問1　Javaプログラミングに関する次の記述を読んで，設問1～3に答えよ。

S社は，従業者数500名（従業員30名，パートタイマ（以下，パートという）470名），営業所が1か所の警備業者であり，各種イベントの警備や交通誘導警備を請け負っている。これまで，パートの勤務時間の集計と給与計算は，手書き帳票を基にして経理課にて行っていた。今回，情報システム課でWebアプリケーションによる勤務時間管理システムを開発して，集計などの作業をシステム化することとした。このシステム化によって，毎月初めの5営業日以内にパートが営業所に赴き，PC操作によって自分の前月分の勤務時間集計表を印刷した後，その表に記載された勤務時間と給与支給金額に誤りがないことを確認して押印し提出するという作業の流れとなった。

なお，パートの勤務時間は，業務監督者が記録し，専任の担当者がデータベースに入力する。

〔システム構成〕

S社の勤務時間管理システムのシステム構成は，図1のとおりである。

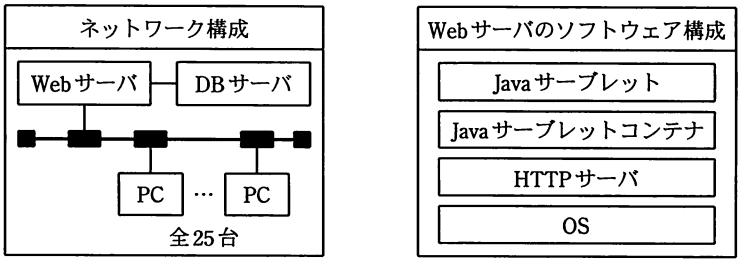

図1　勤務時間管理システムのシステム構成

HTTPサーバ及びJavaサーブレットコンテナはオープンソースソフトウェアを使用しており，Javaサーブレットコンテナは Java Platform, Enterprise Edition 5（Java EE 5）に準拠したものである。また，利用者用PCは，25台を用意してパート全員で共用する。

システムの利用者IDには，6けたの数字からなる従業者番号を使用する。

〔システム仕様〕

S 社の情報システム課には従業員が 4 名在籍しているが，これまで T 君がほぼ 1 人でシステム開発，管理を行っており，今回の開発も T 君が 1 人で担当することとなった。T 君は，これまで Java サーブレットモデルを用いたシステム開発を経験したことがなかった。しかし，プログラム上でメモリを使用する際の境界チェックをしなくても Java バーチャルマシンによってメモリの使用状況が管理されるために，C 言語又は C++言語で開発したプログラムにおいてよく発生する［ a ］によるセキュリティ上の脆弱(ぜいじゃく)性が発生しないという利点を重視して，Java サーブレットモデルを用いた開発を決定した。

今回開発する勤務時間管理システムの仕様は，図 2 のとおりである。

1. 勤務時間管理システムは，利用者 ID 及びパスワードを使用して本人確認を行う。
2. 利用者用 PC 上のブラウザ操作によって，Web サーバ内の Java サーブレットは勤務時間集計表を PDF ファイルとして動的に作成し，HTTP サーバの Web 公開領域に保存する。
   なお，勤務時間集計表の作成においては既存の勤務時間集計表作成プログラムを呼び出す。
3. Web サーバ内の Java サーブレットは，勤務時間集計表へアクセスするためのリンク情報を含んだ HTML 文書をブラウザへ送信する。利用者は，ブラウザ上に表示されたリンクをクリックして，勤務時間集計表をダウンロードする。
4. 定期的に起動されるプロセスによって，作成から 1 時間以上経過した勤務時間集計表を自動的に削除する。

図 2　勤務時間管理システムの仕様（抜粋）

〔勤務時間集計表の作成及びリンク表示のプログラム〕

T 君は，図 2 の仕様に基づいて Java プログラムを作成した。その Java プログラムのうち，勤務時間集計表を作成して，その表へのリンクを表示する部分について図 3 に示す。プログラム完成後，仕様に基づいた機能試験を試験環境にて行い，問題がないことを確認した。

```
package jp.co.s_sha.kinmuhyo;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class PDFDownloader extends HttpServlet {
   private static final long serialVersionUID = 1L;

   static final String KINMUHYO_LOCAL_PATH = "c:\\inetpub\\wwwroot\\KinmuHyo";
   static final String KINMUHYO_URL_PATH = "../KinmuHyo/";
   File tempPDF;

   public void init() {
      (省略)   // データベースへ接続
   }

   public void destroy() {
      (省略)   // データベースから切断
   }

   protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
      String tempUserID = request.getRemoteUser();   // tempUserID に利用者 ID を代入
      tempPDF = new File(KINMUHYO_LOCAL_PATH, tempUserID + ".pdf");
      makePDF(tempUserID, tempPDF);
      response.setContentType("text/html; charset=utf-8");
      PrintWriter out = response.getWriter();
      (省略)   // 定型的な HTML 出力
      out.println("<a href=\"" + KINMUHYO_URL_PATH + tempPDF.getName() +
                  "\">こちら</a>からダウンロードしてください。");
      (省略)   // 定型的な HTML 出力
   }

   protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
      (省略)   // doGet メソッドの呼出し
   }

   protected void makePDF(String userID, File output)
            throws IOException {
      (省略)   /* 勤務時間集計表作成プログラムを呼び出し，利用者 ID (userID) で示され
                  るパートの勤務時間集計表を output で示されるファイルとして作成 */
   }
}
```

