

特定非営利活動法人 UML モデリング推進協議会

UML モデリング技能認定試験

L3 サンプル問題

知識問題

※本書の一部または全部を無断で複写、複製、転載、テープ化、ファイル化することを禁じます。

※UML、Unified Modeling Language は OMG (Object Management Group) の商標です。

A.結合度と凝集度

問題

S1 結合度と凝集度の問題:正しい文章をすべて選べ

ここで、モジュールとは、クラスや手続きやパッケージなどのソフトウェアの論理的な単位のこと、言語や設計手法の違いによって色々な単位が存在します。

1. 結合度や凝集度は、構造化分析・構造化設計における概念であり、オブジェクト指向には直接関係しない。
2. 凝集度とは、あるモジュール内のデータや手続きが関係しあっているかどうかを示す指標であり、凝集度は高い方がよい。
3. 結合度とは、あるモジュール内のデータや手続きが関係しあっている度合いを示す指標であり、結合度は高い方がよい。
4. 凝集度とは、あるシステム全体のデータや手続きが関係しあっているかどうかを示す指標であり、凝集度は高い方がよい。
5. 結合度とは、あるシステム全体のデータや手続きが関係しあっているかどうかを示す指標であり、結合度は高い方がよい。

B.オブジェクト指向設計原則

問題

S2 設計原則に関する問題:正しい文章をすべて選べ

1. オブジェクトとクラスとは対立する概念であり、オブジェクトを重視した設計が求められる。
2. オブジェクト指向設計においては、データ(属性)、手続き(操作)、インターフェース(シグニチャ)などをすべて対等に扱った設計が求められる。
3. オブジェクト指向設計においては、個々のクラスのソースコード自体の構成管理に配慮することが設計の基本である。
4. あるソフトウェアの振る舞いを拡張する必要がある場合に、どうしても元のソースコードにある程度修正を加えることは不可避なので、ソフトウェアの構成管理・版管理を徹底することが肝要である。
5. あるソフトウェアの振る舞いを拡張する必要がある場合に、妥当な設計を適用すればそのソフトウェアの元のソースコードを変更することなく拡張に対応することは可能である。
6. オブジェクト指向設計においては、個々のクラスの実装のしかたではなく、それらのクラスの利用のされ方に焦点を当てた設計が求められる。

問題

S3 設計原則に関する問題:スーパークラス A とそのサブクラス B が妥当な継承関係にあり、いずれもメソッド $m1(x)$ を定義している場合に満たさなければいけない性質に関して正しい文章をすべて選べ。

1. UML で $a: A$, $b: B$ と宣言する場合、オブジェクト a やオブジェクト b の型はそのオブジェクトが直属するクラスに限られる。
2. クラス A のインスタンス a に対して、UML で $a: B$ と宣言してもよい
3. クラス B のインスタンス b に対して、UML で $b: A$ と宣言してもよい
4. クラス A のインスタンスはクラス B のインスタンスとしても振舞える必要がある。
5. クラス B のインスタンスはクラス A のインスタンスとしても振舞える必要がある。
6. スーパークラス A のメソッド $m1(x):R$ の戻り値の型が R であり、そのサブクラス B のメソッド $m1(x):S$ の戻り値の型が S であるとき、S は R と同じかあるいはそのサブクラスでなければならない。
7. スーパークラス A のメソッド $m1(x):R$ の戻り値の型が R であり、そのサブクラス B のメソッド $m1(x):S$ の戻り値の型が S であるとき、S は R と同じかあるいはそのスーパークラスでなければならない。

