Java5

Sun Certified Programmer for Java 2 Platform 5.0 Upgrade Exam
Java5.0 でだいぶ機能追加があったけれど、丁寧に説明してくれてます。
ただ、試験範囲から外れちゃったところも多いので、こちらの本もあわせて読んでおくとよいかも (もったいないか ?)

http://suned.sun.co.jp/JPN/certification/progdetails.html

チェックポイント

宣言、初期化、スコープ

■クラス（抽象クラス、およびあらゆる形式の PDFJ::Text=HASH(0xa41fb30) を含む）、インタフェース、列挙型を宣言し、パッケージおよびインポート文（static インポートを含む）を正しく使用しているコードを書く。

■インナークラス

インナークラス

■クラス、インタフェース、列挙型を宣言

クラス、インタフェース、列挙型を宣言

■静的インポート

静的インポート

■型と変数

型と変数

■列挙型

列挙型

・プリミティブ型、配列型、列挙型、オブジェクトを static、インスタンス変数、ローカル変数として宣言、初期化、使用するコードを書く。変数名に正しい識別子を使用する。

■可変長引数

可変長引数

・static メソッドと static でないメソッドの両方を宣言するコードを書く。この場合、JavaBeans の命名規則に準拠したメソッド名を使用する。可変長引数リストを宣言、使用するコードを作成する。

■共変戻り値

共変戻り値

・与えられたコード例で、あるメソッドが別のメソッドを正しくオーバーライドまたはオーバーロードしているかを確認し、そのメソッドに対する正しい戻り値（共変戻り値を含む）を識別する。

■ SJC-P コンストラクタ

・与えられたクラスおよびスーパークラスに対し、1 つまたは複数のクラスに対するコンストラクタを作成する。与えられたクラスに対し、デフォルトコンストラクタが作成されるかどうかを判断し、作成される場合はそのコンストラクタのふるまいを特定する。与えられたネストまたはネストされていないクラスに対し、そのクラスをインスタンス化するコードを書く。

フロー制御

■ SJC-P 選択

SJC-P 選択

・if 文または switch 文を実装するコードを書いて、それらの文に対して正しい式を識別する。

■ SJC-P 繰返し

SJC-P 繰返し

・for、拡張 for ループ (for-each)、do、while、ラベル、break、continue など、あらゆる型のループや反復子を実装するコードを書き、ループ実行中と実行後のループカウンタ変数の値を説明する。

■ SJC-P 例外

SJC-P 例外

・例外と例外処理句 (try、catch、finally) を使用し、例外をスローするメソッドおよびそのメソッドをオーバーライドするメソッドを宣言するコードを書く。
・次のいずれかがスローされる結果になる状況を認識できる。
(ArrayIndexOutOfBoundsException、ClassCastException、 IllegalArgumentException、IllegalStateException、NullPointerException、 NumberFormatException、AssertionError、ExceptionInInitializerError、 StackOverflowError、NoClassDefFoundError) これらのうちのどれが仮想マシンでスローされるかを知っており、その他については、プログラムでスローすべき状況を認識できる。

API コンテンツ

■ SJC-P オートボクシング

SJC-P オートボクシング

・プリミティブラッパークラス（Boolean 型、Character 型、Double 型、Integer 型など）、autoboxing/unboxing を使用するコードを書く。String、StringBuilder、StringBuffer クラス間の違いを説明する。

■ SJC-P ファイル操作

SJC-P ファイル操作

■ SJC-P 直列化

SJC-P 直列化

・ファイルシステムのナビゲーション、ファイルの読み込み、ファイルの書き込みを含むシナリオが与えられたとき、java.io パッケージのクラスのうち、BufferedReader、BufferedWriter、File、FileReader、FileWriter、PrintWriter を使用して（組み合わせを含む）正しいソリューションを作成する。 java.io パッケージの API のうち、DataInputStream、DataOutputStream、FileInputStream、FileOutputStream、ObjectInputStream、ObjectOutputStream、Serializable を使用してオブジェクトをシリアライズまたはデシリアライズするコードを書く。さらに、 transient 変数および private 指定された readObject メソッ

ドと writeObject メソッドを宣言して使用する Serializable クラスを作成する。シナリオやコード例が与えられたとき、デシリアライズ中、オブジェクトの継承チェーンでコンストラクタが呼び出されるか、呼び出されるとするといつどのコンストラクタが呼び出されるかを知っている。

