情報処理概論期末試験

問１． Java 言語とアルゴリズムに関する次の文章の空欄 (a) ～ (r) を埋めよ。

- クラス Prob11 のオブジェクト参照型変数 p11 を宣言し，p11 を Prob11 のディフォルトのコンストラクタを使って作成したオブジェクトの参照情報で初期化するときは，(a) と記述する。
- 一つのクラス内のメソッドを区別は (b) によって行われ，(b) は (c) と引数の (d) の並びからなる。一つのクラスに (b) が同じで，戻り値の型が異なるメソッドを定義しようとすると，(e) 時にエラーが発生する。メソッドのオーバーロードとは，(f) のことである。例えば，メソッド (g) とメソッド (h) を定義することである (短くても良いがメソッドの定義になっていること)。
- クラス Class1 を元に，拡張・変更した新しいクラス Class2 を宣言することを，Class1 を (i) して，Class2 を宣言すると言う。クラスの宣言で (i) することを示すためには，キーワード (j) を用いる。Class1 に対して Class2 を子クラスまたは (k) クラスと呼び，Class2 に対して Class1 を親クラスまたは (l) クラスと呼ぶ。
- 親クラスにあるメソッドと同じ (b) を持ったメソッドを，子クラスで定義することをメソッドの (m) と呼ぶ。
- あるクラスのオブジェクトの参照情報は，その親クラスのオブジェクト参照型の変数に代入することができる。このような仕組みを，オブジェクト指向における (n) と呼ばれる。
- データを昇順あるいは降順に並び替えるソートの方法には，(o)，バブルソート，(p)，マージソート，ヒープソートなどがある。データ数 N に対して，計算量が N^2 のオーダになるソートとして (q) が，計算量が $N \log_2 N$ のオーダになるソートとして (r) がある。

問２． 店のオブジェクトの配列から，最も適する店を探して表示するプログラムを設計した。

(1) クラス Shop は，店 1 つのデータを格納するためのクラスである。

- フィールド name に店の名前，フィールド maxNum に用意できる最大数，フィールド price1 に部品 1 の価格，フィールド price2 に部品 2 の価格を格納する。
- Shop(String name, int maxNum, int price1, int price2) はコンストラクタで，引数の name, maxNum, price1, price2 を同じ名前のフィールドに格納する。
- lt(Shop s1, Shop s2) は，s1 と s2 の価格を比べて，s1 の方が安い場合は true を，そうでない場合は false を戻り値とする**クラス**メソッドである。ここで，安いとは，price1 の値が小さいか，price1 値が同じ場合は price2 の値が小さいことである。

(2) クラス ShopArray は，複数の店のデータを配列に格納し，処理するためのクラスである。

- フィールド shops, chosenShops は，Shop のオブジェクト参照型配列 (各要素が Shop のオブジェクトを参照している配列) である。
- Shops(int nShops) はコンストラクタである。クラス Shop のオブジェクト参照情報を格納する配列のための，長さ nShops の領域を 2 つ確保し，それぞれを shops と chooseShops に代入する。
- setData(int sn, String inName, int maxNum, int price1, int price2) は，クラス Shop のオブジェクトを，名前 name，用意できる最大数 maxNum，価格 price1 と price2 でコンストラクトし，その参照情報を shops[sn] に代入する。
- choose(int num) は，必要とする数を確保できない店を除外するためのメソッドである。shops[0] から shops[shops.length - 1] までの店から，用意できる最大数が num 以上のものを探しだし，その参照情報を順番に配列 chosenShops[] に格納する。その戻り値は，chosenShops[] に格納した店の数とする。
- sortSelect(int chosenNum) は，chosenShop[0] から chosenShop[chosenNum - 1] を，選択ソートによって，価格の昇順 (配列で価格は安いものが先にくる順番) に並び替えるメソッドである (価格が同じ場合の順番は問わない)。

