問2　言語処理系（ソフトウェア）　(H27春-FE 午後問2)

【解答】

［設問1］　エ
［設問2］　a－ア
［設問3］　b－オ，c－エ

【解説】

言語処理系に関する問題である。言語処理系には，インタプリタやコンパイラなどがある。いずれも午前試験では必須の用語であり，その特徴なども過去に出題されているため，学習済みの受験者が多かったと推測している。したがって，設問1は確実に解いておきたい。設問2は仮想計算機の特徴を問う内容であるが，設問内容のポイントは設問文に記述されているので，詳細を知らなくても解くことができる設問である。設問3は仮想計算機を用いた場合のインタプリタ方式と動的コンパイル方式の違いを，実行時間で考えさせる内容である。多少難易度が高いと思うが，設問文に記述されている内容を読み取れば，計算内容は把握できる。全体を通じてのポイントは，設問3で記述されている説明文から，実行時間計算のための前提条件を読み取る点にある。なお，計算ミスにも注意する必要がある。

参考までに，インタプリタとコンパイラの特徴について確認しておく。

1. インタプリタ

原始プログラムの命令を，一つずつ翻訳しながら実行する。プログラムを修正して実行するまでのターンアラウンドタイムが短く，また，途中までしかできていないプログラムを試しに実行してみるということもできるので，プログラムのデバッグ（修正）に便利である。一方，命令一つ一つに，それを解釈する手間がかかるため，完成後のプログラムで，データ処理を実行させるときの効率は低くなる。

2. コンパイラ

原始プログラムを，機械語の目的プログラム（オブジェクトモジュール）にコンパイル（翻訳）し，一括変換する。目的プログラムはリンカ（連係編集プログラム）によって，実行形式のモジュール（ロードモジュール）に変換され実行される。コンパイルの処理過程でプログラムのサイズを小さくしたり，実行時間を短縮したりなどの最適化が図れる。完成されたプログラムでデータ処理を実行させるときの効率はインタプリタに比べ高い。

なお，動的コンパイルとは，プログラム言語の実装で用いられ，プログラムの性能を実行時に向上させるための手法，技術を言う。これを用いた言語としてJavaがある。

［設問1］

コンパイラ方式の利点は，「2. コンパイラ」の解説で示したように，最適化が図れることである。よって（エ）が正解である。（ア）～（ウ）はいずれもインタプリタに関する説明である。

［設問2］

仮想計算機Vを用いた内容である。仮想機械（VM：Virtual Machine）のことで，ソフトウェアによって仮想的に構築されたコンピュータのことを言う。ソフトウェアやハードウェアを動作させるために必要な基盤となるハードウェアやOS，ミドルウェアなどをプラットフォームという。ソフトウェアの作成における仮想機械は，プログラミング言語の実行コードを，実際の物理的なプラットフォームに適したコードに変換して実行してくれる。そのため，ソフトウェア開発者（プログラマ）は，プラットフォームごとの違いを意識せずにソフトウェアを開発できるようになる。

Javaで開発されたプログラムは，特定の機種やOSに依存しない独自形式であるJavaバイトコードというコード体系のプログラムに変換されて配布される。一方，Javaプログラムを実行するためのソフトウェアをJavaVMというが，Javaバイトコードという中間コードで記述されたコンピュータプログラムを翻訳し，そのプラットフォームで実行可能な機械語のコードに変換して実行する。この仕組みによって，Java言語によるプログラムは開発時に各機種・OSに対応したプログラムを用意する必要はなく，一つの形式でJavaVMの動作する各環境に対応することができる。

・空欄a：中間コードの特徴に関することであるが，設問文に，「異なるOSやハードウェア上で動作する仮想計算機Vを用意する」，また，〔図1の説明〕の（1）で「言語Xで記述された原始プログラムを，仮想計算機Vで実行できる中間コードに変換する」と記述されているので，（ア）の「特定のOSやハードウェアに依存しない」が正解である。

