

CLASSES DO PROJETO ROVER

| CLASSE: ModuloComunicacao | |
|--|--|
| ATRIBUTOS | DESCRIÇÃO |
| mensagemRecebida | Contém a mensagem que foi recebida, após ser decifrada pelo módulo. |
| mensagemTransmitida | Contém a mensagem que o módulo transmitiu utilizando a antena. |
| gastoEnergetico | É o valor que será debitado da bateria que alimenta o <i>rover</i> . |
| Antena | Possui a referência para o objeto antena, que é responsável por transmitir mensagens (de fato) para o espaço sideral. |
| MÉTODOS | DESCRIÇÃO |
| codificaMensagem(mensagem : String) : String | Codifica uma mensagem antes de repassar para a antena, utilizando qualquer algoritmo de codificação. |
| decodificaMensagem(mensagem : String) : String | Decodifica uma mensagem que chegou pela antena, para depois avisar o <i>rover</i> . |
| transmiteMensagem(mensagem : String) : void | Utilizado quando o <i>rover</i> precisa transmitir alguma mensagem para Terra. |
| recebeMensagem() : void | Ler a mensagem vinda da antena após ter recebido o aviso de notificação pelo update(). |
| update(o : Observable, arg : Object) : void | Responsável por informar ao módulo de comunicação que chegou alguma mensagem na antena. |
| notifica() : void | É chamado quando o estado do módulo de comunicação é alterado, fazendo todos os interessados no módulo serem avisados. |
| getCanal() : int | Recupera o canal de comunicação utilizado pela antena. |
| setCanal(canal : int) : void | Define o canal de comunicação utilizado pela antena. |
| setMensagem(mensagem : String) : void | Adiciona a última mensagem recebida à variável de instância mensagem. |
| getMensagem() : String | Recupera a última mensagem que foi recebida pelo módulo de comunicação. |
| getGastoEnergetico() : int | Recupera o valor que é gasto pela bateria em cada utilização de transmissão do módulo de comunicação. |

| CLASSE: Antena | |
|---|---|
| ATRIBUTOS | DESCRIÇÃO |
| mensagem | Contém a mensagem recebida pela antena do espaço sideral. |
| operacional | Informa se a antena está com seu funcionamento perfeito ou operacional. |
| canal | Informa o canal que a antena está transmitindo e recebendo. |
| MÉTODOS | DESCRIÇÃO |
| transmiteOndasRadio(mensagem : String) : void | Transmite a mensagem recebida pelo módulo de comunicação para o espaço |

| | |
|--|---|
| | sideral. Utiliza um <i>streaming</i> de binários (representação). |
| <code>recebeOndasRadio() : void</code> | Responsável por pegar o sinal vindo do espaço sideral. É chamado pelo método <code>update()</code> . |
| <code>notifica() : void</code> | É chamado quando o estado da antena é alterado, fazendo todos os interessados na antena serem avisados. |
| <code>update(o : Observable, arg : Object) : void</code> | Responsável por informar antena que chegou alguma mensagem no espaço sideral. |
| <code>getMensagem() : String</code> | Recupera a última mensagem que foi recebida pela antena. |
| <code>setMensagem(mensagem : String) : void</code> | Adiciona a última mensagem recebida à variável de instância <code>mensagem</code> . |
| <code>getCanal() : int</code> | Recupera o canal de comunicação configurado na antena. |
| <code>setCanal(canal : int) : void</code> | Define o canal de comunicação a ser utilizado pela antena. |

| CLASSE: EspacoSideral | |
|---|---|
| ATRIBUTOS | DESCRIÇÃO |
| <code>sinais</code> | Mapeamento contendo os sinais enviados para o espaço sideral. Neste mapeamento a chave é o canal e o objeto é o sinal. |
| <code>instancia</code> | Contém a referência para o único objeto da classe <code>EspacoSideral</code> (aplicação do padrão singleton). |
| MÉTODOS | DESCRIÇÃO |
| <code>getInstancia() : EspacoSideral</code> | Devolve a referência para a única instância que existe da classe <code>EspacoSideral</code> . |
| <code>getSinal(canal : int) : String</code> | Recupera a mensagem existente para um dado sinal. A mensagem, após lida, continua no espaço sideral até que uma nova mensagem seja incluída utilizando-se da mesma chave. |
| <code>setSinal(canal : int, sinal : String) : void</code> | Adiciona uma mensagem ao mapeamento (<code>HashMap</code>) de sinais. |

