



Iniciado em sábado, 11 Jun 2022, 15:32

Estado Finalizada

Concluída em domingo, 12 Jun 2022, 17:19

Tempo empregado 1 dia 1 hora

Questão 1

Correto

Vale 2,00 ponto(s).

Marcar questão

Peso Ideal

Segundo uma tabela médica, o peso ideal de uma pessoa está relacionado com a sua altura e com o sexo da pessoa. Implemente um programa que receba como entradas a altura (A) e o sexo (S); a seguir ele calcula e imprime o peso ideal dessa pessoa, considerando:

- A não pode ser negativo;
- Caso $S = 'm'$ (masculino), $\text{peso_ideal} = (72.7 \times A) - 58$;
- Caso $S = 'f'$ (feminino), $\text{peso_ideal} = (62.1 \times A) - 44.7$;
- Caso o sexo seja inválido, ou seja, S diferente de 'm' e 'f', a mensagem "Entrada inválida!", deverá ser impressa na tela.

Entrada:

Uma entrada do tipo float para a altura (em metros), que deve ser maior ou igual a zero, e uma entrada do tipo char para o sexo, onde o primeiro caractere deve ser 'm' (masculino) ou 'f' (feminino).

Saída:

Uma saída que corresponde ao peso ideal (em kg), que é um float com 1 casa decimal apenas.

Dica:

O método `charAt(0)` pode ser utilizado para considerar o primeiro caractere informado na entrada pelo usuário.

A instrução `<entrada>.nextLine();` pode ser utilizada entre a leitura de um valor do tipo numérico e um valor do tipo `char/String` para remover um caractere/espaco do buffer do teclado.

For example:

Input	Result
1.65 f	57.8 kg
-1.4 m	Entrada inválida!

Answer: (penalty regime: 0, 0, 0, 1, 2, ... %)

```
1 import java.util.*;
2
3 public class PesoIdeal {
4
5     public static void main(String[] args) {
6
7         Scanner input = new Scanner(System.in);
8         Float height = input.nextFloat();
9         input.nextLine();
10        char sex = input.next().charAt(0);
11        double pesoIdeal = 0;
12
13        if(height < 0 || (sex != 'm' && sex != 'f')) {
14            System.out.println("Entrada inválida!");
15        }
16    }
17 }
```

Navegação do questionário



Mostrar uma página por vez

Terminar revisão

	Input	Expected	Got	
✓	1.65 f	57.8 kg	57.8 kg	✓
✓	-1.4 m	Entrada inválida!	Entrada inválida!	✓
✓	1.4 M	Entrada inválida!	Entrada inválida!	✓
✓	1.8 m	72.9 kg	72.9 kg	✓
✓	2.21 m	102.7 kg	102.7 kg	✓
✓	1.65 f	57.8 kg	57.8 kg	✓
✓	1.8 m	72.9 kg	72.9 kg	✓
✓	2.21	102.7 kg	102.7 kg	✓

✓	m			✓
✓	1.8 m	72.9 kg	72.9 kg	✓
✓	1.65 f	57.8 kg	57.8 kg	✓

Passou em todos os teste! ✓

Questão 2

Correto

Vale 2.00 ponto(s).

🚩 Marcar questão

Cálculo do IPTU

Uma empresa contratou você para fazer um programa que calcule os valores do IPTU de imóveis de uma determinada cidade, conforme o tipo do loteamento e a área dos mesmos. Deverão ser considerados apenas dois tipos de loteamento: 1 e 2. Para cada tipo de loteamento, se a área do imóvel for menor que 200 m², efetua-se um cálculo de IPTU; se for maior ou igual a 200 m², efetua-se outro cálculo de IPTU. A tabela abaixo mostra como o cálculo deve ser efetuado para cada caso.

Tipo de loteamento	0 < área < 200 m ²	Área ≥ 200 m ²
1	$\text{IPTU} = \text{ÁREA} \times 1,0$	$\text{IPTU} = \text{ÁREA} \times 1,2$
2	$\text{IPTU} = \text{ÁREA} \times 1,1$	$\text{IPTU} = \text{ÁREA} \times 1,3$

Implemente um programa que leia o tipo de um loteamento e a área do mesmo e apresente o valor do IPTU de um determinado imóvel, calculado conforme a tabela acima. Caso o valor do tipo de loteamento e/ou a área do imóvel forem inválidos, uma mensagem de entrada inválida deve ser impressa na tela.

