姓名＿＿＿＿＿＿

学籍番号＿＿＿＿＿＿

学部＿＿＿＿＿＿

クラス＿＿＿＿＿＿

科目番号＿＿＿＿＿＿

担当教員＿＿＿＿＿＿

装

訂

線

# 大連理工大学

科目名：　オブジェクト指向技術 UML　問題種類：　A　試験形式：　閉　巻

所属学部：国際情報ソフトウェア学部　試験の実施日：2020 年　月　日

問題用紙合計ページ数　7

| | 一 | 二 | 三 | 四 | 五 | 六 | 七 | 八 | 九 | | 合計 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 配点 | 14 | 16 | 24 | 6 | 14 | 2 | 4 | 10 | 10 | | 100 |
| 得点 | | | | | | | | | | | |

| 得点 | |
|---|---|

一、以下の各説明に最もあてはまる語句を選び、A～D の記号で答えよ。

A (1) クラスから生成されたオブジェクト。

(A) インスタンス　(B) プロパティ

(C) カプセル　(D) インタフェース

D (2) データとそれに対する処理をまとめてモジュール化すること。

(A) 生成　(B) 継承　(C) 集約　(D) カプセル化

D (3) オブジェクトに操作の実行を依頼する仕組み。

(A) オーバーロード　(B) オーバーライド

(C) 集約　(D) メッセージパッシング

B (4) 子クラスが親クラスの性質を引き継ぐこと。また、その仕組み。

(A) 生成　(B) 継承　(C) 集約　(D) カプセル化

C (5) 全体とその構成部品の関係。

(A) 生成　(B) 継承　(C) 集約　(D) カプセル化

/ (6) 同じ名前のメソッドを複数宣言すること（互いに上書きしない）。

(A) オーバーロード　(B) オーバーライド

(C) 集約　(D) メッセージパッシング

] (7) 子クラスのメンバが親クラスのメンバを上書きすること。

(A) オーバーロード　(B) オーバーライド

(C) 集約　(D) メッセージパッシング

| (1) | (2) | (3) | (4) |
|---|---|---|---|
| (5) | (6) | (7) | |

| 得点 | |
|---|---|

二、各文の空欄にあてはまる最も適切な語句を A～D の記号で答えよ。

C (1) オブジェクトの 3 つの特性とは、状態、振る舞い、[　　]である。
(A) 操作　(B) 属性　(C) 識別性　(D) プロパティ

C (2) 開発者によるコードの読み書きを容易にするために、同一の処理に対して複数の記述方法を認めるなどのプログラミング言語の機能を[　　]と呼ぶ。
(A) メッセージパッシング　(B) インヘリタンス
(C) シンタックスシュガー　(D) セマンティクス

C (3) オブジェクト指向分析の[　　]モデルでは、クラスの名前、属性、振る舞いやクラス間の関連を確定する。
(A) 概念　(B) 分析　(C) 設計

I (4) クラス A がクラス B を具体化している時、クラス B はクラス A の[　　]である。
(A) 特化　(B) 汎化　(C) 実装　(D) 子クラス

C (5) クラス A がクラス B を[　　]するとき、クラス B はクラス A の親クラスである。
(A) 委譲　(B) 集約　(C) 継承　(D) 複合

A (6) クラス A がクラス B を継承するとき、両クラスの間に[　　]関係が成立する。
(A) is-a　(B) has-a　(C) part-of

B (7) [　　]関係は全体と部分の関係や所有関係を表現する。
(A) is-a　(B) has-a　(C) import

C (8) [　　]は強い集約関係を意味する。
(A) 継承　(B) 委譲　(C) 複合　(D) 実装

E (9) 別のインスタンスに処理の一部を依頼することを[　　]という。
(A) 継承　(B) 委譲　(C) 特化　(D) 汎化

C (10) 中身（実装）がないメソッドを[　　]メソッドという。
(A) 静的　(B) 動的　(C) 抽象　(D) 具象

I (11) 1 つのインスタンスが複数クラスのインスタンスとみなせる性質を[　　]という。
(A) mutability　(B) correlation　(C) covariant　(D) polymorphism