C.分析原則

問題

S4 クラス識別のガイドライン:分析モデルに登場させるクラスを識別する基準として妥当なものをすべて選びなさい。

分析モデリングとは、設計の上流工程として開発すべきシステムの対象業務領域を理解し要求を記述するための概念クラスをモデル化する作業のこととする。

1. その業務を成り立たせている重要な管理・記録すべき対象を分析クラスの候補とする
2. オブジェクトはすべて現実世界の具体的な実体に対応するべきである
3. 「食事」「実施」「制御」などのように動作から派生したオブジェクトは実体とみなすには議論の余地があり、クラスとして定義するのは問題である。
4. 業務の現場に具体的に登場する要素だけが分析時にはクラスとして定義される。
5. 分析モデルにおいては、業務上の慣習や設計上の制約などを排除して、最大限一般的で汎用的なクラス定義を目指しておく必要がある。
6. 設計モデルだけでなく分析モデルにおいても業務をモデル化する際の目的・観点を明確にしないと識別したクラス候補の妥当性は議論できない。

D.モデリング技法

問題

以下、テーマと形式だけ示します（内容は省略）

ビジネスプロセスとユースケースの関係：ビジネスプロセスを定義したアクティビティ図とユースケース図やユースケース定義との関係として妥当なものをすべて選びなさい。

ユースケース定義とクラスの関係：ユースケース定義とクラス識別との関係として妥当なものをすべて選びなさい。

ロバストネス分析：ロバストネス分析の説明として妥当なものをすべて選びなさい。

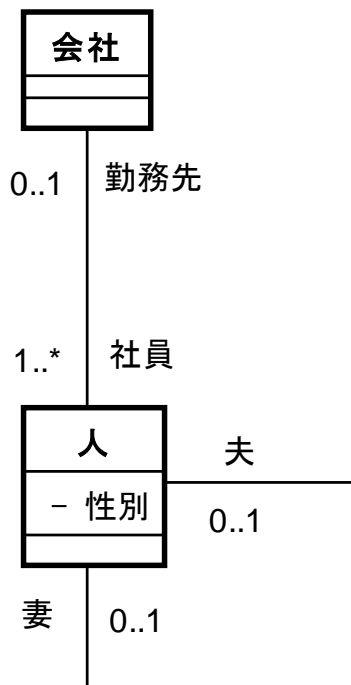
CRC 技法：クラスを識別する際に CRC という方針で識別する方法が提案されている。CRC の説明として妥当なものをすべて選びなさい。

E. OCL

問題

S5 OCL の基本理解：

以下の文章にもっとも良く合致する OCL 式を選択肢の中から、A の問題はひとつ、B の問題はふたつを選択してください。



- A. ある会社で結婚している社員の集合
 1. context 会社 self.社員.select(p | p.夫.notEmpty)
 2. context 会社 self.社員.select(p | p.夫.notEmpty
or p.妻.notEmpty)
 3. context 会社 self.社員->select(p | p.夫 -> notEmpty)
 4. context 会社 self.社員->select(p | (p.夫 -> notEmpty)
or (p.妻 -> notEmpty))
- B. ある会社で社内結婚している社員のカップル数
 1. context 会社 (self.社員->select(p | p.夫.勤務先 = self))->size()
 2. context 会社 (self.社員->select(p | p.夫.妻=p))->size()
 3. context 会社 (self.社員->select(p | p.夫.勤務先->includes(self)))
->size()
 4. context 会社 (self.社員->select(p | p.夫.妻->includes(p)))
->size()

問題

S6 OCL の基本理解:

このクラス図に対する妥当な OCL 式をすべて選べ。

口座
残高:Integer
預金する(金額: Integer) 引き出す(金額:Integer) 残高確認(): Integer

1. context 口座::預金する(金額: Integer)
pre: 金額 < 残高
post: 残高 = 残高 + 金額
2. context 口座::預金する(金額: Integer)
post: 残高 = 残高 + 金額
3. context 口座::引き出す(金額: Integer)
pre: 残高 > 0
post: 残高 = 残高@pre - 金額
4. context 口座::引き出す(金額: Integer)
pre: 金額 <= 残高
post: 残高 = 残高@pre - 金額
5. context 口座::残高確認(): Integer
post: 残高
6. context 口座::残高確認(): Integer
post: result = 残高