

1、A,C,D

問題 パッケージに関する説明として、正しいものを選びなさい。(3つ選択)

- ☒ A. 名前空間を提供する。
- ☐ B. パッケージ名にはドメイン名を逆にしたものを使用しなければならない。
- ☒ C. アクセス制御を提供する。
- ☒ D. クラスの分類を可能にする。
- ☐ E. パッケージ宣言はソースファイルの1行目に記述する。

パッケージに関する問題です。

パッケージとは次の3つの機能を提供するものです。

- ・名前空間を提供し、名前の衝突を避ける
- ・アクセス修飾子と組み合わせてアクセス制御機能を提供する
- ・クラス分類を可能にする

大規模なソフトウェアが複数存在するエンタープライズシステムでは、数え切れないほどのクラスが存在します。このとき、単純なクラス名だけでは、ほかのシステムのクラスと名前が衝突してしまいます。このような事態を避けるために、コンパイラやJVMは、クラスを「パッケージ名+クラス名」の完全修飾クラス名で扱います(選択肢A)。

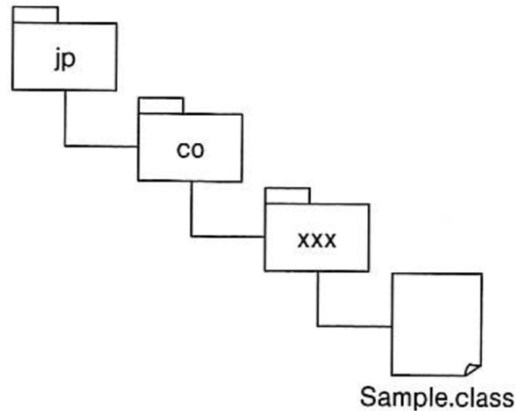
このようにパッケージは名前の衝突を避けるために使うものなので、パッケージ名はできるだけ一意なものが推奨されます。

そこで、慣習としてパッケージ名にはドメイン名を逆にしたものが利用されます。たとえばxxx.co.jpであれば、jp.co.xxxという具合です。もちろん、これはあくまでも慣習であって決まりではありません。ドメイン名以外のものも使用できます(選択肢B)。

情報隠蔽の実現にはパッケージが欠かせません。パッケージがあるおかげで、公開するものと非公開にするものを明確に分け、アクセス制御が可能になります(選択肢C)。

パッケージは、ディレクトリ構造とマッピングされます。例えば、jp.co.xxx.Sampleという完全修飾クラス名を持つクラスは、次のようなディレクトリに配置されます。

【パッケージ化されたクラスの配置】



このようにパッケージとディレクトリ構造がマッピングされると、数多くあるクラスを分類整理することができ、ソフトウェアの管理が容易になります。(選択肢D)

書式

package パッケージ名;

パッケージ宣言は、ソースファイルの先頭行で宣言しなければいけません。
パッケージ宣言よりも前に記述できるのはコメントだけです。(選択肢E)

以上のことから、選択肢A,C,Dが正解です。

なお、パッケージ宣言をしなかったクラスは無名パッケージに所属していると見なされます。
無名パッケージは学習用に用意されたものであり、実際の開発では推奨されません。

2、A,B

問題 インポート宣言に関する説明として、正しいものを選びなさい。(2つ選択)

- ☒ A.java.langパッケージに所属するクラスは、インポート宣言を省略できる。
- ☒ B.同じパッケージに所属するクラスは、インポート宣言を省略できる。
- ☐ C.異なるパッケージに所属するクラスを利用する場合には、必ずインポート宣言をしなければならない。
- ☐ D.インポート宣言は、ソースファイルの先頭行から列挙しなければならない。
- ☐ E.インポート宣言では、利用したいクラスが所属するパッケージ名までを指定する。

パッケージのインポートに関する問題です。

コンパイラやJVMはクラスを完全修飾クラス名でしか扱えません。

パッケージ宣言しなかった場合ですら、そのクラスはデフォルトのパッケージ(無名パッケージ)に所属していると思なされます。

そのため、次のようにソースコードでも完全修飾クラス名でクラスを指定しなければいけません。

例：クラスを完全修飾クラス名で指定した場合

```
1. public class Main {  
2     public static void main(String[] args) {  
3         java.lang.String str = "100";  
4         int val = java.lang. Integer.parseInt(str);  
5         java.math.BigDecimal decimal = new java.math.BigDecimal(val);  
6         System.out.println(decimal. intValue());  
7     }  
8. }
```

一見してわかるとおり、完全修飾クラス名でプログラムを記述すると、とても冗長で読みにくいコードになります。

そこで、パッケージ名を省略し、クラス名だけで記述できるようインポート宣言をします(次のコードの1～3行目)。

例：インポート宣言した場合

```
1. import java.lang. String;  
2. import java.lang. Integer;  
3. import java.math. BigDecimal;  
4.  
5. public class Main {  
6     public static void main(String[] args) {  
7         String str = "100";  
8         int val = Integer.parseInt(str);  
9         BigDecimal decimal = new BigDecimal(val);  
10        System.out.println(decimal.intValue());  
11    }  
12. }
```

