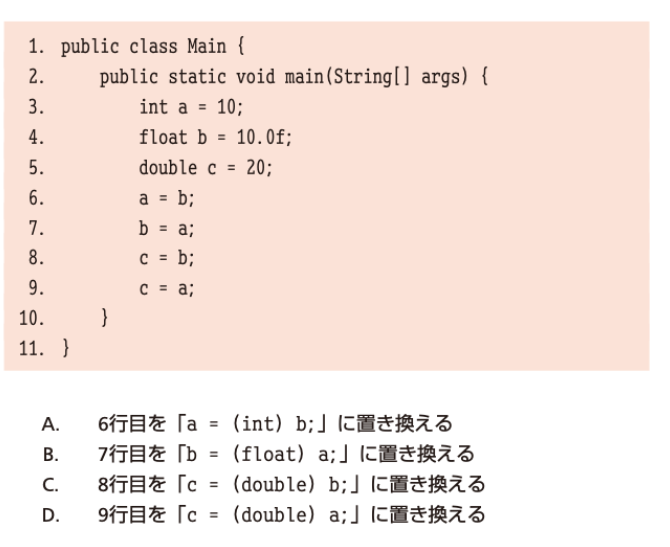
間違った問題１：



１．間違った原因

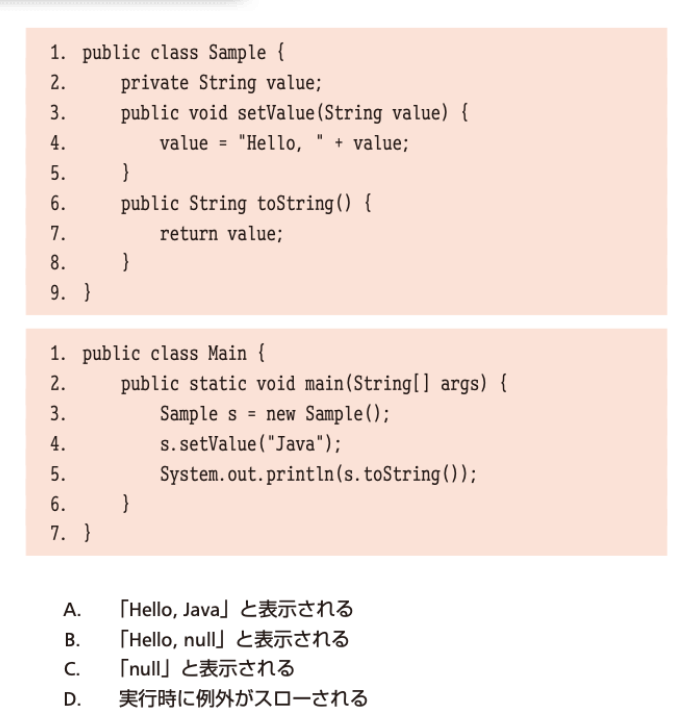
データ型の大きさを間違った。

２．正しい答え

A

大きいデータ型 から小さいデータ型を変換する時キャストが必要です。  
大きさ順まとめ（小さいから大きい）　int、float 、double

間違った問題2：



１．間違った原因

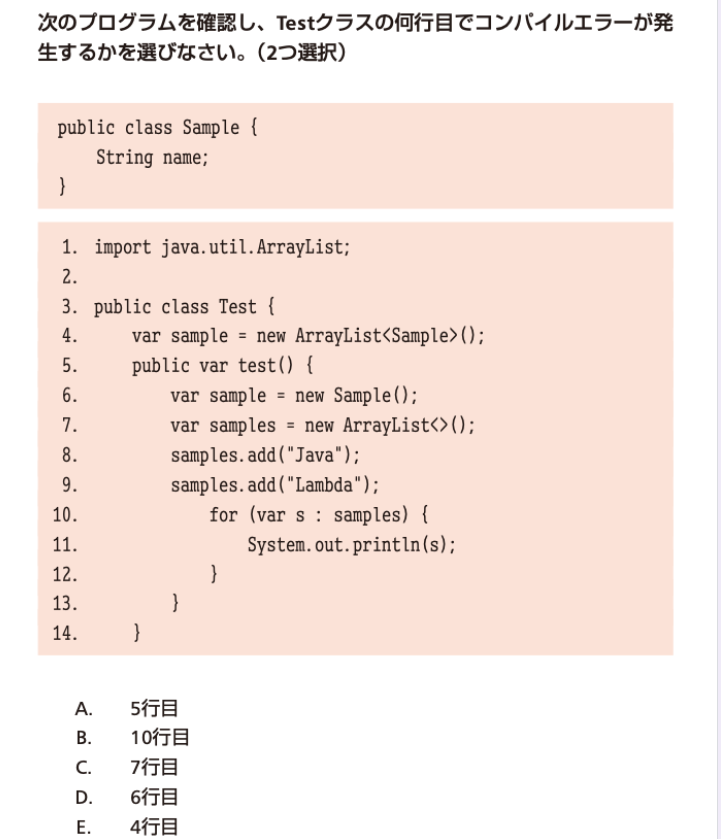
valuaがnullであるため、 NullPointerExceptionが発生と思いました

２．正しい答え

Ｃ

正常に実行できます。

間違った問題3：



１．間違った原因

