|  |  |
| --- | --- |
|  | COM220 – Computação Orientada a Objetos **Prática Aula 14** |

Um construtor precisa contratar o serviço de trabalhadores para a construção de residências. Os trabalhadores podem ser contratados de duas formas: por **diária** ou por **empreito**.

Cada trabalhador contratado por diária combina com o construtor o valor da sua diária. Cabe ao construtor controlar os dias trabalhados pelos diaristas para realizar o pagamento pelos serviços, o que ocorre ao final de cada mês. Dessa forma, um diarista pode trabalhar um único dia do mês, ou vários dias. Os diaristas são penalizados se chegarem atrasados para o serviço, perdendo 10% do valor da diária quando ocorre um atraso. Se atrasar mais de meia hora, o construtor dispensa o trabalho do diarista naquela data e o diarista, obviamente, não será remunerado. Os diaristas podem optar por receber refeição nos dias em que estão trabalhando. Neste caso, são descontados R$10,00 do valor da diária para custear a refeição. Os diaristas também recebem um auxílio creche no valor de R$100,00 por cada dependente menor que 6 anos. Assim, se um diarista que cobra diária no valor de R$100,00 trabalhar dois dias no mês e tiver dois dependentes menores que 6 anos, ele receberá R$400,00 do construtor (se não atrasar e não solicitar refeição). O diarista que não teve nenhuma diária num dado mês não tem direito a receber o auxílio creche naquele mês. Note que a classe abstrata trabalhador tem uma lista chamada *listaDependentes* na qual informações acerca dos dependentes dos trabalhadores são armazenadas.

A classe *Diaria* deve ser usada para controlar o trabalho dos diaristas. Assim, cada dia trabalhado resulta num objeto dessa classe. Note que, além dos 3 atributos que definem a data da diária, essa classe possui dois atributos booleanos: *refeição* e *atraso*. Esses objetos servem para denotar se o funcionário solicitou refeição ou se atrasou no dia trabalhado. Cada diarista tem uma lista chamada *listaDiaria*, na qual são armazenadas as instâncias de *Diaria*.

O trabalhador contratado por empreito define com o construtor o valor do empreito. Por exemplo, para o empreito “Construção de muros”, o trabalhador empreiteiro solicitou o valor de R$4.000,00. Neste caso, a quantidade de dias que o empreiteiro gastará para realizar o trabalho não é relevante. É apenas necessário informar o mês no qual o trabalho será finalizado. No final daquele mês, o empreiteiro recebe o valor referente à execução do trabalho realizado. Note que se houver atraso na execução do trabalho, o empreiteiro terá um desconto de 20% no valor combinado. Assim, para cada trabalho empreitado deve-se criar um objeto da classe *Empreito*, contendo o mês, o ano, a descrição do empreito, e o atributo booleano *atrasoEntrega* que diz se o trabalho foi realizado no prazo. Cada empreiteiro tem uma lista chamada *listaEmpreito* na qual são armazenadas as instâncias de Empreito.

No final do mês, quando é feito o pagamento aos trabalhadores, o sistema deve emitir um recibo no qual aparece o nome do trabalhador e o valor recebido. Para este fim, foi definido o método *imprimeRecibo* na classe *Trabalhador*. Para calcular o valor que deve impresso no recibo, foi definido o método abstrato *calculaPagto*, que também faz parte da classe *Trabalhador*. Uma figura contendo o recibo de 3 trabalhadores é apresentada no verso desta prova. Note que esta figura apresenta o resultado do código de teste fornecido. Seu projeto, ao executar o código de teste, deve prover um resultado idêntico.

Com base nessas informações, foi realizada a modelagem fornecida com esta prova. Esta modelagem explora os relacionamentos de herança e facilita o uso de polimorfismo. Nesta questão, você deve implementar as classes dessa modelagem, **seguindo à risca** o modelo de domínio apresentado. O código de teste deve executar sem erros a fim de validar sua implementação.

**Importante:**

Colocar seu nome e número de matrícula no topo da prova como comentário. Usar seu nro de matrícula para nomear seu arquivo py. Exemplo: “2019099088.py”

Código de teste:

if \_\_name\_\_ == "\_\_main\_\_":

    listaTrab = []

    d1 = Diarista("111222", "Joao Silva", 100)

    d1.insereDependente("Pedro Silva", 4)

    d1.insereDependente("Ana Silva", 2)

    d1.adicionaDiaria(10, 3, 2022, False, False)

    d1.adicionaDiaria(12, 4, 2022, False, True)

    d2 = Diarista("222333", "Jose Cruz", 120)

    d2.insereDependente("Paula Cruz", 3)

    d2.insereDependente("Mario Cruz", 10)

    d2.adicionaDiaria(5, 4, 2022, False, False)

    d2.adicionaDiaria(6, 4, 2022, True, False)

    d2.adicionaDiaria(7, 4, 2022, True, True)

    e1 = Empreiteiro("333444", "Marcio Souza")

    e1.adicionaEmpreito(3, 2022, "Fundações", 6000, False)

    e1.adicionaEmpreito(4, 2022, "Construção muros", 4000, False)

    e1.adicionaEmpreito(4, 2022, "Instalação dos pisos", 7000, True)

    listaTrab.append(d1)

    listaTrab.append(d2)

    listaTrab.append(e1)

    for trab in listaTrab:

        trab.imprimeRecibo(4, 2022)

Resultado impresso pelo código de teste:

Nome: Joao Silva

Valor recebido: 290.0

Nome: Jose Cruz

Valor recebido: 428.0

Nome: Marcio Souza

Valor recebido: 9600.0

Boa Prova!