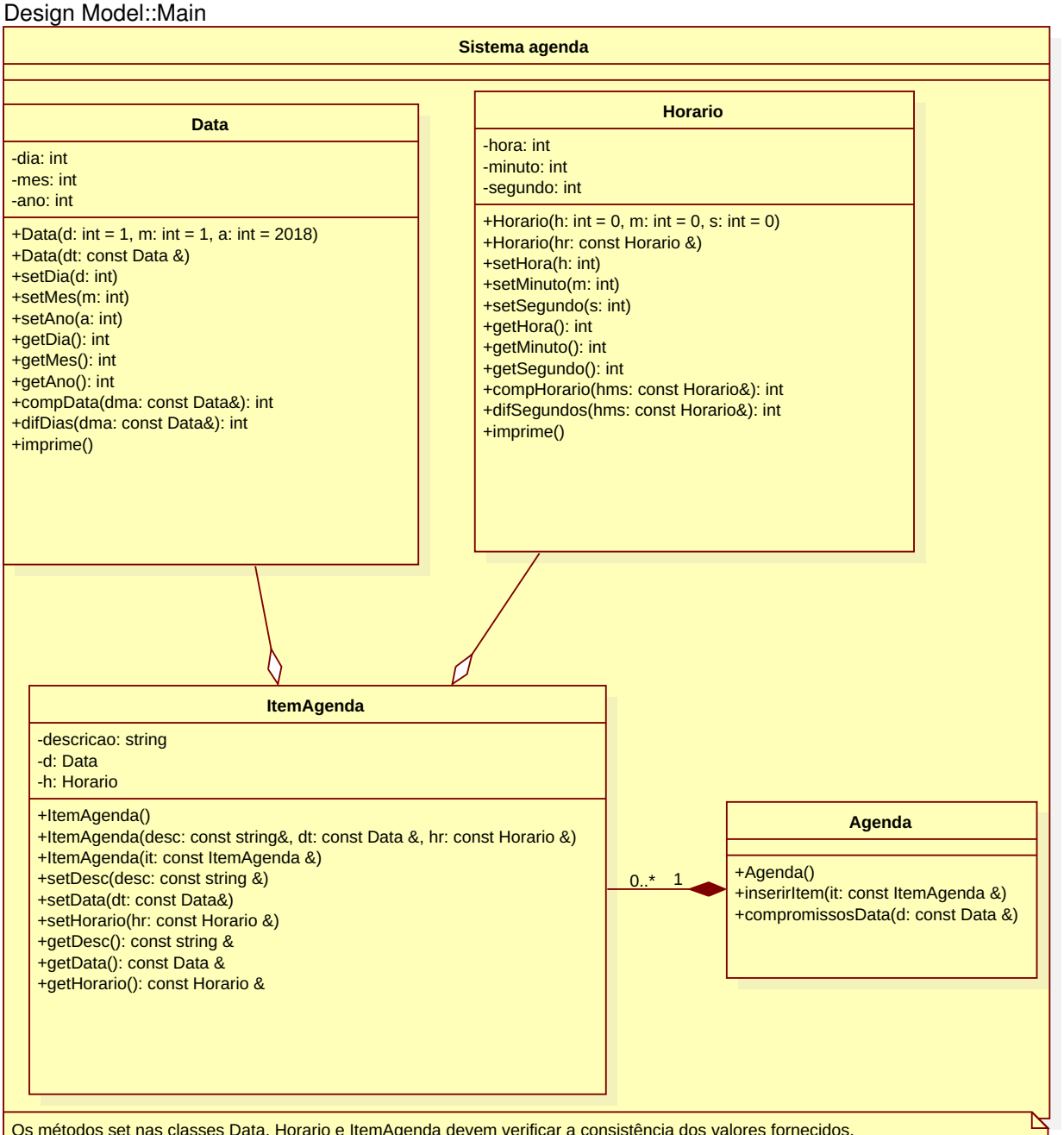
Design Model::Main



Os métodos set nas classes Data, Horario e ItemAgenda devem verificar a consistência dos valores fornecidos.
Por exemplo, na classe Data os valores nos atributos devem respeitar as seguintes restrições:

1 <= dia <= 30; 1 <= mes <= 12; 2018 <= ano <= 2020

E na classe Horario as restrições sâo:

0 <= hora < 24
0 <= minuto < 60
0 <= segundo < 60

Caso algum valor fornecido esteja fora do intervalo válido, o método deve arredondar o valor para o valor válido mais próximo. Por exemplo, na classe Horario, se o valor fornecido para hora for 27, o valor devera ser substituido por 23 (por simplicidade, nao iremos emitir mensagens de erro). Ou se na classe Data o valor do mes for 0, ele foi substituído por 1.

Os métodos compData e compHorario retornam -1, 0 ou +1 indicando se o seu objeto é menor, igual ou maior do que o objeto passado como parâmetro

O método difDias da classe Data retorna a diferenca em dias entre seu objeto e o objeto passado como parâmetro

O método difSegundos da classe Horario retorna a diferenca em segundos entre seu objeto e o objeto passado como parâmetro

O método compromissos da classe Agenda deve imprimir todos os compromissos da data d passando como parâmetro. Os compromissos na dada data devem ser impressos na mesma ordem em que foram originalmente inseridos na agenda. Para imprimir um compromisso, imprima o horario, um espaco em branco, a descricao e uma nova linha.

Nas classes Data e Horario inclua a sobrecarga dos operadores de leitura >> (via cin) e escrita << (via cout).

Ao ler/escrever valores, considere os valores de datas do menos significativo para o mais significativo (i.e., dia, mes, ano) e os valores de horario do mais significativo para o menos significativo (i.e., hora, minuto, segundo).

Metodos que nao modificam o objeto (exemplo: getHorario()) devem ser marcados como const na implementacao.