図３　勤務時間管理システムのプログラム（抜粋）

〔トラブルの発生と対策〕

その後，S 社では勤務時間管理システムの運用を開始した。しかし，パートから，①マニュアルどおりに操作したにもかかわらず，ダウンロードした勤務時間集計表が他人のものだったという苦情があった。この問題が発生したときには 25 名の利用者が同時に勤務時間管理システムへアクセスしていた。T 君は，この問題の原因を調査するためにプログラムを見直していたが，なかなか原因を特定できなかった。さらに，最初にこの問題が発生してから 3 日後に，23 名の利用者が同時に勤務時間管理システムへアクセスしていた際にも，同じ問題が発生した。

この状況を見て，情報システム課の H 課長は勤務時間管理システムの運用停止を指示するとともに，T 君単独でのシステム開発には無理があると考え，セキュリティにも詳しいシステムコンサルタントの U 氏に支援を要請した。

支援を開始した U 氏は，最初に図 3 のプログラムを検査して，問題の原因と修正方法を指摘した。その指摘に従い，T 君はプログラムを修正し，テストを行って問題が発生しないことを確認した。

さらに U 氏は，悪意をもったパートが②他人のパスワードを使用しなくても，容易に他人の勤務時間集計表をダウンロードできる脆弱性があることを指摘した。

指摘を受けた T 君は，作成する勤務時間集計表のファイル名をランダム文字列とすれば脆弱性は解消できるのではないかと提案した。しかし U 氏は，そもそも個人情報が記録されたファイルを Web 公開領域に保存することに問題があると指摘して，脆弱性を解消するために図 2 の仕様を図 4 のように修正することを提案した。

1. 勤務時間管理システムは，利用者 ID 及びパスワードを使用して本人確認を行う。
2. 利用者用 PC 上のブラウザ操作によって，Web サーバ内の Java サーブレットは勤務時間集計表を PDF ファイルとして動的に作成し，Web 公開領域ではなく Java サーブレット経由でだけアクセス可能な領域に保存する。
   なお，勤務時間集計表の作成においては既存の勤務時間集計表作成プログラムを呼び出す。
3. Web サーバ内の Java サーブレットは，［ b ］をブラウザへ送信する。利用者は，ブラウザの PDF プラグイン機能によって表示された内容を印刷する。
4. 定期的に起動されるプロセスによって，作成から 1 時間以上経過した勤務時間集計表を自動的に削除する。

図 4　勤務時間管理システムの修正仕様（抜粋）

その後，T 君は U 氏の支援を受けつつ，図 4 の仕様に沿って勤務時間管理システムの修正を行い，脆弱性が解消されたことを確認した。この結果を受けて，H 課長は勤務時間管理システムの運用再開を指示した。

設問 1　本文中の［　a　］に入れる適切な字句を 15 字以内で答えよ。

設問 2　本文中の下線①の苦情について，(1)～(4)に答えよ。

(1) マニュアルどおりに操作したにもかかわらず，他人の勤務時間集計表をダウンロードしてしまった原因について，図 3 のプログラム上の変数名を用いて，60 字以内で述べよ。

(2) この苦情の原因となるプログラムのバグで発生する問題は何と呼ばれるか。15 字以内で答えよ。

(3) 上記 (2) のバグを修正するためのプログラムの変更内容を，45 字以内で具体的に述べよ。ただし，図 2 に示した仕様は変更しないものとする。

(4) このプログラムに関して，上記 (2) のバグの有無を確認するためにテストを実施したい。どのような HTTP リクエストをどのように Web サーバに送信すればよいか。50 字以内で述べよ。

設問 3　本文中の下線②の脆弱性について，(1)，(2)に答えよ。

(1) この脆弱性を突いて，他人の勤務時間集計表をダウンロードする方法を 50 字以内で述べよ。

(2) この脆弱性を解消する修正仕様に関して，図 4 中の［　b　］に入れる適切な字句を 20 字以内で答えよ。