

Examen professionnel Développement d'applications TIC

Informations sur l'examen

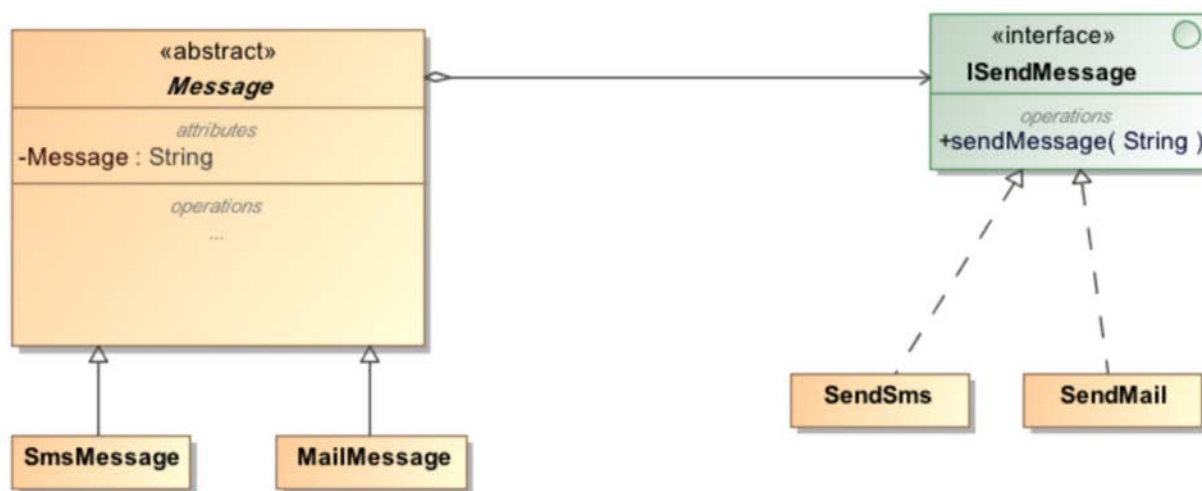
- Pour chaque problème, le nombre maximal de points que vous pouvez obtenir en cas de bonne réponse est mentionné. Le nombre de points est aussi indicatif du temps nécessaire à la résolution d'un problème (points-minutes).
- Veuillez écrire de façon lisible. Les réponses illisibles ne seront ni corrigées ni évaluées.
- Les corrections que vous apportez à vos réponses doivent être clairement reconnaissables en tant que telles (p. ex. en biffant). Les réponses ambiguës ne seront ni corrigées ni évaluées.
- Pour chaque problème, vous disposez d'un champ de réponse où indiquer votre solution. Les textes ou esquisses qui sortent du champ défini pour votre solution ne seront ni corrigés ni évalués. Si vous devez apporter d'importantes corrections, des feuilles prévues à cet effet se trouvent à la fin du livret d'examen. Le cas échéant, veuillez renvoyer clairement à cette feuille de correction dans le champ de réponse du problème.
- Les indications quant à la forme de la réponse et les limitations définies pour la solution d'un problème (p. ex. «en 3 phrases» ou «5 mots-clés au maximum») doivent être respectées. Les experts ne corrigeront ni n'évalueront les réponses ne respectant pas la forme prescrite ou les réponses supplémentaires, c'est-à-dire que le corrigé débutera par les premiers éléments mentionnés.
- Vos réponses doivent se référer concrètement au cas présenté. Si, par exemple, vous devez indiquer des mesures à prendre ou les prochaines étapes à effectuer, des réponses générales telles que «planifier» ou «communiquer» ne suffisent pas.

Implémentation d'un design pattern

Situation initiale

Vous travaillez dans l'unité de développement de logiciels d'une petite entreprise spécialisée dans l'implémentation de solutions de communication de différents types. Pour le module de communication central, vous avez reçu un extrait d'un diagramme de classe représentant les classes pour l'envoi d'un message texte.

Le diagramme de classe illustré pour l'envoi d'un message texte n'est pas complet et se base sur le pattern XY.



| Problème 1 | | | | 16 points |
|--|------------------------------|-----------|-----------------------|-----------------------------------|
| Identifiez sur la base du diagramme de classe présenté dans la situation initiale les (prototypes des) méthodes de la classe «Message». | | | | |
| Forme de la réponse Indiquez dans le champ de réponse du tableau, pour chaque méthode identifiée, le nom de la méthode, le type de retour (return type), le ou les paramètres ainsi que la visibilité de la méthode. | | | | |
| Champ de réponse | | | | <i>Corrigé (laisser vide)</i> |
| Nom de la méthode | Type de retour (return type) | Paramètre | Visibilité (modifier) | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| <i>Remarques concernant le corrigé (laisser vide)</i> | | | | |

| | |
|--|-----------------------------------|
| Problème 2 | 15 points |
| <p>Programmez la classe «SmsMessage» et ses méthodes à partir du diagramme de classe. Indiquez la syntaxe correcte en Java ou C#.</p> <p>Il n'est pas nécessaire d'indiquer les déclarations d'importation de packages, de classes ou constructeurs.</p> | |
| Forme de la réponse Classe programmée correctement, avec variables et méthodes. | |
| Champ de réponse | <i>Corrigé (laisser vide)</i> |
| <i>Remarques concernant le corrigé (laisser vide)</i> | |