| CLASSE: Coordenada | |
|-----------------------------------|--|
| ATRIBUTOS | DESCRIÇÃO |
| <code>x</code> | Valor do eixo x no plano cartesiano. |
| <code>y</code> | Valor do eixo y no plano cartesiano. |
| MÉTODOS | DESCRIÇÃO |
| <code>getQuadrante() : int</code> | Informa o número do quadrante relativo à coordenada. Se estiver na origem retorna 0. |

| CLASSE: Substancia | |
|----------------------|---|
| ATRIBUTOS | DESCRIÇÃO |
| <code>nome</code> | Nome da substância (determinado aleatoriamente de uma lista). |
| <code>formula</code> | Fórmula da substância de acordo com o nome. |

| | |
|-------------------|---|
| massa | Massa da substância (até 1000g determinado aleatoriamente). |
| MÉTODOS | DESCRIÇÃO |
| Getters & Setters | |

| CLASSE: Rocha | |
|-----------------------|--|
| ATRIBUTOS | DESCRIÇÃO |
| substancias[] | Um array de substância com tamanho máximo de 10 elementos (definido aleatoriamente). |
| MÉTODOS | DESCRIÇÃO |
| getMassaTotal() : int | Retorna o total das massas das substâncias que formam a rocha. |

| CLASSE: Amostra | |
|------------------------------|---|
| ATRIBUTOS | DESCRIÇÃO |
| rocha | Referência para a rocha que faz parte da amostra. |
| coordenada | Coordenada onde a rocha foi colhida pelo rover. |
| horaLocal | Hora que a rocha foi empacotada na amostra. |
| MÉTODOS | DESCRIÇÃO |
| getRocha() : Rocha | Recupera a rocha da amostra. |
| getCoordenada() : Coordenada | Recupera a coordenada onde a rocha foi colhida. |
| getHoraLocal() : LocalTime | Recupera a hora em que a rocha foi empacotada na amostra. |

| CLASSE: DepositoAmostra | |
|--------------------------------------|--|
| ATRIBUTOS | DESCRIÇÃO |
| amostras[] | Array contendo todas as amostras que estão em espera para serem analisadas. O depósito tem o tamanho definido na criação. |
| totalAmostras | Número total de amostras presentes no depósito. No mínimo zero e no máximo o tamanho do depósito. |
| proxAmostra | Índice que controla qual será a próxima amostra que será analisada. Lembre-se que, se não existir nenhuma referência no elemento o depósito está vazio. |
| finalFila | Índice que controla qual é o índice onde uma nova amostra deve ser inserida. Lembre-se que, se o elemento já tiver alguma referência o depósito está lotado. |
| MÉTODOS | DESCRIÇÃO |
| getAmostra() : Amostra | Recupera uma amostra do depósito. |
| setAmostra(amostra : Amostra) : void | Adiciona uma amostra ao depósito. |

| CLASSE: ModuloAnalises | |
|------------------------|--|
| ATRIBUTOS | DESCRIÇÃO |
| gastoEnergetico | É o valor que será debitado da bateria que alimenta o rover. |

| | |
|--|---|
| <code>coordenadaAgua</code> | É o valor da coordenada da última amostra analisada que continha água. |
| <code>totalAgua</code> | É o total de água que foi encontrado na última amostra analisada. A cada nova análise esse valor é reiniciado. |
| <code>relatorio</code> | É um relatório informando todas as substâncias e suas respectivas massas que foram encontradas durante a análise da amostra. |
| <code>deposito</code> | Local de onde o módulo de análises tira as amostras. |
| MÉTODOS | DESCRIÇÃO |
| <code>analisaAmostra() : boolean</code> | Responsável por iniciar a análise de uma amostra. Se a amostra tiver mais de 500g de água, ele deverá retornar true (achou água). |
| <code>ModuloAnalises(deposito : DepositoAmostras)</code> | O construtor do módulo de análises deve receber o depósito de amostras. |