Desenvolva os exercícios abaixo utilizando conceitos de Lógica de Programação e de Programação Orientada a Objetos.

1. Um matemático está necessitando de várias funções relacionadas a um número inteiro positivo. Suponha a definição de uma classe Inteiro Positivo que apresenta o seguinte atributo: um número X. Implemente os seguintes métodos:
2. um método setValor, que realiza a consistência necessária para garantir que X seja um inteiro positivo;
3. um método para retornar o número X multiplicado por outro objeto de *InteiroPositivo*;
4. um método para calcular o fatorial de X;

Fatorial (X) = X \* (X-1) \* (X-2) \* (X-3) \* … \* 2 \* 1

1. um método para identificar os divisores inteiros de X e a quantidade de divisores. Exemplo: para o número 12, os divisores são 1, 2, 3, 4, 6, 12 e a quantidade de divisores é 6;
2. um método para calcular a série de Fibonacci formada por X elementos;

Fibonacci = 1, 1, 2, 3, 5, 8, 13, …

1. um método para retornar o valor de H, baseado na fórmula:

, também expresso por 

1. um método para retornar o valor de I, baseado na fórmula:
2. um método para retornar o valor de S, baseado na fórmula:
3. um método para retornar o valor de P, baseado na fórmula:

, considerando X termos.

1. A vampire number has an even number of digits and is formed by multiplying a pair of numbers containing half the number of digits of the result. The digits are taken from the original number in any order. Pairs of trailing zeroes are not allowed. Examples include:

1260 = 21 \* 60

1827 = 21 \* 87

2187 = 27 \* 81

Write a program that finds all the 4-digit vampire numbers.

Vampire numbers first appeared in a [1994](http://en.wikipedia.org/wiki/1994) post by [Clifford A. Pickover](http://en.wikipedia.org/wiki/Clifford_A._Pickover) to the [Usenet](http://en.wikipedia.org/wiki/Usenet) group sci.math, and the article he later wrote was published in chapter 30 of his book *Keys to Infinity*.

1. Há um rumor que circula na internet informando que George Lucas (o criador da série Star Wars) utiliza uma fórmula para criar os nomes para os personagens em suas histórias (Jar Jar Binks, ObiWan Kenobi etc). A fórmula, supostamente, é:

Seu primeiro nome na série Star Wars:

1. Pegue as três primeiras letras de seu sobrenome
2. Adicione a ele as duas primeiras letras de seu nome

Seu sobrenome na série Star Wars:

1. Pegue as duas primeiras letras do sobrenome de solteira de sua mãe
2. Adicione a ele as três primeiras letras do nome da cidade onde você nasceu

E agora sua missão: crie uma classe chamada NameGenerator. Esta classe deve ter um método chamado generateStarWarsName que gera um nome completo Star Wars conforme explicado e cuja assinatura é public static String generateStarWarsName(String[] entrada).

O retorno é a String gerada e o parâmetro de entrada é um vetor de String contendo na primeira posição seu nome completo, na segunda posição o nome de solteira de sua mãe e na terceira posição o nome da cidade onde você nasceu. Investigue a classe String para auxiliá-lo.

Exemplo:

entrada[0] = “João Aparecido da Silva” saída gerada → Siljo Doblu

entrada[1] = “Dores”

entrada[2] = “Blumenau”

1. Uma transportadora controla a sua frota de veículos e as manutenções efetuadas neles. De cada veículo sabe-se a placa, o modelo e a data de aquisição. Cada manutenção efetuada em um veículo é identificada por um número e são anotados o custo (R$), descrição do serviço efetuado e tempo parado do veículo (em dias). De acordo com o tempo acumulado de parada, um veículo tem calculado um índice de sucateamento, seguindo a tabela abaixo:

| **Tempo acumulado (dias)** | **Índice** |
| --- | --- |
| 0 a 20 | 0 |
| 21 a 50 | 0,5 |
| 51 a 80 | 1,0 |
| 81 a 120 | 1,5 |
| Acima de 120 | 2,0 |

