

[1] 以下の問いに答えなさい。

- (1) インターネットプロトコル・TCP・UDPに関する以下の記述からそれぞれ正しいものをすべて選べ。
 (ア) ネットワーク層のプロトコルである。
 (イ) アプリケーション層のプロトコルである。
 (ウ) ネットワークアクセス層のプロトコルとして、イーサネットを用いなければならない。
 (エ) ヘッダにシーケンス番号が含まれている。
 (オ) 可変長のデータを送信できる。
 (カ) 53バイトの固定長のデータを単位とする通信方式である。
 (キ) 受信した内容に誤りがないことを確認するためのフロー制御を行う。
 (ク) スライディングウィンドウによるフロー制御を行う。
 (ケ) ヘッダにあて先ポート番号が含まれており、通信相手のアプリケーションを特定する。
 (コ) ヘッダにIPアドレスとポート番号を含む。
 (サ) 3ウェイハンドシェイクにより、接続を確立する。
 (シ) 通信相手にパケットが届くことを保証する。
 (ス) 輻輳制御は行わないが、順序保証を行う。
 (セ) IPアドレスを参照することで、パケットの経路を決定し、通信相手のコンピュータにパケットを届ける。
 (ソ) 各パケットに一意的番号が振られており、下位プロトコルでパケットを分割されても復元できる。
- (2) 以下の記述から間違っているものを選べ。
 (ア) 途中でデータが抜け落ちて問題がないアプリケーションなので、オーバーヘッドの少ないUDPを選択した。
 (イ) 通信相手の準備が整ってから処理を進めるアプリケーションなので、ユーザの入力を受け付けるたびに、メッセージを送信する設計とした。
 (ウ) バイナリデータを送受信する場合、通信相手とバイトオーダーがそろっていないとは限らないので、送信側/受信側ともにネットワークバイトオーダーへからの変換を行うのが望ましい。
 (エ) 送信ごとにサイズの異なるデータをやりとりしたいが、バイナリデータでは可変長データを送信できないためテキストデータを用いた。
 (オ) テキストデータは人間にも解読しやすいので、デバッグしやすい。

- (3) HTTPに関する以下の記述から間違っているものを選べ。
 (ア) Hyper Text Transfer Protocol の略で、ハイパーテキストの転送が主ではあるが、画像・音声などのバイナリデータの転送にも用いられる。
 (イ) サーバの応答は、3桁の数字で表されるステータスコードで始まる。
 (ウ) TCPによる接続後、はじめにクライアントからのリクエストが行われる。
 (エ) クライアントからのリクエストは、1行目が要求するファイルなどを示す行で、2行目から空行までがヘッダである。
 (オ) リクエストのメソッドにはGETやPOSTがある。
- (4) ウェブプログラミングに関する以下の記述から正しいものを選べ。
 (ア) CGIとは、ウェブブラウザにてプログラムを読み込み実行する仕組みのことである。
 (イ) ウェブサーバ上でプログラムを実行し、都度HTTPデータを生成することで、動的なウェブページを実現することができる。
 (ウ) JavaScriptは、ウェブブラウザで実行されるサーバサイド技術である。
 (エ) サーバサイドの処理負荷を軽減するために、あらかじめウェブブラウザの一部としてプログラムを起動しておくFastCGIという技術がある。
 (オ) jQueryはJavaScriptのライブラリで、ウェブサーバで動作する。

- [2] ネットワーク通信によりババ抜き(トランプゲーム)の対戦を行うプログラムを作成する。あなただったらどんなプロトコルを設計しますか?ただし、以下の仕様を満たすものとする。仕様に記述されていない点については、自由に方針を決めてよい。特に、データの形式と通信手順については、自由記述で詳細に示しなさい。画面にどのように表示するかは記述しなくてよい。解答欄が足りない場合、解答用紙の裏面も使用してよい。
 --<仕様>--
 ・参加人数・カードを引く順番は、毎回固定でよい。
 ・手札がなくなった段階で勝ち抜けとし、ある参加者の順位を少なくともその参加者自身を知ることができるようにする。
 ・1つのゲームが終了後、参加者全員が望む場合は、もう1度最初からゲームを再開する。