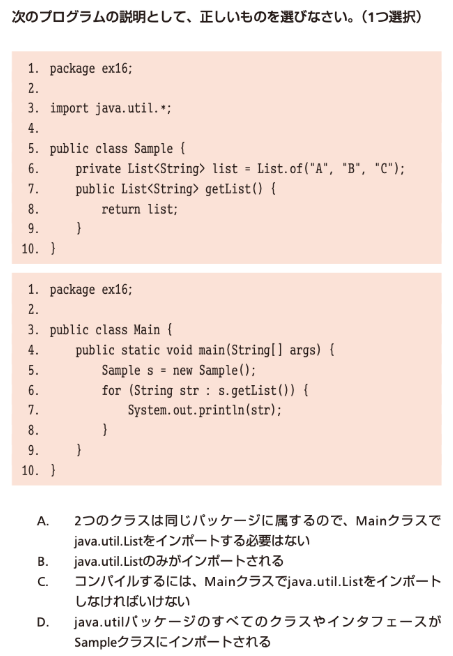
var はメンバ変数の宣言にも使えないことを忘れました。

２．正しい答え

AE

varはメソッドの戻り値と使えない、メンバ変数の宣言にも使えない。

間違った問題4：



１．間違った原因

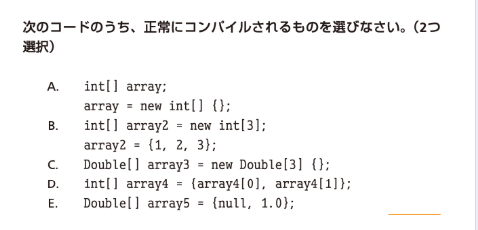
Mainクラスでjava.util.listをインポートする必要がない考えを用い、理由を見ていないままAが正解だと思いました。

２．正しい答え

D java.util.list.\*

[\*]はすべてのものをインストールするの表示です。

間違った問題5：



１．間違った原因

・コンパイルできないものを選択すると勘違いしました。

２．正しい答え

AE

正しい表現は:

① int[] array ;