Implemente um programa Java OO que permita ao usuário cadastrar os veículos e realizar as operações:

1. avisar quando da inconsistência de qualquer dado através de exceções;
2. a quantidade de vezes que um mesmo serviço – descrição informada por parâmetro - foi efetuado em um veículo (retorno *int*);
3. o índice de sucateamento de um veículo (retorno *float*);
4. o custo médio diário de uma manutenção (custo / tempo parado; retorno *float*);
5. o custo médio das manutenções de um veículo (retorno *float*);
6. comparado com outro veículo, qual dos dois tem o maior índice de sucateamento (retorno *int*: positivo caso o objeto receptor tenha índice maior; zero caso sejam iguais; negativo caso o objeto parâmetro tenha índice maior);

Exemplo dos dados:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Placa: XXX-1111  Modelo: Caminhão VW Titan 2110  Data aq: 10/03/2012 | |  |  |  |  | | --- | --- | --- | --- | | Manutenção | Descrição | Custo (R$) | Tempo | | 1 | Troca do pára-brisas | 623,00 | 1 | | 4 | Substituição do cardan | 2134,00 | 3 | | 21 | Troca de filtro de ar do ar-condicionado | 89,00 | 1 | | Saída:  Índice de sucateamento: 0  Custo médio das manutenções: R$ 948,66 |
| Placa: XXX-2222  Modelo: Utilitário Mercedez PRO  Data aq: 15/04/2011 | |  |  |  |  | | --- | --- | --- | --- | | Manutenção | Descrição | Custo (R$) | Tempo | | 3 | Substituição do disco de embreagem | 457,00 | 1 | | 5 | Troca do tambor de freio | 399,00 | 1 | | 65 | Substituição do disco de embreagem | 501,00 | 1 | | Saída:  Índice de sucateamento: 0  Custo médio das manutenções: R$ 452,33 |

1. A empresa “5S” procura oferecer aos seus funcionários ferramentas adequadas às práticas de trabalho exigidas pela corporação. Sabendo da necessidade de uma ferramenta de trabalho colaborativo para a “5S”, um grupo de alunos de BCC da FURB resolveu criar uma startup para alavancar investimentos visando construir tal ferramenta.

Eles decidiram iniciar o desenvolvimento da ferramenta pela construção da agenda eletrônica. A agenda deve armazenar os diferentes compromissos que uma pessoa tem ao longo dos dias.

A manipulação da agenda deve ocorrer da seguinte forma:

1. o usuário digita a data desejada. Caso já exista tal data na agenda, deve ser impressa a quantidade de compromissos de alta prioridade e os dados de todos os compromissos. Se não existe a data, está sendo criado um novo dia na agenda (é possível indicar alguma efeméride do dia (String));
2. Com o dia já criado, ele digita a hora, a descrição do compromisso, o seu tempo estimado (em minutos) e a sua prioridade (Alta, Normal, Baixa). Caso já exista um compromisso iniciando na mesma hora, este novo compromisso não deve ser inserido e o usuário deve ser avisado (usar tratamento de exceções para esta situação) – para facilitar não precisa conferir o tempo do compromisso, apenas a hora de início.

A qualquer momento o usuário pode solicitar:

1. os dados do compromisso mais curto (menor tempo) de toda a agenda;
2. o tempo médio dos compromissos de um determinado dia – digitado pelo usuário;
3. a relação de compromissos de uma determinada prioridade em um determinado dia – digitados pelo usuário.

Uma imagem contendo Tabela

Descrição gerada automaticamente

Os elementos modelados devem ser implementados. Pode haver outros métodos e atributos.

Faça um programa Java orientado a objetos que forneça as funcionalidades descritas. Mas antes de desenvolver a interface com usuário, implemente alguns testes unitários.

