**(/100)**

| **Critères à évaluer** | **Insuffisant** | **Faible** | **Bon** | **Très bien** | **Excellent** |
| --- | --- | --- | --- | --- | --- |
| **(20)** **Production d’un diagramme de classe conforme à l’énoncé (00Q6.1, 00Q6.2)** | | | | | |
| **(20) Conformité du diagramme à l’énoncé** | * Absences ou inadéquation de la majorité des classes et des relations décrites dans l’énoncé. | * Absence ou inadéquation de la majorité des classes **ou** des relations. * Identification correcte à partir d’un énoncé de la majorité des classes **ou** relations toutefois inadéquations de la majorité des attributs, des méthodes, des noms de relations ou des cardinalités des relations. | * Identification correcte à partir d’un énoncé de la majorité des classes et de la majorité des relations entre celles-ci.   Pour les classes identifiées :   * La majorité les attributs sont adéquats. Présence de quelques erreurs mineures (modificateur, type) * La majorité des méthodes sont adéquates. Présence de quelques erreurs mineures (paramètres, type des paramètres, valeurs de retour, type des valeurs de retour) * La majorité les relations sont adéquates. Présence de quelques erreurs mineures (cardinalités des associations). | * Identification correcte à partir d’un énoncé de toutes les classes et de toutes les relations entre celles-ci. * Tous les attributs sont adéquats. Présence de quelques erreurs mineures (modificateur, type) * Toutes les méthodes sont adéquates. Présence de quelques erreurs mineures (paramètres, type des paramètres, valeurs de retour, type des valeurs de retour) * Toutes les relations sont adéquates. Présence de quelques erreurs mineures (cardinalités des associations). | * Identification correcte à partir d’un énoncé de toutes les classes et de toutes les relations entre celles-ci. * Tous les attributs sont adéquats et corrects (type, modificateur). * Toutes les méthodes sont adéquates et correctes (paramètres, type des paramètres, valeurs de retour, type des valeurs de retour) * Toutes les relations sont adéquates et correctes (nom et cardinalités des associations). |
|  | **0** | **6** | **12** | **18** | **20** |
| **Auto-évaluation ->>** |  |  |  |  |  |

| **Critères à évaluer** | **Insuffisant** | **Faible** | **Bon** | **Très bien** | **Excellent** |
| --- | --- | --- | --- | --- | --- |
| **(15) Choix et positionnement appropriés des éléments graphiques pour l’affichage et la saisie (00Q6.4)** | | | | | |
| (**15) Choix et positionnement appropriés des contrôles graphiques** | - Absence ou mauvais choix de la majorité des contrôles. | * Absence ou mauvais choix de plusieurs contrôles graphiques. * Plusieurs contrôles graphiques sont de taille inappropriée ou sont mal positionnés. * Plusieurs gestionnaires d’événement sont disfonctionnels | * Absence ou mauvais choix de quelques contrôles graphiques * Quelques contrôles graphiques sont de taille inappropriée ou sont mal positionnés. * La majorité des gestionnaires d’événement sont fonctionnels pour activer des contrôles ou lancer des fenêtres. | * Choix judicieux de la majorité des contrôles graphiques * La majorité des contrôles graphiques sont de taille appropriée et bien positionnés. * Les couleurs sont utilisées de manière   appropriée la plupart du temps. | * Choix judicieux de tous les contrôles graphiques. * Tous les contrôles graphiques sont de taille appropriée et bien positionnés. * Utilisation appropriée des couleurs. * Tous les gestionnaires d’événement sont correctement codés et permettent d’activer les contrôles ou de lancer des fenêtres. |
|  | **0** | **6** | **10** | **12** | **15** |
| **Auto-évaluation ->>** |  |  |  |  |  |

