

選択した問題は、選択欄の(選)をマークしてください。マークがない場合は、採点されません。

問 11 次の Java プログラムの説明及びプログラムを読んで、設問 1, 2 に答えよ。

[プログラムの説明]

電気料金プラン（以下、プランという）を比較するためのプログラムである。電気料金は、基本料金、及び電気の使用量（以下、電力量という）から算出される電力量料金から成る。電力量の単位は、キロワット時（kWh）である。電力量料金を算出するための 1kWh 当たりの料金（以下、料金単価という）は、電力量に応じて段階的に変わる。プラン A 及びプラン B の料金表を表 1 に示す。

表 1 プラン A 及びプラン B の料金表

プラン		プラン A	プラン B
基本料金 (円)		1123.30	1040.10
料金単価 (円/kWh)	120 kWh まで	19.62	18.17
	120 kWh 超過 300 kWh まで	26.10	24.17
	300 kWh 超過分	30.12	27.77

例えば、プラン A で、使用した電力量が 200.5 kWh のとき、120 kWh 分に対して 19.62 円/kWh、残りの 80.5kWh に対して 26.10 円/kWh の料金単価が適用される。

(1) 抽象クラス TierTable は、段階的に変化する値のペアをテーブルとして表現する。値は、型 double で与えられる。

① コンストラクタは、可変長の引数で与えられた double の数値 2 個ずつをペアとして配列にし、更にその配列を要素とする配列を生成し、フィールド pairs に保持する。引数 tiers の長さが奇数のときは、IllegalArgumentException を投げる。

② 抽象メソッド map は、引数で与えられた数値を別の数値に変換して返す。

(2) クラス TieredRateTable は、料金単価のテーブルを表す。

① コンストラクタは、引数で与えられた電力量とそれに対応する料金単価のペア

からテーブルを作成する。

- ② メソッド `map` は、引数で与えられた電力量から電力量料金を計算し、その値を返す。
- (3) クラス `RatePlan` は、プランを表す。
  - ① コンストラクタは、引数で与えられたプラン名、基本料金、料金単価のテーブルで表されるインスタンスを生成する。
  - ② メソッド `getName` は、プラン名を返す。
  - ③ メソッド `getPrice` は、引数で与えられた電力量から電力量料金を計算し、その値と基本料金の合計を型 `int` の数値で返す。このとき、小数点以下は切り捨てられる。
- (4) クラス `Main` は、表 1 を基にプラン A 及びプラン B を表す `RatePlan` のインスタンスを生成し、電力量が 543.0 kWh のときの電気料金を比較する。メソッド `main` を実行すると、図 1 の結果が得られた。

プラン B が 1175 円安い

図 1 メソッド `main` の実行結果

[プログラム 1]

```
a class TierTable {
    final double[][] pairs;

    TierTable(double... tiers) {
        if (tiers.length % 2 == 1) {
            throw new IllegalArgumentException("不正な長さ: " + tiers.length);
        }
        double[][] a = new double[tiers.length / 2][2];
        for (int i = 0; i < tiers.length; i += 2) {
            a[b] = new double[] { tiers[i], tiers[i + 1] };
        }
        this.pairs = a;
    }

    abstract double map(double amount);
}
```

## [プログラム 2]

```

class TieredRateTable extends TierTable {
    TieredRateTable(double... tiers) {
        super(tiers);
    }

    double map(double amount) {
        double charge = 0;
        for (int i = 0; i < pairs.length; i++) {
            if (i + 1 < pairs.length && amount > pairs[i + 1][0]) {
                charge += (pairs c - pairs d) * pairs[i][1];
            } else {
                charge += (amount - pairs[i][0]) * pairs[i][1];
                break;
            }
        }
        return charge;
    }
}

```

## [プログラム 3]

```

class RatePlan {
    private final String name;
    private final double basicCharge;
    private final TierTable pricingTiers;

    RatePlan(String name, double basicCharge, TierTable pricingTiers) {
        this.name = name;
        this.basicCharge = basicCharge;
        this.pricingTiers = pricingTiers;
    }

    String getName() { return name; }

    int getPrice(double amount) {
        return (int) (basicCharge + pricingTiers.map(amount));
    }
}

