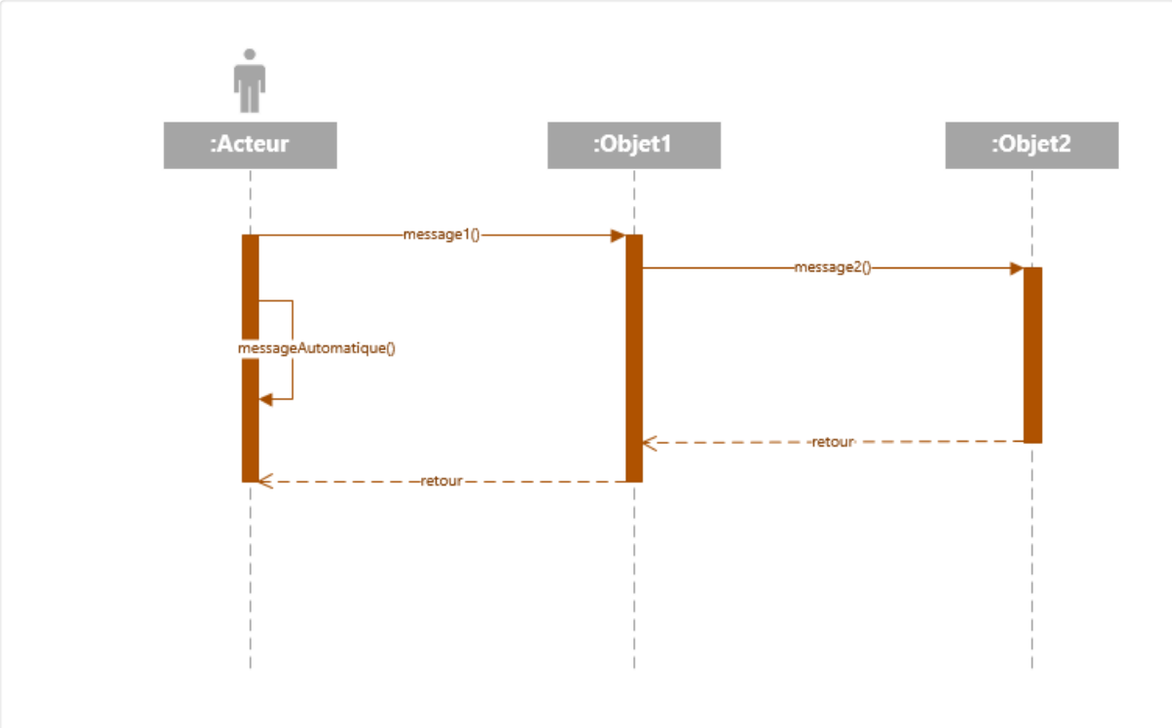
	Réponse attendue	Réponse saisie	Réponse discordante	
A	☐	☑	Oui (+1)	DHCP
B	☐	☑	Oui (+1)	socket
C	☐	☐	Non	FTP
D	☑	☐	Oui (+1)	Peer to Peer
E	☐	☐	Non	DNS

Question 36

Question à réponse unique

Le diagramme ci-dessous est un

The diagram below is a:



Réponses correctes	1 point obtenu sur 1

	Réponse attendue	Réponse saisie	Réponse discordante	
A	☑	☑	Non	Diagramme de séquence *Sequence diagram*
B	☐	☐	Non	Diagramme de composant *Component diagram*
C	☐	☐	Non	Diagramme de classe *Class diagram*
D	☐	☐	Non	Diagramme d'activité *Activity diagram*
E	☐	☐	Non	Diagramme de collaboration *Collaboration diagram*
F	☐	☐	Non	Diagramme de déploiement *Deployment diagram*