

IP – Semana 4 – Lista de Problemas
(alunos em 1ª inscrição)

Miguel Goulão
miguel.goulao@di.fct.unl.pt

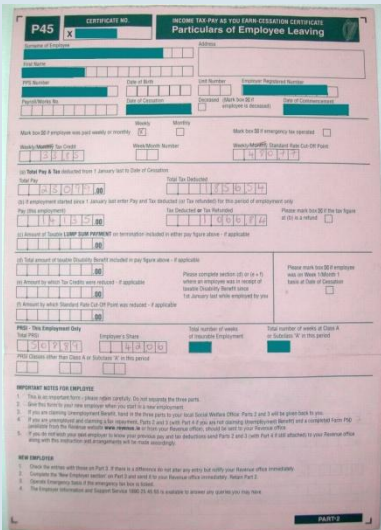
Objetivos

- No final da 4ª semana, o aluno deverá ser capaz de:
 - Usar o BlueJ na construção de classes simples, definindo:
 - Constantes, variáveis de instância
 - Múltiplos construtores de instâncias da classe, operações modificadoras e de consulta, usando:
 - Chamadas a operações da classe dentro de outras operações
 - Operações aritméticas na implementação de métodos
 - Operações da biblioteca Math
 - Operações lógicas na implementação dos métodos
 - Variáveis locais/temporárias dentro dos métodos
 - **Instruções de controle: if-then-else e switch**
 - Dada uma especificação de uma classe em língua natural, modelar a classe, recorrendo aos elementos acima descritos

Simulador do IRS

Simulador do IRS

- Defina em Java uma classe IRSSimulator cujos objectos permitem o cálculo do IRS (Imposto sobre Pessoas Singulares) para os escalões existentes em Portugal.
- Programe a sua classe no BlueJ.
- Teste um (ou vários) objectos IRSSimulator, e verifique se se comportam como esperado.



Simulador do IRS

- É determinado o escalão de imposto consoante o montante da matéria colectável (aqui chamado rendimento bruto).
- O imposto a pagar corresponde à aplicação da taxa para esse escalão, menos uma parcela a abater.
- A tabela do cálculo do IRS a pagar em 2009 segue a tabela abaixo.

Taxas gerais 2009						
		Escalões			Taxa	Parcela a abater
1º		até	4.755		10,50%	-
2º	De mais	4.755	até	7.192	13,00%	118,87
3º	De mais	7.192	até	17.836	23,50%	874,03
4º	De mais	17.836	até	41.021	34,00%	2.746,81
5º	De mais	41.021	até	59.450	36,50%	3.772,33
6º	De mais	59.450	até	64.110	40,00%	5.853,08
7º		superior a	64.110		42,00%	7.135,28

Simulador do IRS

- Cada objecto IRSSimulator deve conter a informação constante da tabela do IRS 2009
 - Para as parcelas a abater, deve ser usada somente a parte inteira (118, 874, 2746, 3772, 7135).
- Cada objecto IRSSimulator é criado indicando o rendimento bruto
- Sobre esse rendimento bruto, o objecto deve calcular os seguintes componentes:
 - escalão aplicável (1-7)
 - taxa de IRS correspondente ao escalão
 - montante a pagar de imposto
 - líquido restante
- Os valores de montantes são em euros (sem cêntimos)

Simulador do IRS

- Operações reconhecidas (interface de IRSSimulator) :

```
public void setGrossIncome(int grossIncome)
```

Atribui novo valor para o rendimento bruto.

```
public int grossIncome()
```

Devolve o valor do rendimento bruto.

```
public float taxRate()
```

Taxa de IRS aplicável ao rendimento bruto registado no objecto.

```
public int taxPayable()
```

Valor do imposto a pagar para o rendimento bruto registado no objecto.

```
public int netIncome()
```

Valor do líquido restante após cobrança do imposto.

```
public int taxBracket()
```

Escalão (1-7) correspondente ao rendimento bruto registado no objecto.

Simulador do IRS

- Operações reconhecidas (interface de IRSSimulator) :

```
public int deductible()
```

Parcela a abater ao montante a pagar (coluna direita da tabela de IRS 2009)

```
public String asString()
```

Devolve uma cadeia de caracteres (ou seja, uma *String*) mostrando:

- Rendimento bruto
- Escalão a que corresponde
- Imposto a pagar
- Líquido restante

Nota: veja o formato da cadeia de caracteres no traço de execução apresentado no slide seguinte

Simulador do IRS

```
IRSSimulator irs = new IRSSimulator(15000);
irs.taxBracket()
3 (int)
irs.taxRate()
0.235 (float)
irs.deductable()
874 (int)
irs.taxPayable()
2651 (int)
irs.netIncome()
12349 (int)
irs.asString()
"Gross: 15000, bracket 3, 2651 payable, net 12349" (String)
irs.setGrossIncome(65000)
irs.taxBracket()
7 (int)
irs.taxRate()
0.42 (float)
irs.taxPayable()
20165 (int)
irs.netIncome()
44835 (int)
```

Stand de Limonada

Recordemos o exercício que
começou a fazer na teórico-prática

Stand de Limonada

- Este exercício **não é para programar, mas para analisar.**
- No entanto, se quiser, poderá fazê-lo como exercício adicional.



Stand de Limonada

- Imagine que é o feliz possuidor de um *stand* de venda de limonada e quer fazer uma aplicação para gerir tanto as vendas da limonada, como o *stock* de ingredientes.
- Os únicos ingredientes utilizados são limões, açúcar e água.
- A receita de tanto sucesso é utilizar 5 limões, 100g de açúcar e 1l de água por cada *jarro*, que dá para 5 copos.
- Cada copo é vendido por 0.75€.
No entanto, se uma pessoa comprar 5 ou mais copos de uma vez, o preço *baixa* para 0.65€.
- Uma vez que é necessário que esteja sempre alguém para efectuar as vendas, é possível através deste sistema *encomendar* uma certa quantidade de um ingrediente para repor os stocks.
Assume-se que a quantidade encomendada fica logo disponível.

Stand de Limonada

- O espaço do *stand* é limitado, pelo que não é possível ter em *stock* mais do que 50 limões, 5kg de açúcar e 30l de água.
- É necessário controlar o número total de vendas (tanto a preço normal como a preço reduzido), bem como o valor total das vendas (dinheiro em caixa).
- Analise o enunciado com cuidado e indique:
 1. As constantes necessárias: nome, tipo e o que representa.
 2. As variáveis internas à classe que utilizaria: nome, tipo e o que a variável representa.
 3. As operações (métodos) a definir para este sistema: nome, tipo de retorno, parâmetros e objectivo pretendido.

E agora, implemente o Stand!

Já o implementou tudo?

Se a resposta for sim, recupere esse trabalho e acrescente-lhe o necessário para suportar a aquisição de ingredientes e o cálculo de lucros!

- Primeiro, enriqueça a especificação do problema.
- Depois, implemente-a.