Entrada:

O programa necessita de duas entradas do tipo inteiro para representar o tipo de loteamento e a área do imóvel em m². O tipo de loteamento deve conter o valor 1 ou 2 e a área do imóvel deve ser maior que zero, caso contrário as entradas são inválidas.

Saída:

A saída é o valor do IPTU calculado de acordo com o tipo de loteamento e área do imóvel, que é um float com duas casas decimais.

For example:

Input	Result
1 150	150.00
0 150	Entrada inválida!

Answer: (penalty regime: 0, 0, 0, 1, 2, ... %)

```

1 import java.util.*;
2
3 public class CalculoDoIPTU {
4
5     public static void main(String[] args) {
6
7         double iptu = 0;
8         Scanner input = new Scanner(System.in);
9         int type = input.nextInt();
10        int area = input.nextInt();
11        boolean valid = true;
12
13        if(type == 1 && area > 0) {
14            if(area < 200) {
15
16            }
17        }
18    }
19 }
```

	Input	Expected	Got	
✓	1 150	150.00	150.00	✓
✓	0 150	Entrada inválida!	Entrada inválida!	✓
✓	1 300	360.00	360.00	✓
✓	2 300	390.00	390.00	✓
✓	2 0	Entrada inválida!	Entrada inválida!	✓
✓	3 1000	Entrada inválida!	Entrada inválida!	✓
✓	1 10000	12000.00	12000.00	✓
✓	2 10000	13000.00	13000.00	✓
✓	2 -500	Entrada inválida!	Entrada inválida!	✓
✓	2 -500	Entrada inválida!	Entrada inválida!	✓

Passou em todos os teste! ✓

Questão 3

Consumo de energia elétrica

Correto
Vale 2.00
ponto(s).
🚩 Marcar
questão

A conta de energia elétrica de consumidores residenciais de uma cidade é calculada do seguinte modo, onde o consumo é dado em kilowatts (kw):

- Se o consumo é de até 500 kw, a tarifa é de R\$ 0,02 por unidade.
- Se o consumo é maior que 500 kw, mas não excede 1000 kw, a tarifa é de R\$ 0,10 para os 500 primeiros kw e de R\$ 0,05 para cada kw excedente a 500.
- Se o consumo é maior que 1000 kw, a tarifa é de R\$ 0,35 para os 1000 primeiros kw e de R\$ 0,10 para cada kw excedente a 1000.
- Em toda conta, é cobrada uma taxa básica de serviço de R\$ 5,00, independentemente da quantidade de energia consumida.

Implemente um programa que leia o consumo de energia elétrica de uma residência e imprima a sua conta de energia.

Entrada:

Uma entrada do tipo float que corresponde ao consumo de energia elétrica (em kilowatts). O programa deve considerar que o valor fornecido para o consumo de energia será um valor inteiro positivo, não sendo necessário fazer nenhuma validação dos dados de entrada.

Saída:

Três saídas do tipo float com duas casas decimais. A primeira corresponde ao custo da energia de acordo com o consumo. A segunda saída é o valor fixo referente a taxa básica de serviço. A terceira saída é o valor total a pagar de acordo com o consumo e a taxa básica de serviço. Todas as saídas devem ser apresentadas em uma única linha separadas por um espaço.

For example:

Input	Result
308	6.16 5.00 11.16
547	52.35 5.00 57.35

Answer: (penalty regime: 0, 0, 0, 1, 2, ... %)

```
1 import java.util.*;
2
3 public class ConsumoDeEnergiaEletrica {
4
5     public static void main(String[] args) {
6
7         double value;
8         double basicFee = 5;
9         double total;
10        Scanner input = new Scanner(System.in);
11        float consumption = input.nextFloat();
12
13        if(consumption <= 500) {
14            value = consumption * 0.02;
15        }
```

	Input	Expected	Got	
✓	308	6.16 5.00 11.16	6.16 5.00 11.16	✓
✓	547	52.35 5.00 57.35	52.35 5.00 57.35	✓
✓	1123	362.30 5.00 367.30	362.30 5.00 367.30	✓
✓	654	57.70 5.00 62.70	57.70 5.00 62.70	✓
✓	15.46	0.31 5.00 5.31	0.31 5.00 5.31	✓
✓	1000	75.00 5.00 80.00	75.00 5.00 80.00	✓
✓	1576	407.60 5.00 412.60	407.60 5.00 412.60	✓
✓	1576	407.60 5.00 412.60	407.60 5.00 412.60	✓
✓	1000	75.00 5.00 80.00	75.00 5.00 80.00	✓
✓	15.46	0.31 5.00 5.31	0.31 5.00 5.31	✓