A (12) クラスの実装の一部で使われる型を可変にし、実行時に適切な実体を提供する仕組みを[　　]という。
(A) 総称　(B) 継承　(C) 実装継承　(D) 生成

F (13) シーケンス図において繰り返しを表現する複合フラグメントは[　　]である。
(A) seq　(B) loop　(C) opt　(D) par

I (14) 状態機械図において、ある状態に遷移したときに一度だけ実行される処理は[　　]アクションで示される。
(A) start　(B) entry　(C) begin　(D) do

A (15) フレームワークのうち、個々のプロジェクトの開発者が固有のコードを記述することができる可変部分を[　　]スポットと呼ぶ。
(A) ホット　(B) フローズン　(C) スイート　(D) 仮想

C (16) 既存のソフトウェアの外部的な振る舞いを変えることなく実装を改善する作業を[　　]と呼ぶ。
(A) リバースエンジニアリング　(B) プログラミング
(C) リファクタリング　(D) フォワードエンジニアリング

| (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) |
|---|---|---|---|---|---|---|---|
| (9) | (10) | (11) | (12) | (13) | (14) | (15) | (16) |

| 得点 | |
|---|---|

三、以下の各文は正しいか。
正しい場合は○、誤っている場合は×で答えよ。

× (1) オブジェクト指向プログラミング言語として初めて作られた言語は Java である。

○ (2) クラスの利点の 1 つとして、プログラムで扱うデータと、そのデータに対する操作を 1 つのモジュールとしてまとめられることが挙げられる。

› (3) クラス内部のフィールドやメソッドは、他のクラスから利用しやすいようにすべて公開することが望ましい。

waterfall Model

× (4) オブジェクト指向ソフトウェア開発は、ウォーターフォールモデルに従うのが一般的である。

对、 (5) カプセル化の利点として、そのクラスの使い方がわかりやすくなることが挙げられる。 UMLソフトウェア開発時の共通語彙，設計の意志決定の伝達が容易

举例

○ (6) UML を使う利点として、設計の意思伝達が容易になることが挙げられる。

错 (7) 名詞抽出法は、要求仕様やユースケース記述の名詞に着目して、各クラスのメソッドを明らかにする手法である。

对 (8) クラス図において、内部実装を持たないインタフェースの名前は斜体で記述する。

错 (9) 親クラスの機能を拡張することを目的に継承を用いるべきではない。

对、 (10) 抽象メソッドを持つクラスはインスタンス生成できない。

错 (11) インタフェース導入の利点として、子クラスや子孫クラスにおいてその内容の実装をしなくてもよくなることが挙げられる。

对、 (12) インタフェースを介してメソッド参照することで、直接参照するよりも疎結合になるという利点がある。

错 (13) 一般的に、アクティビティ図はシステム全体の処理の流れを表現する。

○ (14) アクティビティ图において、フォークノードやジョインノードを用いることで並行動作を示すことができる。

错 (15) シーケンス図では、実際に発生するメッセージのやり取りを省略してはならない。

× (16) シーケンス図においては条件分岐を表現する手段はない。

○ (17) コミュニケーション図は、オブジェクト間の相互作用を表現する。

错 (18) 状態機械図における do アクティビティは、その状態の間に一度だけ実行される処理である。

| | | | | | |
|---|---|---|---|---|---|
| (1) | (2) | (3) | (4) | (5) | (6) |
| (7) | (8) | (9) | (10) | (11) | (12) |
| (13) | (14) | (15) | (16) | (17) | (18) |

| 得点 | |
|---|---|

三（前頁の続き）、以下の各文は正しいか。
正しい場合は○、誤っている場合は×で答えよ。

対 (19) 状態機械図においては、1 つの状態が内部状態を含むこともできる。

対 (20) UML を使ったソフトウェア開発においては、複数の図の間で矛盾が発生しないように注意しなければならない。整合性

错 (21) UML を使ったソフトウェア開発においては、複数の図が表現するものの粒度を合わせる必要がある。

对 (22) リスコフの置換原則は、共通の親クラスを持つクラスはいつでも互いに置換可能でなければならないという原則である。

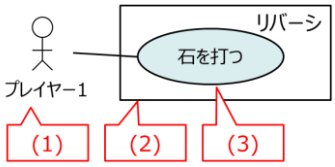
错 (23) オープン・クローズドの原則は、クラスのメンバは必要最低限のもののみ外部に公開し、それ以外は隠蔽するべきであるという原則である。