[3] Java のプログラムに関する以下の問いに答えなさい。

- (1) TCPでソケット通信を行うサーバとクライアントのプログラムの流れを以下に示す。このとき、(a), (b), (c), (d)に入るクラス名・メソッド名をそれぞれ選択肢の中から選べ。ただし、try文の例外処理は省略してある。
 --<サーバ>--

```

    [ (a) ] server = new [ (a) ](port);
    Socket client = server.[ (b) ];
    InputStream input = client.getInputStream();
    OutputStream output = client.getOutputStream();

    //データの送受信

    //接続終了
    client.[ (c) ];
    server.[ (c) ];
    ----
    --<クライアント>--
    [ (d) ] server = new [ (d) ](hostname, port);
    InputStream input = server.getInputStream();
    OutputStream output = server.getOutputStream();

    //データの送受信

    //接続終了
    server.[ (c) ]();
    ----
    
```

 空欄 (a), (b), (c), (d) 共通
 (ア) listen()
 (イ) Socket
 (ウ) accept()
 (エ) read()
 (オ) InetAddress
 (カ) bind()

- (キ) socket()
 (ク) InetAddress
 (ケ) recv()
 (コ) fork()
 (サ) DatagramPacket
 (シ) DatagramSocket
 (ス) close()
 (セ) ServerSocket
 (ソ) connect()
- (2) マウスをクリックした位置と1つ前にクリックした位置を結ぶ赤い直線を追加するプログラムを以下に示す。このとき、(a), (b), (c), (d), (e)に入るプログラムコードを選択肢の中から選べ。

```

    int mouseX;
    int mouseY;
    int previousMouseX;
    int previousMouseY;

    this.panel = new JPanel() {
        @Override
        protected void paintComponent(Graphics g) {
            g.setColor(Color.red);
            g.[ (a) ];
        }
    };

    MouseAdapter mouseAdapter = new MouseAdapter() {
        @Override
        public void mousePressed(MouseEvent me) {
            previousMouseX = [ (b) ];
            previousMouseY = [ (c) ];
            mouseX = [ (d) ];
            mouseY = [ (e) ];

            panel.repaint();
        }
    };
    
```

```
this.panel.addMouseListener(mouseAdapter);
```

空欄(a)

- (ア) drawLine(mouseX, mouseY, previousMouseX, previousMouseY)
- (イ) drawLine(previousMouseY, previousMouseX, mouseY, mouseX)
- (ウ) drawString(previousMouseY, previousMouseX, mouseY, mouseX)
- (エ) drawString("" + mouseX, "" + mouseY, previousMouseX, previousMouseY)
- (オ) drawString(new Point(mouseX, mouseY), previousMouseX, previousMouseY)

空欄 (b), (c), (d), (e) 共通

- (ア) me.getX()
- (イ) me.getY()
- (ウ) getWidth()
- (エ) getHeight()
- (オ) mouseX
- (カ) mouseY
- (キ) previousMouseX
- (ク) previousMouseY
- (ケ) me.pageX()
- (コ) me.pageY()
- (サ) clickX
- (シ) clickY

- (3) TCP で以下の形式のテキストデータを送信するとする。このとき、以下のプログラム中の (a), (b) に入るプログラムコードを選択肢の中から選べ。ただし、変数には送信すべきデータは格納されているものとする。また、try 文の例外処理は省略してある。

--<データ形式>--

```
<time>_<id>_<x>_<y>
```

--<プログラム>--

```
OutputStream output = socket.getOutputStream();  
[ (a) ] dataOut = new [ (a) ](output, true);  
dataOut.[ (b) ];
```

空欄(a)