なお，（イ）は逆の内容であり誤りである。一方（ウ），（エ）については問題文にそうした特徴について言及していないので，適切ではないが，実際にJavaVMも，特定のハードウェアの性能を引き出したり，メモリ使用量を低減したりする機能はもっていない。

［設問3］

インタプリタ方式と動的コンパイル方式の実行時間を求める内容である。図2にはインタプリタ（図2①）も含まれているが，インタプリタと動的コンパイラを使い分けるという，動的コンパイル方式なので，単純にインタプリタを使った場合の処理時間，動的コンパイラを使った場合の処理時間を求めるのではない。ある回数まではインタプリタを使った処理時間，そこから先は動的コンパイラで1度実行形式プログラムに変換された後の処理時間を用いて計算する必要がある点に注意が必要である。また，求める時間の単位は秒であるが，ナノ秒などへの変換など，時間の単位は理解しているものとして説明する。

・空欄b，c：実行時間を計算するための条件は次のとおりである。

〔条件〕

① 主プログラムの実行時間は考えない。

② プロファイル情報を収集する時間と，仮想計算機Vから実行形式プログラムを呼び出すのに必要な処理時間は考えない。

③ 関数Fの中間コードは400命令から成り，関数Fが1回呼び出されたときに実行する中間コードの命令数は，2,000命令である。

④ 動的コンパイル方式を適用した場合，関数Fが101回目に呼び出されるときに動的コンパイラが起動され，関数Fの中間コードを実行形式プログラムに変換する。

⑤ 動的コンパイラの起動時間とコンパイル時間は，実行時間に含める。

⑥ 動的コンパイラの起動時間は0.1秒とし，コンパイル時間は，中間コード1,000命令当たりで0.1秒とする。

⑦ インタプリタによる中間コード1命令の実行時間は500ナノ秒とし，中間コード1命令に対する実行形式プログラムの実行時間は10ナノ秒とする。

(1) インタプリタ方式

主プログラムから関数Fの呼出し回数が400回のときの実行時間である。条件③に，「関数Fが1回呼び出されたときに実行する中間コードの命令数は，2,000命令」となっている。また，条件⑦より，「インタプリタによる中間コード1命令の実行時間は500ナノ秒」なので，求める実行時間をt_1とすると，次のようになる。

t_1＝2,000(命令)×500(ナノ秒)×400(回)＝400×10^6(ナノ秒)＝0.4(秒)

したがって，空欄bの正解は（オ）である。

(2) 動的コンパイル方式

図2と〔図2の説明〕及び〔条件〕をよく把握した上で計算する。〔図2の説明〕(4)，(5)より，動的コンパイラを起動して関数Fを実行形式プログラムに変換した後は，実行形式プログラムの実行時間だけとなるので，計算は，関数Fの呼出し回数が100回までと101回以降に分けて考えればよい。

(A) 関数Fの呼出し回数が100回まで

100回まではインタプリタ方式となるので，実行時間をt_Aとすると，次のようになる。(条件③，⑦参照)

t_A＝2,000(命令)×500(ナノ秒)×100(回)＝100×10^6(ナノ秒)
＝0.1(秒)

(B) 関数Fの呼出し回数が101～400回まで(呼出し回数は300回，条件③，⑥，⑦参照)

101回からは動的コンパイル方式となり，求める実行時間は，動的コンパイラの起動時間とコンパイル時間及び実行形式プログラムの実行時間の合計となる。

実行時間をt_Bとすると，次のようになる。

t_B＝起動時間＋コンパイル時間＋実行形式プログラムの実行時間

＝0.1(秒)＋{400(命令)／1,000(命令)}×0.1(秒)　← 条件③と⑥
　　＋2,000(命令)×10(ナノ秒)×300(回)
＝0.1(秒)＋0.04(秒)＋6×10^6(ナノ秒)＝0.146(秒)

条件⑥

条件③と⑦

したがって，全体の実行時間をt_2すると次のようになる。

t_2＝t_A＋t_B＝0.1(秒)＋0.146(秒)＝0.246(秒)

よって，空欄cの正解は（エ）である。