このように、本来は完全修飾クラス名で記述しなければならないものを、省略表記できるようにするのがインポート宣言です。

なお、ソースコードを完全修飾クラス名で記述するのであれば、インポート宣言は必要ではありません(選択肢C)。

ただし、java.langパッケージに所属するクラスは、インポート宣言する必要はありません。

java.langパッケージは基本的なクラスがまとめられたパッケージであり、このパッケージに所属するクラスは頻繁に利用するため、省略することができます。

加えて、同じパッケージに属するクラスも省略可能です(選択肢A、B)。

インポート宣言は、パッケージ宣言の後ろ、クラス宣言の前に行います。

パッケージ宣言が先頭であることを忘れないようにしましょう(選択肢D)。

例：インポート宣言した場合

```
1. package ex3;
2.
3. import java.math.BigDecimal;
4.
5. public class Main {
6.     public static void main(String[] args) {
7.         String str = "100";
8.         int val = Integer.parseInt(str);
9.         BigDecimal decimal = new BigDecimal(val);
10.        System.out.println(decimal.intValue());
11.    }
12. }
```

インポート宣言は、省略表記したいクラスの完全修飾クラス名を記述します。

クラス名の部分は、アスタリスク「*」を使って「java.util.*」のようにワイルドカード表記が可能で、パッケージ名を省略することもできます(選択肢E)。

3、B,C

問題 例外に関する説明として、正しいものを選びなさい。(2つ選択)

- ☐ A.例外が発生する可能性のある処理を「catch」ブロックで括る。
- ☒ B.複数の例外を個別に処理可能。
- ☒ C.例外をcatchして、新しい例外をスローし直すことを「例外のラップ」という。
- ☐ D.例外が発生しない場合のみ、finallyブロックに処理が移行する。
- ☐ E.例外が発生した場合、呼び出し元へエラーコードを返す。

例外に関する問題です。

プログラム動作時における予期せぬ事象のことを例外といいます。

例外を受け取り、対処するには、以下の書式でtry-catchを使用します。

書式

```
try {
    // 例外が発生する可能性のある処理
} catch(補足したい例外の型 変数名) {
    // 例外が発生した場合の処理
}
```

例外は1つのみではなく、複数の例外を個別処理することができます。

また、例外をキャッチして新しい例外をスローし直すことを例外のラップといいます。

よって選択肢B,Cが正解です。

各選択肢については、以下の通りです。

- A. 例外が発生する処理は「try」ブロックで括ります。
- D. 「finally」ブロックの処理は、例外発生の有無に関わらず実行されます。
- E. 例外が発生した場合、呼び出し元へは例外が送出されます。

4、 A,C

問題 例外クラスに関する説明として、正しいものを選びなさい。(2つ選択)

- ☒ A. `IndexOutOfBoundsException`→配列や文字列などのインデックス番号が範囲外である。
- ☐ B. `IllegalArgumentException`→そのクラスに存在しないメソッドを呼び出そうとした。
- ☒ C. `NullPointerException`→オブジェクトがnullなのにメソッド実行などをしようとした。
- ☐ D. `IOException`→不正または不適切な引数をメソッドに渡した。
- ☐ E. `NoSuchMethodException`→入出力処理に失敗した。

例外クラスには様々なものがあります。
代表的な例外クラスは以下の通りです。

・`java.lang.Exception`継承クラス

クラス名	内容
<code>ClassNotFoundException</code>	指定されたクラスがロードできない
<code>IOException</code>	入出力処理に失敗した
<code>NoSuchMethodException</code>	そのクラスに存在しないメソッドを呼び出そうとした
<code>SQLException</code>	データベースアクセス処理に失敗した

・`java.lang.RuntimeException`

クラス名	内容
<code>ClassCastException</code>	キャストできないクラスをキャストしようとした。
<code>IllegalArgumentException</code>	不正または不適切な引数をメソッドに渡した。
<code>IndexOutOfBoundsException</code>	配列や文字列などのインデックス番号が範囲外である。
<code>NullPointerException</code>	オブジェクトがnullなのにメソッド実行などをしようとした。

・`java.lang.Error`継承クラス

クラス名	内容
<code>OutOfMemoryError</code>	実行に必要なメモリが不足している。
<code>StackOverflowError</code>	再帰が多すぎてスタック領域があふれてしまった。
<code>NoClassDefFoundError</code>	コンパイル時に存在していた依存クラスが存在しない。

よって、選択肢A,Cが正解です。

問題 APIに関する説明として、正しいものを選びなさい。(2つ選択)

- ☐ A.使用するクラスをパッケージで宣言。
- ☒ B.使用方法は、クラスと同様にインスタンス化やstaticフィールド、メソッドを直接呼び出して使用する。
- ☒ C.「ライブラリ」ともいう。
- ☐ D.Java標準のAPIは、機能ごとにはパッケージされていない。
- ☐ E.APIは「JAVAファイル」として提供される。

APIに関する問題です。

OSや開発環境によって、あらかじめ提供されるアプリケーションで頻繁に用いられる汎用的な機能の実装のことをAPIといいます。APIはライブラリとも呼ばれ、「JAR」ファイル形式で提供されます。利用する場合は、通常のクラスと同様、利用したいAPIが含まれるパッケージのimport宣言をし、インスタンス生成をして利用します。

よって、選択肢B,Cが正解です。