Passou em todos os testes! ✓

Questão 4

Correto
Vale 2.00
ponto(s).
🚩 Marcar
questão

Controle de acesso de pessoas

No próximo final de semana ocorrerá a festa da Computação da UnB. O ingresso masculino será de R\$ 12,50 e o feminino será de R\$ 7,40. Um calouro ficou encarregado de operar um programa na portaria para controlar o acesso das pessoas à festa. O programa é executado da seguinte forma:

- Quando chega um homem na festa, ele digita 'h'.
- Quando chega uma mulher na festa ele digita 'm'.
- Quando o calouro quiser encerrar a entrada de dados ele digita 'q'.

O calouro não tem noção de quantas pessoas irão à festa. No momento que a entrada de dados for encerrada, o programa calcula quanto foi arrecadado com os ingressos masculinos e com os ingressos femininos. Também é calculado o total arrecadado.

Entrada:

A entrada será um char, que poderá identificar quem chegou à festa, um homem ('h') ou uma mulher ('m') ou encerrar a entrada de dados ('q'). A entrada de dados será solicitada enquanto não for informado o valor 'q'. Não é necessário a validação dos dados de entrada.

Saída:

Cinco saídas apresentadas em duas linhas. Na primeira linha, a quantidade de homens que chegaram à festa é apresentada e na sequência a quantidade de mulheres, separadas por espaço. Na segunda linha, também separados por espaço, na seguinte ordem: o valor arrecadado com os ingressos masculinos, o valor arrecadado com os ingressos femininos e o arrecadamento total, com duas casas decimais.

Dica:

O método charAt(0) pode ser utilizado para considerar o primeiro caractere informado na entrada pelo usuário.

For example:

Input	Result
m	4 6
m	50.00 44.40 94.40
h	
m	
h	
m	
h	
m	
h	
q	
w	0 0
i	0.00 0.00 0.00
p	
q	

Answer: (penalty regime: 0, 0, 0, 1, 2, ... %)

```
1 import java.util.Scanner;
2
3 public class ControleDeAcessoDePessoas {
4
5     public static void main(String[] args) {
6
7         int men = 0;
8         int women = 0;
9         double menTicket = 0;
10        double womenTicket = 0;
11        double totalTicket = 0;
12        char arrived;
13
14        Scanner input = new Scanner(System.in);
15    }
```

	Input	Expected	Got	
✓	m m h m h m m h q	4 6 50.00 44.40 94.40	4 6 50.00 44.40 94.40	✓
✓	w i p q	0 0 0.00 0.00 0.00	0 0 0.00 0.00 0.00	✓
✓	m h h m q	2 2 25.00 14.80 39.80	2 2 25.00 14.80 39.80	✓
✓	m m m m q	0 5 0.00 37.00 37.00	0 5 0.00 37.00 37.00	✓
✓	m h q	1 1 12.50 7.40 19.90	1 1 12.50 7.40 19.90	✓
✓	m m h h h m m m m q	4 8 50.00 59.20 109.20	4 8 50.00 59.20 109.20	✓

Passou em todos os teste! ✓

Uma loja de materiais de construção precisa de um programa para registrar as suas vendas e controlar o seu estoque. Faça um programa que permita o cadastro de N produtos no estoque da loja. Para cada produto, considere o nome e a quantidade do produto no estoque. Depois do cadastro do estoque, o programa deve registrar as vendas até a palavra "fim" ser informada. Para cada venda, o programa deve solicitar o nome do produto e verificar se o produto existe no estoque. Caso o nome do produto não seja encontrado no estoque, o programa deve apresentar o nome do produto seguido pela mensagem "produto não encontrado". Caso contrário, o programa deve solicitar a quantidade de unidades do determinado produto, apresentar as informações da venda (nome do produto e quantidade), e remover do estoque a quantidade vendida do determinado produto. Ao finalizar as vendas, o programa deve apresentar cada produto do estoque (nome do produto e quantidade).

Entrada:

A primeira linha de entrada consiste em um inteiro positivo $N > 0$, que é quantidade de produtos a serem cadastrados no estoque da loja. As próximas N linhas consistem, cada uma, do nome do produto (uma string sem espaço) e a quantidade de produtos no estoque (um valor inteiro positivo).