- (ア) PrintWriter
- (イ) PrintReader
- (ウ) OutputStreamReader
- (エ) DataOutputStream
- (オ) BufferedReader

空欄(b)

- (ア) printf(time + ' ' + id + ' ' + x + ' ' + y)
- (イ) println(time + " " + id + " " + x + " " + y)
- (ウ) println(time + ' ' + id + ' ' + x + ' ' + y + '\n')
- (エ) print(time + " " + id + " " + x + " " + y)
- (オ) format("%s %d %f %f", time, id, x, y)

[4] JavaScript のプログラムに関する以下の問いに答えなさい。

- (1) 次の文を実行すると表示される文字を選択肢の中から選べ。ただし、document.write() は引数をウェブページに書き出す関数である。

```
for (var i = 0; i < 5; i += 2) {  
  document.write(i + ",");  
}
```

- (ア) 0+2+4
- (イ) 0,2,4,
- (ウ) 6
- (エ) iii

- (2) 変数 cakes が次のように定義されているとき、モンブランの価格(price)の値を取り出す変数を選択肢の中から選べ。

```
var cakes = [  
  { menu:"ティラミス", price:500, cal:410 },  
  { menu:"ショートケーキ", price:380, cal:320 },  
  { menu:"モンブラン", price:450, cal:388 },  
  { menu:"プリン", price:100, cal:210 }  
];
```

- (ア) cakes[2,1]
- (イ) cakes[2][1]
- (ウ) cakes[2]['price']
- (エ) cakes['モンブラン']['price']

- (3) 以下のプログラムを実行すると、何が alert されるかを選べ。

```
var student = new Array(3);  
student[0] = {  
  名前:"櫻井",  
  趣味:"なし",  
  特技:"ラップ"  
}  
student[1] = {  
  名前:"松本",  
  趣味:"買い物",  
  特技:"演技"  
}  
student[2] = {  
  名前:"大野",  
  趣味:"釣り",  
  特技:"ダンス"  
}  
  
x = "";  
for (var key in student) {  
  x = x + student['名前'] + ", ";  
}  
alert(x);
```

- (ア) 0, 1, 2,
- (イ) 櫻井松本大野
- (ウ) 大野松本櫻井
- (エ) 櫻井, 松本, 大野,
- (オ) undefined, undefined, undefined,

- (4) 以下の記述から論理値が true になるものを選べ。

- (ア) 1=="1"
- (イ) 1=="0"
- (ウ) 1==="1"
- (エ) 1!==1
- (オ) !(1==1)

- (5) 次の文を実行すると起きることを選べ。

```
$('#img').attr('src', 'plane.png')  
          .width('150px').fadeOut(5000);
```

- (ア) タグ の画像を plane.png に変更し、5000ミリ秒かけて幅を 150px に変更する。
- (イ) id が img の画像を plane.png に変更し、幅を 150px に変更し、5000ミリ秒かけて消す。
- (ウ) id が img の画像が plane.png で幅が 150px である場合に、5000ミリ秒かけて消す。
- (エ) タグ の画像を plane.png に変更し、幅を 150px に変更し、5000ミリ秒かけて消す。
- (オ) タグ の画像を plane.png に変更し、幅を 150px に変更し、5000秒かけて消す。

- (6) 偶数番目のリスト要素に限り、クリックしたらそのリスト要素だけ文字の色を変えるようにしたい。このとき、(a), (b) に入るプログラムコードを選択肢の中から選べ。

```
[ (a) ].click(function() {  
  [ (b) ].css('color', 'red');  
});
```

空欄(a)

- (ア) \$("li")
- (イ) \$("li:odd")
- (ウ) \$("li:even")
- (エ) \$("li > odd")
- (オ) \$("li > even")

空欄(b)

- (ア) \$("li:odd")
- (イ) \$("li:even")
- (ウ) \$("li:this")
- (エ) \$("this")
- (オ) \$(this)