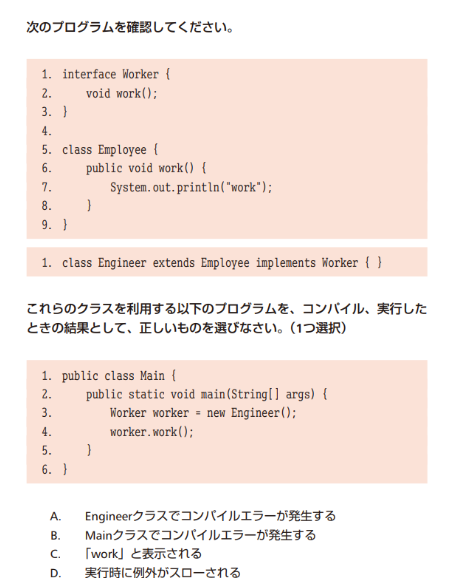
array=new int[]{1,2,3}

②int[] array = {1,2,3};

③int[] array = new int[]{1,2,3};

＊「Double」と書いているため参照型と見られ、nullもできます。Eが正解です。

間違った問題6：



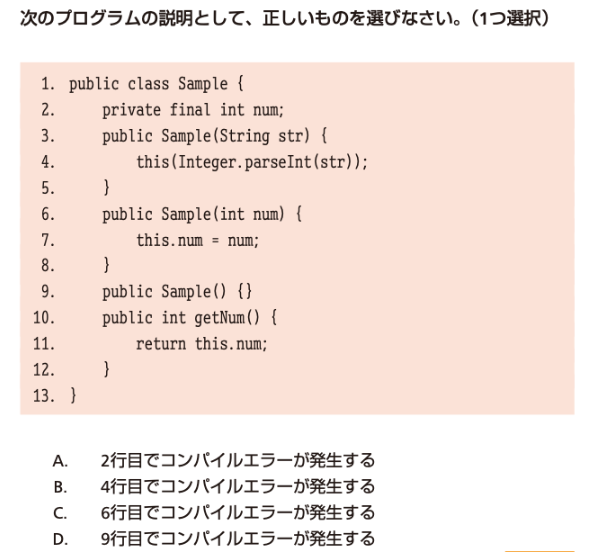
１．間違った原因

Engineerクラスがインターフェースのworkを実装しなければならないと思いました。

２．正しい答え

C、正常に実行できます。

間違った問題7：



１．間違った原因

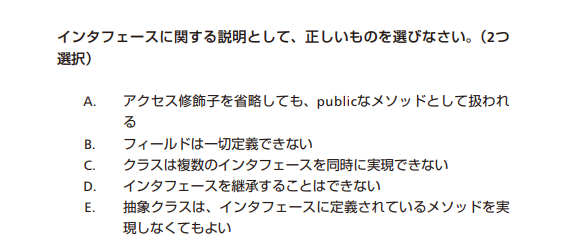
Finalで宣言した場合、すぐに値を設定しなければならないと思いました。

２．正しい答え

Ｄ

ここのint numはメンバ変数であるため、下のソースコードでを設定しなければならない。

間違った問題8：



１．間違った原因

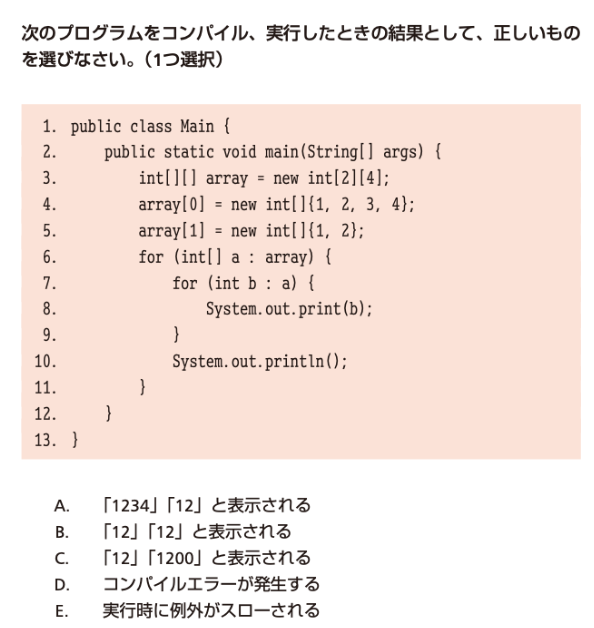
抽象クラスがインターフェースに定義されるメソッドを実現しなければならないと思いました。