■ SJC-P **書式**

SJC-P 書式

■ SJC-P **日付と数値**

SJC-P 日付と数値

- java.text パッケージの標準 J2SE API を使用して、特定のロケール用の日付、数値、通貨の値の正しい書式化または解析を行う。与えられたシナリオで、デフォルトロケールまたは特定のロケールを使用する場合に適切なメソッドを特定する。java.util.Locale クラスの目的と用途を説明する。

■ SJC-P **ストリーム解析** (Scanner)

- java.util パッケージおよび java.util.regex パッケージの標準 J2SE API を使用して文字列またはストリームを書式化または解析するコードを書く。文字列に対しては、Pattern クラスと Matcher クラスおよび String.split メソッドを使用したコードを書く。マッチングの正規表現パターン（.（ドット）、*（アスタリスク）、+（プラス）、?、\d、 \s、\w、[]、() に限定）を認識、使用できる。*、+、? は最長マッチの修飾子に限られ、括弧演算子はグルーピングメカニズムにのみ使用し、マッチング中の内容の捕捉には使用しない。ストリームに対して、Formatter クラスと Scanner クラスおよび PrintWriter.format/printf メソッドを使用したコードを書く。書式文字列で書式化パターン（%b、%c、%d、%f、%s に限定）を認識、使用できる。

並行性

■ SJC-P **並行性**

SJC-P 並行性

- java.lang.Thread と java.lang.Runnable の両方を使用して新しいスレッドを定義、インスタンス化、開始するコードを書く。
- 存在し得るスレッドの状態を認識し、スレッドの状態がどのように変化できるかを識別する。

オブジェクト指向コンセプト

- 密なカプセル化、弱い結合度、高い凝集度を実装するコードを書いて、利点を説明する。
- 与えられたシナリオで、ポリモフィズムを使用するコードを書く。さらに、キャストが必要な場合を識別し、オブジェクト参照のキャストに関するコンパイラエラーとランタイムエラーを識別する。
- コンストラクタ、インスタンス変数または static 変数、インスタンスメソッドまたは static メソッドに関する継承の修飾子の効果を説明する。
- 与えられたシナリオで、オーバーライドまたはオーバーロードされたメソッドを宣言／呼び出すコード、およびオーバーライドまたはオーバーロードされたスーパークラスのコンストラクタを宣言／呼び出すコードを書く。

コレクション / ジェネリックス

- 与えられた設計シナリオで、その設計を正しく実装するにはどのコレクションクラス／インタフェースを使用するかを判断する。(Comparable インタフェースの使用を含む)
- Collections API のうち、特に、Set ＜ E ＞、List ＜ E ＞、Queue ＜ E ＞、Map ＜ K,V ＞（" ＜ "," ＞ " は半角）インタフェースとその実装クラスのジェネリックスを使用するコード

を書く。ジェネリックスを使用しない Collections API の制限事項、およびジェネリックスを使用するためにコードをリファクタリングする方法を知っている。

■ SJC-P ジェネリックス 多態性

SJC-P ジェネリックス 多態性

■ SJC-P ジェネリックス 型パラメータ

・総称型 (Generic type)

SJC-P ジェネリックス 型パラメータ

・クラス／インタフェース宣言、インスタンス変数、メソッド引数、戻り値の型に型パラメータを正しく使用するコードを書き、ジェネリックスを使用したメソッドまたはワイルドカードの型を使用するメソッドを書いて、これらの 2 つのアプローチ間の類似点と相違点を説明する。
・java.util パッケージの機能を使用して、リストのソート、バイナリ検索、リストから配列への変換など、リストの操作を行うコードを書く。リストと配列のソートには java.util.Comparator と java.lang.Comparable インタフェースを使用する。さらに、ソートに対するプリミティブラッパークラスと java.lang.String の「自然順 (natural ordering)」の効果を識別する。

Java 言語の基礎

・与えられたコード例とシナリオで、正しいアクセス修飾子、パッケージ宣言、インポート文を使用して例のコードと相互作用する（アクセスを行う、または継承関係を持つ）コードを書く。
・JAR ファイルの内外にあるクラスの完全修飾名が与えられたとき、そのクラスの正しいディレクトリ構造を構成する。コード例とクラスパスが与えられたとき、そのクラスパスがコードのコンパイルに支障を来たすかどうかを判断する。

その他参考

■ Java アノテーション