Exemplo dos dados:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data: 15/05/2022  Efeméride: Dia Internacional da Família. Em 1804 Napoleão Bonaparte coroa-se Imperador da França e em 1872 nasceu Bertrand Russel. | |  |  |  |  | | --- | --- | --- | --- | | Hora | Descrição | Tempo | Prioridade | | 08:00 | Missa | 120 | A | | 12:00 | Almoço | 90 | N | | 18:30 | Cinema | 100 | A | | Resultados:  Tempo Médio: 103 |
| Data: 20/05/2022  Efeméride: Dia da Independência de Timor-Leste | |  |  |  |  | | --- | --- | --- | --- | | Hora | Descrição | Tempo | Prioridade | | 14:00 | Teste de sistema | 90 | A | | 17:30 | Visita | 30 | B | | Resultados:  Tempo Médio: 60 |

1. A FURB está desenvolvendo um sistema para controlar os dados de seus alunos, tanto da graduação quanto do ensino médio (ETEVI). Para tanto, definiu a seguinte estrutura de classes:
2. Diagrama

   Descrição gerada automaticamenteA classe Aluno é classe-base para as outras duas. Portanto, AlunoEnsinoMedio e AlunoUniversitario herdam as características de Aluno.
3. Um aluno universitário está cursando algum curso oferecido pela universidade e ingressa no curso por diferentes modos: (V)estibular, (E)nem, (S)eletivo especial, (T)ransferência externa, Transferência (I)nterna.
4. O método mostra() é implementado nas classes descendentes mostrando os dados que possui (herdados ou próprios).
5. Devem ser implementados construtores e destrutores próprios para cada classe definida.
6. A interface ao usuário solicitará os dados dos alunos, questionando se é de Ensino Médio ou Universitário, para então criar o objeto correspondente.
7. Ao final, mostrar os dados completos de todos os alunos digitados em ordem alfabética (sugestão: método sort da classe ArrayList).
8. Outras validações: sigla do curso deve ter 3 letras. Nome de curso e do aluno devem ter ao menos 5 caracteres. Ano (EnsinoMedio) vai de 1 a 3.

Exemplo da saída dos dados (dados sublinhados são os dados variáveis):

Carlos Pinheiro está cursando o 1o ano do ensino médio e tem 14 anos.

João Carlos é aluno universitário do curso de SIS – Sistemas de Informação, ingressando por ENEM.

Maria da Silva é aluno universitário do curso de BCC – Ciência da Computação, ingressando por vestibular.

Paula das Neves está cursando o 3o ano do ensino médio e tem 17 anos.

Pedro Santos é aluno universitário do curso de ADM - Administração, ingressando por seletivo especial.

1. Uma impressora precisa de folhas para imprimir os dados recebidos.

As folhas possuem um limite de caracteres por linha e de quantidade de linhas. Existem dois tipos de folhas: A4 e Carta. A folha A4 suporta 20 caracteres por 12 linhas e a folha Carta suporta 8 caracteres em 10 linhas. Quando a impressora recebe uma lista de frases para imprimir, ela passa uma frase de cada vez para a folha. A folha se encarrega de imprimir os caracteres por linha suportados por ela. Os caracteres excedentes são impressos nas linhas subsequentes. Por exemplo, se a impressora passar para a folha Carta uma frase “abcdefghijklmn”, a folha irá imprimir:

abcdefgh

ijklmn

Ao enviar a frase para a folha, esta deve avisar à impressora (como retorno do método) se a folha chegou ao fim. No caso acima, se a última linha fosse “abcdefgh”, então ela não imprimiria “ijklmn”, e avisaria a impressora que chegou ao seu fim. A impressora troca de folha, porém, o texto “ijklmn” é perdido.

Sua aplicação deve permitir ao usuário digitar um texto na interface gráfica (JTextArea) e permitir escolher com que tipo de folhas a impressora está trabalhando. Ao clicar no botão Imprimir, uma simulação da impressão deve ocorrer em tela.

Especifique e implemente as classes necessárias para resolver esse problema.