(3) Kimatsu2 は，メインメソッドのためのクラスである。

このとき，次の問に答えよ。

(ア) 無駄なく，上の仕様を満たすために，プログラム中の下線部 (a)〜(i) に書き入れるべきものを答えよ。

(イ) このプログラムを実行したとき (コマンド java Kimatsu2 によって動かしたとき)，出力されるものを答えよ。
(確認：System.out.println() を呼んでいる箇所はプログラム中に 2 カ所だけである。)

```
class Shop {
  String name;
  int    maxNum, price1, price2;
  Shop(String name, int maxNum, int price1, int price2) {
    this.name   = name;
    this.maxNum = maxNum;
    this.price1 = price1;
    this.price2 = ____________(a)____________;
  }
  static boolean lt(Shop s1, Shop s2) {
    if (s1.price1 < s2.price1) return true;
    else if (____________(b)____________ && ____________(c)____________) return true;
    else return false;
  }
}

class ShopArray {
  Shop[] shops, chosenShops;
  ShopArray(int nShops) {
    shops       = new Shop[nShops];
    chosenShops = new Shop[nShops];
  }
  void setData(int sn, String inName, int maxNum, int price1, int price2) {
    shops[sn] = new Shop(inName, maxNum, price1, price2);
  }
  int choose(int num) {
    int j = 0;
    for (int i = 0 ; i < shops.length ; ++i) {
      if (shops[i].maxNum >= num)  chosenShops[_____(d)_____] = shops[i];
    }
    return j;
  }
  void sortSelect(int chosenNum) {
    for (int i = 0 ; i < chosenNum - 1 ; ++i) {
      int minj = i;
      for (int j = i + 1 ; j < chosenNum ; ++j) {
        if (Shop.lt(____________(e)____________, ____________(f)____________)) minj = j;
      }
      System.out.println(chosenShops[i].name + "のデータと"
                         + chosenShops[minj].name + "のデータを交換");
      Shop tmp          = ____________(g)____________;
      chosenShops[i]    = ____________(h)____________;
      chosenShops[minj] = ____________(i)____________;
    }
  }
}
```

```
class Kimatsu2 {
  public static void main(String[] args) {
    ShopArray sa = new ShopArray(7);
    sa.setData(0, "亜土電子", 50,  100, 22); sa.setData(1, "東名電子", 200,  110, 22);
    sa.setData(2, "信越電子", 150,  100, 20); sa.setData(3, "秋月電子", 120,  100, 25);
    sa.setData(4, "藤商電子", 20,  105, 22); sa.setData(5, "共立電子", 90,  110, 24);
    sa.setData(6, "若松通商", 500,  105, 20);
    int cN = sa.choose(100);
    sa.sortSelect(cN);
    System.out.println("選んだお店は「" + sa.chosenShops[0].name +"」となった！");
  }
}
```

問３． 次のプログラムは，部品管理ためのプログラムの一部である。

(1) クラス Parts のフィールドは，価格を格納する price だけである。メソッド Parts() は，price を引数として，それをフィールドの price に格納するコンストラクターである。メソッド printData() は，そのフィールド price を表示する。

(2) クラス Resister は，クラス Parts を継承している。オリジナルのフィールドは，抵抗値を格納する resistance だけである。メソッド Resister() は，price と resistance を引数とするコンストラクターである。その中で，スーパークラスのコンストラクタを引数 price で呼び出し，引数の resistance をフィールドの resistance に格納する。メソッド printData() は，そのフィールド price と resistance を表示する。メソッド printResistance() は，そのフィールドの中で resistance だけを表示する。

(3) クラス Capacitor は，クラス Parts を継承している。オリジナルのフィールドは，静電容量を格納する capacitance だけである。メソッド Capacitor() は，price と capacitance を引数とするコンストラクターである。その中で，スーパークラスのコンストラクタを引数 price で呼び出し，引数の capacitance をフィールドの capacitance に格納する。メソッド printData() は，そのフィールド price と capacitance を表示する。(メソッド printCapacitance() を作る必要はない。)

