DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

First Semester 2017/2018

Polígonos Regulares

Estruturas de Informação

A designação dos polígonos quanto ao seu número de lados (até 999 lados) pode ser estendida apenas com base na seguinte tabela de prefixos.

Polygon names

Ones		Tens		Twenties		Thirties+		Hundreds	
		10	deca-	20	icosa-	30	triaconta-	100	hecta-
1	hena-	11	hendeca-	21	icosi-hena-	31	triaconta-hena-	200	dihecta-
2	di-	12	dodeca-	22	icosi-di-	32	triaconta-di-	300	trihecta-
3	tri-	13	triskaideca-	23	icosi-tri-	33	triaconta-tri-	400	tetrahecta-
4	tetra-	14	tetrakaideca-	24	icosi-tetra-	40	tetraconta-	500	pentahecta-
5	penta-	15	pentakaideca-	25	icosi-penta-	50	pentaconta-	600	hexahecta-
6	hexa-	16	hexakaideca-	26	icosi-hexa-	60	hexaconta-	700	heptahecta-
7	hepta-	17	heptakaideca-	27	icosi-hepta-	70	heptaconta-	800	octahecta-
8	octa-	18	octakaideca-	28	icosi-octa-	80	octaconta-	900	enneahecta-
9	ennea-	19	enneakaideca-	29	icosi-ennea-	90	enneaconta-		

https://en.wikipedia.org/wiki/List_of_polygons

Pretende-se que desenvolvam uma biblioteca de classes, respetivos métodos e testes, que permitam gerir várias árvores balanceadas com o nome dos polígonos e o seu número de lados bem como algumas funcionalidades.

a) Comece por construir três árvores balanceadas: uma para unidades (1~9), outra para dezenas (10~90) e outra para centenas (100~900) utilizando os ficheiros fornecidos.

b) Com base nas três árvores anteriores elabore um método que devolve o nome de um polígono dado o seu número de lados. Por exemplo, 524 é pentahecta-icosi-tetra-gon -> pentahectaicositetragon

c) Com base nas três árvores anteriores construa uma árvore balanceada que contenha todos os nomes de polígonos regulares de 1 até 999 lados.

d) Elabore um método que devolve o número de lados de um polígono regular através do seu nome.

e) Dado um intervalo de números de lados o método deve devolver os correspondentes nomes dos polígonos por ordem inversa, do maior para o menor número de lados.

f) Dados dois nomes de polígonos encontrar o seu *Lowest Common Ancestor* (antecessor comum mais próximo) na árvore binária.

Data Entrega até 10-Dez-2017

Normas

- A avaliação do trabalho será feita principalmente em função das classes propostas, nomeadamente em termos da sua conformidade com o Paradigma da Programação por Objetos e **eficiência** das estruturas de dados usadas e funcionalidades solicitadas.
- O trabalho deverá ser realizado em **grupos de dois alunos**. Alterações de grupos têm de comunicadas por *email* ao docente das aulas PL, até ao dia **24 Novembro**.
- O projeto deve ser desenvolvido em Java e todas as funcionalidades testadas através de testes unitários e usando os **ficheiros de teste disponibilizados**.
- É obrigatório o uso da ferramenta de controle de versões Git.
- O relatório deverá ser elaborado incrementalmente para cada uma das partes do trabalho e deve servir de ferramenta de avaliação posterior à apresentação. Nele devem apresentar o diagrama de classes, algoritmos dos métodos, análise de complexidade de todas as funcionalidades implementadas, melhoramentos possíveis, ….
- Cada Parte do trabalho deverá ser submetida no moodle até às **24 horas do dia indicado**. A partir das datas indicadas, a nota do trabalho será penalizada **10% por cada dia de atraso** e não se aceitam trabalhos **após dois dias** das datas indicadas.
- A apresentação/avaliação do trabalho final será individual, em datas a fixar com o professor das aulas práticas, na semana de **11-15 Dezembro**.