Em seguida, de acordo com uma quantidade não determinada de operações de venda, deve ser solicitado o nome do produto (uma string sem espaço) até a palavra "fim" ser informada. Para cada nome do produto encontrado no estoque, a quantidade (um valor inteiro positivo) desejada do produto deve ser solicitada em uma nova linha. Considere que esta quantidade sempre será menor ou igual a quantidade do estoque (não é preciso fazer validação).

Saída:

Para cada produto da venda encontrado no estoque, o programa deve apresentar, em uma mesma linha, o nome do produto (uma string) e a quantidade da venda (valor inteiro). Caso o produto não seja encontrado no estoque, o nome do produto deve ser apresentado seguido da mensagem "produto não encontrado". Em seguida, o programa deve apresentar as informações (nome e quantidade) de todos os produtos do estoque após as operações de venda, considerando um produto em cada linha.

Dica:

A instrução "<entrada>.nextLine();" pode ser utilizada entre a leitura de um valor do tipo numérico e um valor do tipo char/String para remover um caractere/espaço do buffer do teclado.

For example:

Input	Result
3 ralo 200 azulejo 1000 tijolo 5000 ralo 1 tijolo 10 fim	ralo 1 tijolo 10 ralo 199 azulejo 1000 tijolo 4990
2 argamassa 50 areia 100 brita areia 100 fim	brita produto não encontrado areia 100 argamassa 50 areia 0

Answer: (penalty regime: 0, 0, 0, 1, 2, ... %)

```
1 import java.util.*;
2
3 public class OperacoesDeVenda {
4
5     public static void main(String[] args) {
6
7         String purchase = "";
8         int quantity;
9         int newValue;
10        Scanner input = new Scanner(System.in);
11        ArrayList<Object> inventory = new ArrayList<>();
12        int n = input.nextInt();
13        input.nextLine();
14
15        for(int i = 0; i < n; i++) {
```

	Input	Expected	Got	
✓	3 ralo 200 azulejo 1000 tijolo 5000 ralo 1 tijolo 10 fim	ralo 1 tijolo 10 ralo 199 azulejo 1000 tijolo 4990	ralo 1 tijolo 10 ralo 199 azulejo 1000 tijolo 4990	✓
✓	2 argamassa 50 areia 100 brita areia 100 fim	brita produto não encontrado areia 100 argamassa 50 areia 0	brita produto não encontrado areia 100 argamassa 50 areia 0	✓
✓	6 ralo 100 azulejo 200 argamassa 50 tijolo 1000 cimento 40 areia 150 argamassa 10 ralo 100 tijolo 500 fim	argamassa 10 ralo 100 tijolo 500 ralo 0 azulejo 200 argamassa 40 tijolo 500 cimento 40 areia 150	argamassa 10 ralo 100 tijolo 500 ralo 0 azulejo 200 argamassa 40 tijolo 500 cimento 40 areia 150	✓

✓	2 tijolo 100 ralo 100 argamassa areia fim	argamassa produto não encontrado areia produto não encontrado tijolo 100 ralo 100	argamassa produto não encontrado areia produto não encontrado tijolo 100 ralo 100	✓
✓	1 azulejo 50 azulejo 10 ralo fim	azulejo 10 ralo produto não encontrado azulejo 40	azulejo 10 ralo produto não encontrado azulejo 40	✓
✓	2 argamassa 70 areia 20 argamassa 10 areia 5 fim	argamassa 10 areia 5 argamassa 60 areia 15	argamassa 10 areia 5 argamassa 60 areia 15	✓
✓	3 tijolo 70 areia 20 ralo 80 tijolo 10 areia 5 ralo 80 fim	tijolo 10 areia 5 ralo 80 tijolo 60 areia 15 ralo 0	tijolo 10 areia 5 ralo 80 tijolo 60 areia 15 ralo 0	✓
✓	1 tijolo 1000 azulejo fim	azulejo produto não encontrado tijolo 1000	azulejo produto não encontrado tijolo 1000	✓
✓	2 ralo 500 tijolo 100 fim	ralo 500 tijolo 100	ralo 500 tijolo 100	✓
✓	3 ralo 1000 argamassa 3000 cimento 5000 fim	ralo 1000 argamassa 3000 cimento 5000	ralo 1000 argamassa 3000 cimento 5000	✓

Passou em todos os teste! ✓

Terminar revisão



ATIVIDADE ANTERIOR

Atividade 2.1 - Lista de exercícios teóricos - Java

PRÓXIMA ATIVIDADE

Atividade 2.3 - Lista extra - C++



Seguir para...



[Obter o aplicativo para dispositivos móveis](#)