| Solution problème 1 | 16 points | | | | | | | | | | | | | | | | | | | | |
|---|------------------------------|-------------------|------------------------------|-----------|-----------------------|------------|------|--------|--------|------------|--------|--|--------|----------------|------|--------------|-----------------|---------------|------|--|-----------------|
| <pre> classDiagram class Message { <<abstract>> -Message : String +setMessage(String) +getMessage() : String +setSendMessage(ISendMessage) +sendMessage() } class ISendMessage { <<interface>> +sendMessage(String) } class SmsMessage class MailMessage class SendSms class SendMail Message < -- SmsMessage Message < -- MailMessage Message < -- ISendMessage SendSms .. > ISendMessage SendMail .. > ISendMessage </pre> | | | | | | | | | | | | | | | | | | | | | |
| [→ adapter l'en-tête en fonction de l'exposé du problème] | | | | | | | | | | | | | | | | | | | | | |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #d3d3d3;"> <th style="width: 25%;">Nom de la méthode</th> <th style="width: 25%;">Type de retour (return type)</th> <th style="width: 25%;">Paramètre</th> <th style="width: 25%;">Visibilité (modifier)</th> </tr> </thead> <tbody> <tr> <td>setMessage</td> <td>void</td> <td>String</td> <td>public</td> </tr> <tr> <td>getMessage</td> <td>String</td> <td></td> <td>public</td> </tr> <tr> <td>setSendMessage</td> <td>void</td> <td>ISendMessage</td> <td>public abstract</td> </tr> <tr> <td>sendMessage()</td> <td>void</td> <td></td> <td>public abstract</td> </tr> </tbody> </table> | | Nom de la méthode | Type de retour (return type) | Paramètre | Visibilité (modifier) | setMessage | void | String | public | getMessage | String | | public | setSendMessage | void | ISendMessage | public abstract | sendMessage() | void | | public abstract |
| Nom de la méthode | Type de retour (return type) | Paramètre | Visibilité (modifier) | | | | | | | | | | | | | | | | | | |
| setMessage | void | String | public | | | | | | | | | | | | | | | | | | |
| getMessage | String | | public | | | | | | | | | | | | | | | | | | |
| setSendMessage | void | ISendMessage | public abstract | | | | | | | | | | | | | | | | | | |
| sendMessage() | void | | public abstract | | | | | | | | | | | | | | | | | | |
| <p>Instructions pour la notation / Clé de répartition des points</p> <p>Pour chaque méthode, 4 points au max.</p> <ul style="list-style-type: none"> - Nom de la méthode correct 1 - Type de retour 1 - Paramètre 1 - Modifier 1 | | | | | | | | | | | | | | | | | | | | | |

| Solution problème 2 | 15 points | | | | | | | | | | | | | | | | | | | | |
|---|-----------|--------------------------------|---|--------------------------------------|--|----------------------------|---|--------------------|---|----------------------------------|---|--------------------------------------|---|-----------------------------------|---|---------------------------------|---|--------------------------------|---|---|---|
| <p>En Java:</p> <pre>public class SmsMessage extends Message{ private ISendMessage sendMessage = new SendSms(); public void setSendMessage(ISendMessage sendMessage) { this.sendMessage = sendMessage; } public void sendMessage() { sendMessage.sendMessage(getMessage()); } }</pre> <p>En C#:</p> <pre>public class SmsMessage : Message { private ISendMessage sendMessage = new SendSms(); public void setSendMessage(ISendMessage sendMessage) { this.sendMessage = sendMessage; } public override void sendMessage() { sendMessage.sendMessage(getMessage()); } }</pre> | | | | | | | | | | | | | | | | | | | | | |
| <p>Instructions pour la notation / Clé de répartition des points</p> <table border="0"><tr><td>- Notation Java ou C# correcte</td><td>5</td></tr><tr><td>- Déclaration de la classe correcte:</td><td></td></tr><tr><td> o Nom de la classe correct</td><td>1</td></tr><tr><td> o Héritage correct</td><td>2</td></tr><tr><td>- Variable IsendMessage indiquée</td><td>1</td></tr><tr><td> o Initialisation par défaut indiquée</td><td>1</td></tr><tr><td>- Méthode setSendMessage indiquée</td><td>1</td></tr><tr><td> o Variable définie correctement</td><td>1</td></tr><tr><td>- Méthode sendMessage indiquée</td><td>1</td></tr><tr><td> o sendMessage.sendMessage(getMessage()) correct</td><td>2</td></tr></table> | | - Notation Java ou C# correcte | 5 | - Déclaration de la classe correcte: | | o Nom de la classe correct | 1 | o Héritage correct | 2 | - Variable IsendMessage indiquée | 1 | o Initialisation par défaut indiquée | 1 | - Méthode setSendMessage indiquée | 1 | o Variable définie correctement | 1 | - Méthode sendMessage indiquée | 1 | o sendMessage.sendMessage(getMessage()) correct | 2 |
| - Notation Java ou C# correcte | 5 | | | | | | | | | | | | | | | | | | | | |
| - Déclaration de la classe correcte: | | | | | | | | | | | | | | | | | | | | | |
| o Nom de la classe correct | 1 | | | | | | | | | | | | | | | | | | | | |
| o Héritage correct | 2 | | | | | | | | | | | | | | | | | | | | |
| - Variable IsendMessage indiquée | 1 | | | | | | | | | | | | | | | | | | | | |
| o Initialisation par défaut indiquée | 1 | | | | | | | | | | | | | | | | | | | | |
| - Méthode setSendMessage indiquée | 1 | | | | | | | | | | | | | | | | | | | | |
| o Variable définie correctement | 1 | | | | | | | | | | | | | | | | | | | | |
| - Méthode sendMessage indiquée | 1 | | | | | | | | | | | | | | | | | | | | |
| o sendMessage.sendMessage(getMessage()) correct | 2 | | | | | | | | | | | | | | | | | | | | |