(4) Kimatsu3 は，メインメソッドのためのクラスである。

このとき，次の問に答えよ。

(ア) クラス Resistor 参考にして，クラス Capacitor を書きなさい。

(イ) このプログラムを実行したとき (コマンド java Kimatsu3 によって動かしたとき)，出力されるものを答えよ。

(ウ) クラス Kimatu3 の b.printResistance(); のコメントを外して，コンパイルあるいは実行しようとするとどのようなことが起きるか，理由と共に記しなさい。

(エ) さらに，変数 b に参照情報が格納されているオブジェクトの printResistance() を，変数 b を使って呼び出すためには，どのように修正すれば良いか記しなさい (この行だけの修正でも良いし，場所を明示して他のクラスを変更しても良い)。

```
class Parts {
  int  price;
  Parts(int price) {
    this.price   = price;
  }
  void printData() {
    System.out.println("price = " + price);
  }
}

class Resister extends Parts {
  double resistance;
  Resister(int price, double resistance) {
    super(price);
    this.resistance = resistance;
  }
  void printData() {
    System.out.println("price = " + price + " resistance = " + resistance);
  }
  void printResistance() {
    System.out.println("resistance = " + resistance);
  }
}
```

```
class Kimatsu3 {
  public static void main(String[] args) {
    Parts a     = new Parts(100);
    Parts b;
    Resister r = new Resister(10, 50.0);
    Capacitor c = new Capacitor(10, 0.001);
    a.printData();
    r.printData();
    c.printData();
    b = r;
    b.printData();
    // b.printResistance();
  }
}
```

解答用紙

学科・類：　　学籍番号：　　名前：

問1.

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(j)

(k)

(l)

(m)

(n)

(o)

(p)

(q)

(r)

問2.

(ア)

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(イ)

問3.

(ア)

(イ)

(ウ)

(エ)

解答例

問1.

(a) `Prob11 p11 = new Prob11();`

(b) シグネチャー

(c) メソッド名

(d) 型

(e) コンパイル

(f) メソッド名が同じで，引数の型の並びが異なるメソッドを定義すること。

(g) `void method1() {}`

(h) `void method1(int i) {}`

(i) 継承 (inheritance)

(j) `extends`

(k) サブ

(l) スーパー

(m) オーバーライド

(n) ポリモーフィズム (多態性)

(o), (p) 選択ソート，クイックソート，バカソートなどから 2 つ

(q) 選択ソート，バブルソートなどから 1 つ

(r) クイックソート，マージソート，ヒープソートなどから 1 つ

問2.
(ア)

(a) `price2`

(b) `s1.price1 == s2.price1`

(c) `s1.price2 < s2.price2`

(d) `j++`

(e) `chosenShops[j]`

(f) `chosenShops[minj]`

(g) `chosenShops[i]`

(h) `chosenShops[minj]`

(i) `tmp`

(イ)

東名電子のデータと信越電子のデータを交換
東名電子のデータと秋月電子のデータを交換
東名電子のデータと若松通商のデータを交換
選んだお店は「信越電子」となった！

問3.

(ア)

```
class Capacitor extends Parts {
  double capacitance;
  Capacitor(int price, double capacitance) {
    super(price);
    this.capacitance = capacitance;
  }
  void printData() {
    System.out.println("price = " + price + " capacitance = " + capacitance);
  }
}
```

(イ)

```
price = 100
price = 10 resistance = 50.0
price = 10 capacitance = 0.0010
price = 10 resistance = 50.0
```

(ウ)

コンパイル時に，メソッド printResistance() が見つからないというエラーが発生する。
変数 b は Parts のオブジェクト参照型変数である。そのため，コンパイル時には，
クラス Parts の中からだけからメソッドを探すため，メソッドが見つからないことになる。

(エ)

```
((Resister) b).printResistance();
```

と，変数 b を Resiter にキャストする。