| **Critères à évaluer** | **Insuffisant** | **Faible** | **Bon** | **Très bien** | **Excellent** |
| --- | --- | --- | --- | --- | --- |
| **(35) Programmation adéquate de la logique applicative selon les principes de l’orientée objet, validation des données entrées et gestion appropriée des erreurs (00Q6.3, 00Q6.5)** | | | | | |
| **(12) Programmation adéquate des classes** | Absence ou erreurs de codage de la majorité des membres de classe. | Absence de plusieurs membres de classe ou présence d’ erreurs dans le codage de plusieurs membres de classe. | * Quelques erreurs se retrouvent dans le codage des classes. | * Codage correcte de l’héritage et de tous les membres de classe pour toutes les classes du programme avec quelques erreurs mineures. | * Tous les membres de classe suivants sont correctement codés pour toutes les classes du programme : * Les constructeurs, les attributs d’instance incluant les attributs issus des associations, les attributs de classe et les propriétés (accesseurs et modificateurs). * L’héritage entre classes est correctement codé. |
|  | **0** | **8** | **12** | **18** | **20** |
| **Auto-évaluation ->>** |  |  |  |  |  |
| **(8) Programmation adéquate des méthodes** | La majorité des méthodes présentent des erreurs de codage. | Plusieurs erreurs de codage sont présentes par catégorie de méthode | Quelques erreurs de codage sont présentes par catégorie de méthode. | * Présence d’erreurs de codage mineures dans les méthodes. * Absence de quelques méthodes str(). | Codage correcte de ces types de méthodes pour toutes les classes :   * Les méthodes magiques * Les méthodes d’instance * Les méthodes de classes * Les méthodes statiques |
|  | **0** | **3** | **5** | **7** | **8** |
| **Auto-évaluation ->>** |  |  |  |  |  |
| **(5) Validation des données entrées** | - Aucune ou très peu de validations ne sont effectuées sur les entrées des utilisateurs. | * Mise en place d’une certaine forme de validation inconsistante et limitée. | * La validation est appliquée à toutes les entrées de l’utilisateur avec quelques erreurs de détection d’entrées invalides. * Les messages d’erreurs gagneraient à être conviviaux. | * La validation des entrées selon les principes de l’orienté objet est appliquée à toutes les entrées de l’utilisateur. Certains disfonctionnements mineurs peuvent se produire. | * La validation des entrées selon les principes de l’orienté objet est exhaustive avec affichage de messages d’erreurs informatifs et conviviaux. |
|  | **0** | **2** | **3** | **6** | **5** |
| **Auto-évaluation ->>** |  |  |  |  |  |
| **(3) Sérialisation/Désérialisation correcte des objets complexes** | - Les erreurs commises empêchent la séarialisation / déséréalisation et entrainent une perte d’information importante. | * Présence de plusieurs erreurs de sérialisation/désérialisation d’objets. Ces erreurs engendrent quelques perte d’information. | * Présence de quelques erreurs de sérialisation/désérialisation des objets. | * La séréalistion et la déséréalisation des objets complexes est correcte. Présence de quelques erreurs mineures. | * La sérialisation et la désérialisation sont correctes et permettent la gestion des objets complexes (avec associations). |
|  | **0** | **1** | **2** | **2.5** | **3** |
| **Auto-évaluation ->>** |  |  |  |  |  |
| **(4) Programmation adéquate de la logique applicative** | - La majorité des exigences de l’énoncé de se retrouvent pas dans la logique applicative. | * Plusieurs ajustements sont nécessaires pour une correspondance complète avec les exigences de l’énoncé. | * Quelques ajustements sont nécessaires pour une correspondance complète avec les exigences de l’énoncé. | * La logique applicative répond généralement aux exigences de l’énoncé. Quelques ajustements mineurs sont nécessaires pour une correspondance complète. | * La logique applicative répond entièrement à l’énoncé. * Le code est hautement modulaire. |
|  | **0** | **2** | **3** | **3.5** | **4** |
| **Auto-évaluation ->>** |  |  |  |  |  |
| **(3) Gestion correcte et exhaustive des exceptions** | Absence de gestion d’exception | La gestion des exceptions est correcte. Plusieurs exceptions sont non gérées. | * La gestion des exceptions est correcte. Quelques exceptions sont non gérées. | * La gestion des exceptions est correcte.La plupart des exceptions attendues sont gérées. | * La gestion des exceptions attendues est correcte et exhaustive. |
|  | **0** | **1** | **2** | **2.5** | **3** |