２．正しい答え

AE

問題が書いている通り。

間違った問題9：



１．間違った原因

二重のfor文で読みつらい、分からなくなった。

２．正しい答え

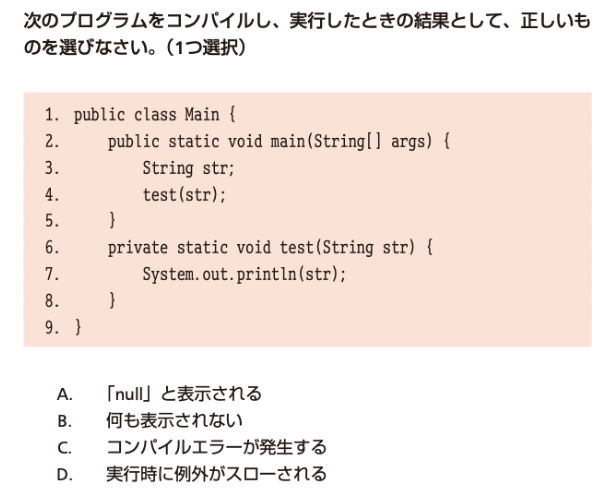
A

一回目の循環は0番０～３号を出力する

二回目の循環は1番０～１号を出力する

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0号 | 1号 | 2号 | 3号 |
| 0番 | 1 | 2 | 3 | 4 |
| 1番 | 1 | 2 | / | / |

