

SA31 課題 3

課題名称

オーバーライドとオーバーロードの確認

課題区分

訓練

納期

2025 年 12 月 3 日(水)

12/3 は課題確認の時間とする。

内容

次の二種のプログラムを JavaLesson2 プロジェクト内に作成する。

[1] 立方体と直方体

1.5、3.2、5.8、7.4、9.7 の各値を一辺とする立方体、および、直方体の表面積と体積を計算、表示するプログラムを作成する。なお、この問題で作成するクラスは、以下の通りであり、JavaLesson2 のプロジェクト内の以下のパッケージに作成する。

local.hal.sa31.java.lesson2.asgmt1

[CalcCubeAndCuboid](#)

フィールドに格納された各辺の値をもとに表面積と体積を計算するクラスである。具体的には以下の仕様となる。

1. メンバは以下のものとする。これらメンバのアクセス修飾子、フィールドの型、メソッドの戻り値の型、引数の有無も含めて引数の定義などは、各自で考える。
 - 高さを表すフィールド _height。
 - 幅(横の長さ)を表すフィールド _width。
 - 奥行き(縦の長さ)を表すフィールド _depth。
 - コンストラクタ。
 - 直方体、もしくは、立方体の表面積を取得するメソッド getArea()。
 - 直方体、もしくは、立方体の体積を取得するメソッド getVolume()。
2. CalcCubeAndCuboid クラスは、new するときに高さ、幅、奥行きの 3 辺を受け取るようにする。ただし、幅と奥行きは省略でき、その場合は、1 辺の値のみ受け取り、立方体として扱う。

[CalcFromOneEdge](#)

main() メソッドが含まれた実行クラスである。具体的には以下の仕様となる。

1. 実行内容として、1 辺の長さとして、1.5、3.2、5.8、7.4、9.7 それぞれの場合を想定して計算を行なっていく。
2. 立方体の場合は、上記それぞれの値に対して表面積、体積を計算し、表示する。
3. 直方体の場合は、上記のそれぞれの値を高さとし、4 倍したものを幅、8 倍したものをお行きとして表面積と体積を計算し、表示する。

[表示例](#)

表示例は次の通り。なお、小数計算では、どうしても誤差が発生してしまい、結果としておかしなものが表示されているものがあるが、今回はその点は不問とする。これらの誤差を考慮した丸め処理については、今後紹介していく。

■■■■ 一辺 1.5 の計算結果

立方体の表面積: 13.5 | 体積: 3.375

横 6.0 奥行き 12.0 の直方体の表面積: 198.0 | 体積: 108.0

■■■■ 一辺 3.2 の計算結果

立方体の表面積: 61.44000000000001 | 体積: 32.76800000000001

横 12.8 奥行き 25.6 の直方体の表面積: 901.1200000000002 | 体積:
1048.5760000000002

■■■■ 一辺 5.8 の計算結果

立方体の表面積: 201.84 | 体積: 195.112

横 23.2 奥行き 46.4 の直方体の表面積: 2960.319999999997 | 体積: 6243.584

■■■■ 一辺 7.4 の計算結果

立方体の表面積: 328.5600000000006 | 体積: 405.2240000000005

横 29.6 奥行き 59.2 の直方体の表面積: 4818.88 | 体積: 12967.168000000001

■■■■ 一辺 9.7 の計算結果

立方体の表面積: 564.54 | 体積: 912.6729999999998

横 38.8 奥行き 77.6 の直方体の表面積: 8279.92 | 体積: 29205.535999999993

[2] ドーナツ

注文したドーナツと価格、さらに、ドリンクとのセット価格を表示するプログラムを作成する。なお、この問題で作成するクラスは、以下の通りであり、JavaLesson2 のプロジェクト内の以下のパッケージに作成する。

local.hal.sa31.java.lesson2.asgmt2

Donuts

ドーナツを表すクラスである。メンバは以下のものとする。これらメンバのアクセス修飾子、フィールドの型、メソッドの戻り値の型、引数の有無も含めて引数の定義などは、各自で考える。

- ドーナツ名を表すフィールド _name。
- そのドーナツの単価を表すフィールド _price。
- そのドーナツの注文した個数を表すフィールド _quantity。
- コンストラクタ。

引数としてフィールドと同じものを受け取り、それらの値をフィールドに格納する。

- getTotalDonutsPrice() メソッド。
注文したドーナツの合計金額を得る。
- showOrder() メソッド。
注文したドーナツに関して、「○○が△△個で合計###円」と表示する。なお、

「〇〇」にはドーナツ名が、「△△」には注文した個数が、「###」には合計金額が表示される。

DonutsWithDrink

上記 Donuts クラスを継承したクラスである。メンバは以下のものとする。

- ドリンク名を表すフィールド _drinkName。
- ドリンクの金額を表すフィールド _drinkPrice。
- コンストラクタ。
処理に必要な引数を全て受け取る。
- getPriceWithDrink() メソッド。
ドリンク代も含めた金額を得るメソッド。
- showOrder() メソッド。
オーバーライドとなる。注文したドーナツに関して、「〇〇が△△個で合計###円」と表示させた上で、さらに、「ドーナツと **とのセットで合計\$\$\$円」と表示する。なお、「**」にはドリンク名が、「\$\$\$」にはドーナツ代とドリンク代の合計金額が表示される。

ShowDonutsPrice

main() メソッドが含まれた実行クラスである。Donuts クラスと DonutsWithDrink クラスをそれぞれ 1 個ずつ new して、次の実行結果となる。

実行結果

オールドファッショング 3 個で合計 360 円
チョコファッショング 2 個で合計 280 円
ドーナツとアイスコーヒーとのセットで合計 630 円

提出物と提出先

1. アプリのデモ → 講師まで直接見せにくる
デモ時にソースコードも確認する。
2. 上記作成した 5 クラスのファイル。zip 化する必要はない。
→ Teams の課題