| **(20) Production de tests adéquats (00Q6.5)** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Critères à évaluer** | **Insuffisant** | **Faible** | **Bon** | **Très bien** | **Excellent** |
| **(20) Production de tests adéquats (00Q6.5)** | * Absence de tests unitaires ou couverture faible du code.   Les assertions sont absentes ou peu nombreuses. | * Les jeux d’essai permettent une couverture limitée du code. * Plusieurs tests ne s’exécutent pas correctement. | * Les jeux d’essai permettent une couverture modérée du code. Certains cas exceptionnels et/ou des chemins complexes sont laissés de côté. * Les assertions utilisées permettent de vérifier le comportement du code dans différentes situations * La plupart des tests s’exécutent correctement. | * La plupart des chemins conditionnels complexes sont couverts par les tests unitaires. * Les assertions utilisées permettent de couvrir la plupart des aspects du comportement du code. | * Les jeux d’essai permettent une couverture complète du codage incluant, les cas limites et les chemins conditionnels complexes. * Les assertions utilisées permettent de couvrir tous les aspects du comportement du code. * Tous les tests s’exécutent sans erreurs. |
|  | **0** | **8** | **12** | **18** | **20** |
| **Auto-évaluation ->>** |  |  |  |  |  |

| **(10) Production d’une documentation de qualité respectueuse des normes du cours et respect des règles d’orthographe et de grammaire (00Q6.6)** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Critères à évaluer** | **Insuffisant** | **Faible** | **Bon** | **Très bien** | **Excellent** |
| **(7 Respect des normes de programmation** | La majorité des normes de programmation ne sont pas respectées pour :   * Les noms des variables, fonctions ou classes * Les docstrings   ET pour la clarté du code | Plusieurs normes de programmation ne sont pas respectées pour :   * Les noms des variables, fonctions ou classes * Les docstrings   Ou pour la clarté du code | Respect des normes de programmation dans la plupart des cas pour :   * Les noms significatifs, * Les docstrings * Les commentaires   Et La clarté du code   * Présence de quelques erreurs de grammaire et d’orthographe. | Respect des normes de programmation dans la plupart des cas pour :   * Les noms significatifs, * Les docstrings * Les commentaires   Et La clarté du code  Les erreurs mineures sont seulement au niveau de la clarté des descriptions des docstring ou de grammaire. | **Tous (Toutes)** :  - Les noms de variables, fonctions et classes sont descriptifs et reflètent leur fonction ou leur contenu.  - Les fonctions, méthodes et classes sont correctement documentées à l'aide de docstrings.  - Le code est accompagné de commentaires clairs et pertinents qui expliquent le fonctionnement complexe ou non évident.  - Le code est organisé de manière logique, évitant les constructions complexes et favorisant la lisibilité.  - Respect des règes de grammaire et d’orthographe. |
|  | **0** | **2** | **4** | **6** | **7** |
| **Auto-évaluation ->>** |  |  |  |  |  |
| **(3) Respect rigoureux des normes UML** | La majorité des conventions UML ne sont pas respectées pour au moins trois des catégories des membres de classes suivantes :   * Les classes * Les attributs * Les propriétés * Les méthodes * Les relations | La majorité des conventions UML sont respectées pour au plus deux des catégories des membres de classes suivantes :   * Les classes * Les attributs * Les propriétés * Les méthodes * Les relations | Respect des conventions UML dans la plupart des cas pour les :   * classes * attributs * méthodes * méthodes * et les relations   avec quelques omissions (les paramètres et valeurs de retour des méthodes, type d’attributs, les noms de l’association) | Respect des conventions UML dans la plupart des cas pour les :   * classes * attributs * méthodes * méthodes * et les relations   avec des omissions mineures (les paramètres et valeurs de retour des méthodes) | Conformité aux conventions UML de toutes les classes, les attributs, les propriétés, les méthodes et les relations. |
|  | **0** | **1** | **2** | **2.75** | **3** |
| **Auto-évaluation ->>** |  |  |  |  |  |