对 (24) リファクタリングにおいては、振る舞いの不変性を保証するため、リファクタリング適用の前後にテストを行わなければならない。

| (19) | (20) | (21) | (22) | (23) | (24) |
|---|---|---|---|---|---|
| | | | | | |

| 得点 | |
|---|---|

四、以下に示したユースケース図の各部の名称を
A～D の記号で答えよ。




A (1)の選択肢：
(A) アクタ　(B) クラス　(C) ステレオタイプ　(D) メッセージ

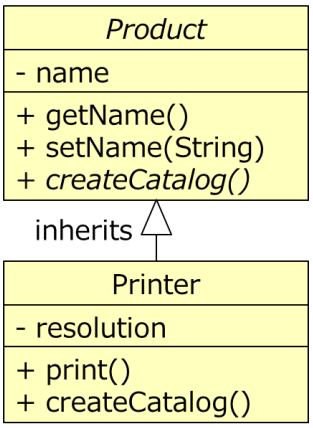
B (2)の選択肢：
(A) クラス　(B) システム境界　(C) アクタ　(D) メッセージ

B (3)の選択肢：
(A) クラス　(B) ユースケース　(C) ユースケース記述　(D) アクタ

| (1) | (2) | (3) |
|---|---|---|
| | | |

| 得点 | |
|---|---|

五、右に示したクラス図について、以下の各文は正しいか。正しい場合は○、誤っている場合は×で答えよ。ただし、*Product*(クラス名)と *Product.createCatalog()*は斜体表記であることに注意せよ。




× (1) Product クラスで宣言されているフィールドは3つである。 2

○ (2) Printer クラスは Product クラスを継承している。

○ (3) Printer 型のインスタンスは getName()メソッドを持つ。

× (4) 任意のクラスにおいて、Product 型変数 p があるとき、p.name というフィールドアクセスは可能である。

× (5) 任意のクラスにおいて、Product 型変数 p があるとき、p.print()というメソッド呼び出しは可能である。

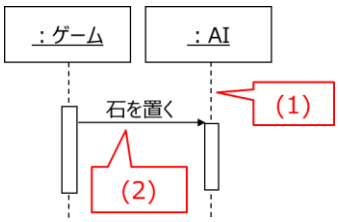
○ (6) 任意のクラスにおいて、Printer 型変数 p があるとき、p.createCatalog()というメソッド呼び出しは可能である。