```

[プログラム 4]

```
public class Main {
    public static void main(String[] args) {
        RatePlan planA = new RatePlan("プランA", 1123.30,
            new TieredRateTable(0, 19.62, 120, 26.10, 300, 30.12));
        RatePlan planB = new RatePlan("プランB", 1040.10,
            new TieredRateTable(0, 18.17, 120, 24.17, 300, 27.77));

        double amount = 543.0;
        int d = planA.getPrice(amount) - planB.getPrice(amount);
        if (d < 0) {
            System.out.printf("%sが%d円安い%n", planA.getName(), -d);
        } else if (d > 0) {
            System.out.printf("%sが%d円安い%n", planB.getName(), d);
        } else {
            System.out.println("両プランで同額");
        }
    }
}
```

設問1 プログラム中の  に入れる正しい答えを、解答群の中から選べ。

aに関する解答群

ア abstract	イ final	ウ private
エ protected	オ public	カ static

bに関する解答群

ア i	イ i - 1	ウ i * 2
エ i / 2	オ i / 2 + 1	カ i + 1

c, dに関する解答群

ア [i + 1][0]	イ [i + 1][1]	ウ [i + 1][i]
エ [i][0]	オ [i][1]	カ [i][i + 1]

設問2 割引プランは、指定した他のサービスを電気と一緒に利用した場合には、プラン A やプラン B という当初のプランで計算した電気料金を、電気料金の額に応じて割り引くプランである。割引プランの割引率の例を、表 2 に示す。

表 2 割引プランの割引率の例

電気料金	割引率
5,000 円未満	1%
5,000 円以上 8,000 円未満	3%
8,000 円以上	5%

プログラム 5 は、割引率を求めるためのクラス DiscountTable である。メソッド map は、引数で与えられた電気料金から割引率を求め、その値を返す。ここで、割引率は小数で与えるものとする。例えば、1%は 0.01 である。

[プログラム 5]

```
class DiscountTable extends TierTable {
    DiscountTable(double... tiers) {
        super(tiers);
    }

    double map(double amount) {
        for (int i = pairs.length - 1; i >= 0; i--) {
            if (amount >= pairs[i][0]) {
                return pairs[i][1];
            }
        }
        throw new IllegalArgumentException("amount = " + amount);
    }
}
```

プログラム 6 は、割引プランを表すためのクラス DiscountPlan である。DiscountPlan は、クラス RatePlan を拡張し、上位クラスである RatePlan のメソッド getPrice で求めた電気料金から割引率を求め、割引を適用した金額を電気料金として計算する。プログラム中の  に入れる正しい答えを、解答群の中から選べ。

[プログラム 6]

```
class DiscountPlan e RatePlan {
    private final TierTable discountTiers;

    DiscountPlan(String name, double basicCharge,
                 TierTable pricingTiers, TierTable discountTiers) {
        super(name, basicCharge, pricingTiers);
        this.discountTiers = discountTiers;
    }

    int getPrice(double amount) {
        int price = f.getPrice(amount);
        return (int) (price * g);
    }
}
```

eに関する解答群

- |           |              |           |
|-----------|--------------|-----------|
| ア extends | イ implements | ウ imports |
| エ public  | オ throws     |           |

fに関する解答群

- |                     |                   |             |
|---------------------|-------------------|-------------|
| ア ((RatePlan) this) | イ RatePlan        | ウ super     |
| エ this              | オ TieredRateTable | カ TierTable |

gに関する解答群

- ア (1.0 - discountTiers.map(amount))
- イ (1.0 - discountTiers.map(basicCharge))
- ウ (1.0 - discountTiers.map(price))
- エ discountTiers.map(amount)
- オ discountTiers.map(basicCharge)
- カ discountTiers.map(price)