**Representando datas**

A classe *MyDate* modela uma data. Implemente esta classe conforme definido no diagrama abaixo e descrito nos próximos parágrafos.



A classe contém os seguintes atributos ​​de instância privativos:

* ano (int): entre 1 e 9999;
* mes (int): entre 1 (janeiro) e 12 (dezembro);
* dia (int): entre 1 e 28 | 29 | 30 | 31, considerando que o último dia depende do mês e do ano (no mês de fevereiro, para controlar anos bissextos).

A classe também contém os seguintes atributos ​de classe privativos (apresentados com sublinhado no diagrama de classes):

* strMonths (String[]): com os nomes dos meses do ano;
* strDays (String[]): com os nomes dos dias da semana;
* dayInMonths (int[]): com o número de dias por mês.

A classe tem os seguintes métodos de classe públicos (apresentados com sublinhado no diagrama de classes):

* IsLeapYear (ano): retorna TRUE se o ano recebido como parâmetro é bissexto. Um ano é bissexto se for divisível por 4 mas não por 100 ou se for divisível por 400;
* isValidDate (ano, mes, dia): retorna TRUE se o ano, mês e dia recebidos como parâmetro constituem uma data válida, considerando anos bissextos;
* getDayOfWeek (ano, mes, dia): retorna o dia da semana para uma determinada data, sendo que 0 representa domingo, 1 representa segunda, ..., 6 representa sábado. Retorna -1 se a data não for válida[[1]](#footnote-1).

A classe tem um construtor com três parâmetros: ano, mês e dia. Ele deve chamar o método setDate() (descrito abaixo) para definir os valores dos atributos ​​de instância.

Por fim, a classe tem os seguintes métodos públicos:

* *setDate*(ano, mes, dia): chama o método *isValidDate*() para verificar se o ano, mês e dia recebidos como parâmetro constituem uma data válida. Preenche a data com zeros, caso os parâmetros recebidos não sejam válidos;
* *setYear*(ano): verifica se o ano recebido como parâmetro está entre 1 e 9999. Preenche o ano da data corrente com zero, caso o ano recebido como parâmetro não seja válido;
* *setMonth*(mes): verifica se o mês recebido como parâmetro está entre 1 e 12, de forma similar ao método anterior;
* *setDay*(dia): verifica se o dia recebido como parâmetro está entre 1 e o número de dias do mês atual, considerando anos bissextos;
* *getYear*(), *getMonth*(), *getDay*(): retorna o valor para o ano, mês e dia, respectivamente;
* *toString*(): retorna uma string no formato "xxx dd de mmm, aaaa". Por exemplo, um resultado poderia ser "Terça-feira 14 de fevereiro, 2012";
* *nextDay*(): atualizar a instância para o dia seguinte da data atual e retornar a instância. Lembre-se que *nextDay*() para 31/12/2000 será o dia 01/01/2001;
* *nextMonth*(): atualizar a instância para o próximo mês e retornar a instância. Considere que nextmonth() para 31/10/2012 será o dia 30/11/2012;
* *nextYear*(): atualizar a instância para o próximo ano e retornar a instância. Lembre-se que nextYear() para 29/02/2012 será o dia 28/02/2013;
* *previousDay(), previousMonth(), previousYear():* similar aos métodos anteriores*.*

1. Algoritmo em http://www.inf.ufrgs.br/~cabral/Dia\_do\_Seculo.html. [↑](#footnote-ref-1)