× (7) 任意のクラスにおいて、Product 型変数 p があるとき、p.createCatalog()というメソッド呼び出しは可能である。抽象函数不能调用

| (1) | (2) | (3) | (4) |
|---|---|---|---|
| (5) | (6) | (7) | |

| 得点 | |
|---|---|

六、右に示したシーケンス図の各部の名称を A～D の記号で答えよ。




C (1)の選択肢：

(A) アクタ　(B) メッセージ　(C) 生存線　(D) 活性区間

A (2)の選択肢：

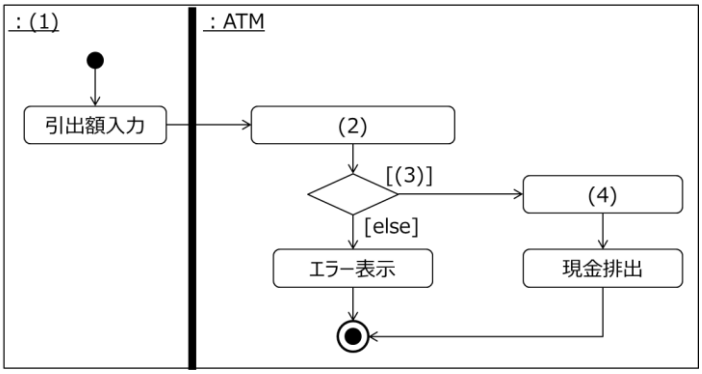
(A) メッセージパッシング　(B) 生成メッセージ

(C) 戻りメッセージ　(D) 非同期メッセージ

| (1) | (2) |
|---|---|

| 得点 | |
|---|---|

七、下のアクティビティ図は、ATM の預金引出処理を簡易的に表現したものである。以下の仕様に合うように、図中の空欄を埋めよ。A～D の記号を使って答えること。




- まず、利用者は引出額を入力する。
- 次に、ATM は預金残高を確認する。
- 預金残高が引出額以上の場合、預金引出処理、現金排出を行い、終了する。
- そうでない場合、エラーを表示し、終了する。

B (1)の選択肢：

(A) 引出額　(B) 利用者　(C) 入力　(D) ATM

B (2)の選択肢：

(A) 預金引出処理　(B) 預金残高確認

(C) エラー表示　(D) 終了

B (3)の選択肢：

(A) 預金残高＞引出額　(B) 預金残高≧引出額

(C) 預金残高＜引出額　(D) 預金残高≦引出額

A (4)の選択肢：

(A) 預金引出処理　(B) 預金残高確認

(C) エラー表示　(D) 終了

| (1) | (2) | (3) | (4) |
|---|---|---|---|
| | | | |

| 得点 | |
|---|---|

八、Java 言語に関して、以下の各問に A～D の記号で答えよ。

(1) クラス外部から参照不可能であることを示すクラスメンバの可視性はどれか。

(A) public (B) default (C) protected (D) private

(2) 静的なクラスメンバであることを示すキーワードはどれか。

(A) final (B) const (C) extends (D) static

(3) クラスの継承を行う際に使用するキーワードはどれか。

(A) extends (B) final (C) override (D) implements

(4) 抽象クラスを宣言する際、クラスに付加するキーワードはどれか。

(A) abstract (B) final (C) extends (D) implements

(5) 親クラスのメンバを参照する際に使用するキーワードはどれか。

(A) override (B) overload (C) super (D) this

(6) オブジェクトの型を調べる際に使用する演算子はどれか。

(A) is (B) instanceof (C) try (D) this

(7) 例外処理において、例外を発生する可能性がある内容を記述するブロックの前に記述するキーワードはどれか。

(A) try (B) catch (C) finally (D) do

(8) オブジェクトの不存在を意味するキーワードはどれか。

(A) empty (B) this (C) null (D) zero

(9) 値として true か false のみを持つデータ型はどれか。

(A) bool (B) boolean (C) logic (D) binary

(10) 浮動小数点数を表すデータ型はどれか。

(A) int (B) double (C) long (D) short

| (1) | (2) | (3) | (4) | (5) |
|---|---|---|---|---|
| (6) | (7) | (8) | (9) | (10) |

| 得点 | |
|---|---|

九、以下の各文が示す適用目的に最も合致する GoF のデザインパターンを A～D から選べ。

(1) オブジェクトが状態変化したときに、それに応じた処理を行う。

(A) Iterator (B) Singleton (C) Observer (D) Visitor

(2) 配列内部の要素に対して特定の順番で処理を行う。

(A) Observer (B) Iterator (C) Builder (D) Template method

(3) 複雑な処理を行うときにシンプルな窓口を用意する。

(A) Facade (B) Visitor (C) Singleton (D) Prototype

(4) 機能のクラス階層と実装のクラス階層を分離する。

(A) Template method (B) Strategy (C) Bridge (D) Visitor

(5) 木構造等の階層的なデータ構造を表現する。

(A) Singleton (B) Iterator (C) Composite (D) Interpreter

| (1) | (2) | (3) | (4) | (5) |
|---|---|---|---|---|