間違った問題10：



１．間違った原因

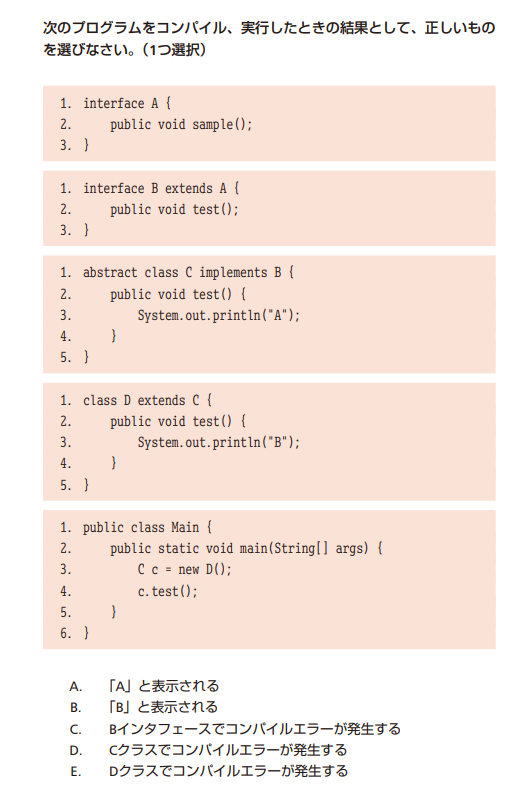
初期化でいないstrがnullで、使えると思いました。

２．正しい答え

Ｃ

4行目がエラー、3行目に初期化でいないもの使えません。

間違った問題11：



１．間違った原因

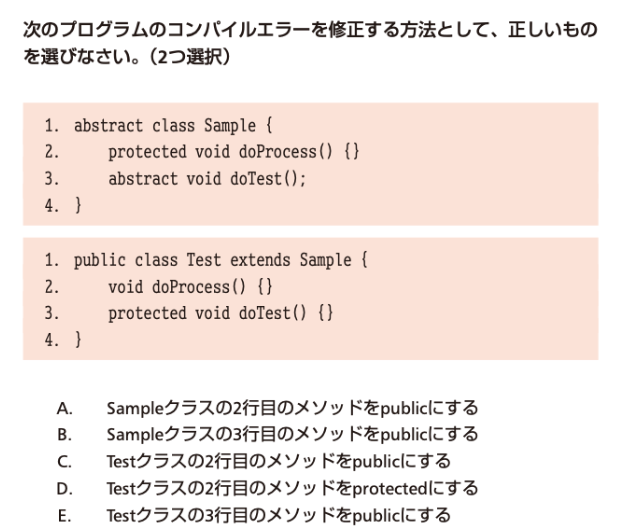
sampleが次の子クラスで実装していないと直ぐにエラーになると思いました。

２．正しい答え

Ｄ

一番下の子クラスで、実装もしていないからここでエラーが発生する。

間違った問題12：



１．間違った原因

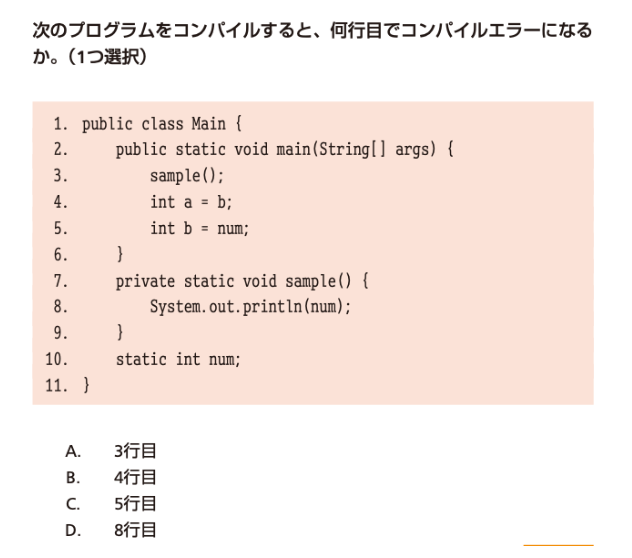
・親クラスにアクセス修飾子がない場合、子クラスは必ずpublicつけると思いまして、Cの選択肢で2行目のエラーを処解決したから、Dを見ていませんでした。

２．正しい答え

CD

アクセス修飾子が等しい範囲又は大きい方を使います。

間違った問題13：



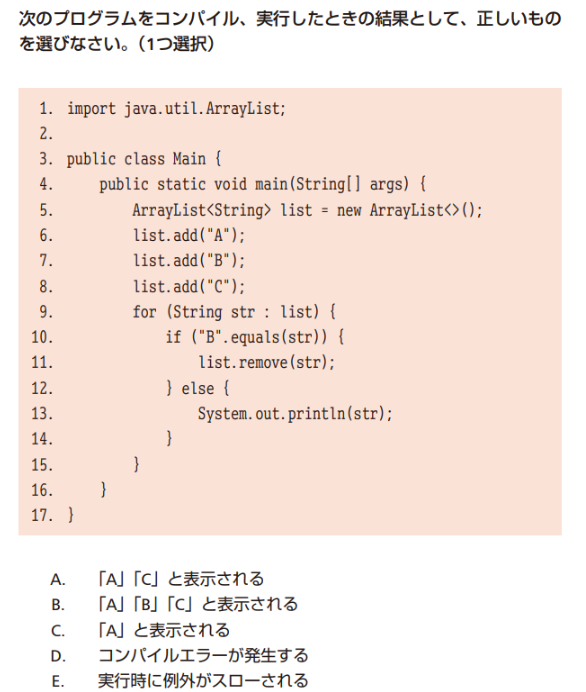
１．間違った原因

4行目でｂが宣言していないので使えないのはエラーの原因となると思いました。しかし不注意でここは５行目と勘違いました。

２．正しい答え

A

間違った問14：



１．間違った原因

ｂを削除された後ｃの場所が変わることを忘れました。

２．正しい答え

D

ｂを削除された後ｃの場所が変わって、Aのみが表示される。