Prática 5 Classes

Objetivos

- Compreender os conceitos de classe e instâncias
- Aplicar metodologias orientadas a objetos

Tópicos

- Construtores, atributos e métodos
- Métodos especiais (`toString(), equals(), get…(), set…(), ...`)

Exercício 5.1

Crie uma classe que permita modelar uma data (class *DateYMD*).
Esta classe deve conter os seguintes métodos estáticos:
- um método booleano que indique se o valor inteiro que represente um mês ([1;12]) é válido: validMonth(int month)
- um método inteiro que devolva o número de dias de um determinado mês, num determinado ano: monthDays(int month, int year)
- um método booleano que indique se um ano é bissexto: leapYear(int year)
- um método booleano que indique se uma data composta por dia, mês e ano, é válida: valid(int day, int month, int year)

(Nota: Ao desenvolver estes métodos aproveite métodos desenvolvidos em aulas anteriores.)

A classe deve também permitir instanciar objetos que representem uma data específica (válida). Nesse sentido, considere que a representação interna do objeto é composta por três atributos inteiros (day, month, year).
Deve ser possível aplicar externamente as seguintes operações sobre objetos deste tipo:
- definir uma data: set(int day, int month, int year);
- consultar os valores do dia, mês e ano (day, month, year);
- incrementar a data (increment);
- decrementar a data (decrement);
- método toString que devolva a data no formato AAAA-MM-DD.

A classe deve ter um construtor que defina uma data (válida) indicando um dia, mês e ano.

(Nota: desenvolva a classe garantindo que não é possível que nenhum dos seus objetos represente uma data inválida [por exemplo, 31 de fevereiro de 2022].)

Para testar esta classe, crie um programa de teste, com o seguinte menu:

```
Date operations:
1 - create new date
2 - show current date
3 - increment date
4 - decrement date
0 - exit
```

Exercício 5.2

Construa uma classe que represente um calendário. Esta classe deve permitir registar o número de eventos agendados para cada dia do ano, sugerindo-se para isso o uso de um vetor bidimensional de inteiros.
A classe deve fazer uso da classe *DateYMD* desenvolvida no exercício anterior, e deve incluir:
- um construtor que recebe o ano e o dia da semana (entre 1-domingo e 7-sábado) em que começa o ano;
- métodos que devolvem esses dados (consultas/getters): year() e firstWeekdayOfYear();
- um método que devolva o dia da semana em que começa um dado mês (no ano do calendário): firstWeekdayOfMonth(month);
- métodos que permitam adicionar/remover um evento numa data: addEvent(*DateYMD*); removeEvent(*DateYMD*);
- método toString que devolva a representação de um mês de calendário: printMonth(month); nesta representação, cada dia deve ser precedido de * caso exista pelo menos um evento agendado nessa data;
- método toString que devolva o calendário para todo o ano:

```
        January 2023
 Su  Mo  Tu  We  Th  Fr  Sa
  1   2   3   4   5   6   7
  8   9  10  11  12 *13  14
 15  16  17  18  19  20  21
 22  23  24  25  26  27  28
 29  30  31

       February 2023
 Su  Mo  Tu  We  Th  Fr  Sa
              1   2   3   4
 *5   6   7   8   9  10  11
 12  13  14  15  16  17  18
 19  20  21  22  23  24  25
 26  27  28

...
```

Para testar esta classe, crie um programa de teste, com o seguinte menu:

```
Calendar operations:
1 - create new calendar
2 - print calendar month
3 - print calendar
0 – exit
```