|  |  |
| --- | --- |
| logoiefpnovo-cor | **INSTITUTO DO EMPREGO E FORMAÇÃO PROFISSIONAL, I. P.** |
|  | DELEGAÇÃO REGIONAL DO NORTE  CENTRO DE EMPREGO E FORMAÇÃO PROFISSIONAL DE BRAGA |

2023

Plano de TReino

Unidade 19 (0816) - Java

IEFP, IP – Centro de EMprego e Formação Profissional de Braga | Rui Boticas 2023

1. **Cronograma**

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMANA 01** | **DATA INÍCIO** | | **DATA FIM** |
|  | |  |
| **Tema** | Solidificar conhecimentos em conceitos estruturantes da programação | |
| **Objetivos** | * Adquirir conhecimentos sobre os tipos primitivos das varáveis e conceitos associados; * Adquirir/solidificar conhecimentos sobre o espaço que as variáveis ocupam em memória; * Conhecer o escopo/âmbito de uma variável; | |
| **Exercícios** | [**Faça Click Aqui**](#exercicios_semana_1) | |

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMANA 02** | **DATA INÍCIO** | | **DATA FIM** |
|  | |  |
| **Tema** | Estruturas de controlo do Java;  Estruturas cíclicas;  Variáveis estruturadas (“arrays”). | |
| **Objetivos** | * Solidificar conhecimentos sobre variáveis; * Tomar/solidificar conhecimentos sobre varáveis estruturadas e manipuladas sem qualquer dúvida; * Elaborar métodos que permitam criar estruturas de controlo de fluxo de um programa; * Manipular de forma eficiente um “array” de qualquer tipo, bem como as posições que este contém; | |
| **Exercícios** | [**Faça Click Aqui**](#exercicios_semana_2) | |

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMANA 03** | **DATA INÍCIO** | | **DATA FIM** |
|  | |  |
| **Tema** | Trabalhar os objetos e sua estrutura | |
| **Objetivos** | * Aperfeiçoar o uso das variáveis compreendendo a sua visibilidade (publicas, privadas, etc); * Consolidar o conhecimento das constantes; * Adquirir competências de manuseamento de objetos; * Aprender a criar métodos de instância e efetuar correta invocação dos mesmos | |
| **Exercícios** | [**Faça Click Aqui**](#exercicios_semana_3) | |
|  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMANA 04** | **DATA INÍCIO** | | **DATA FIM** |
|  | |  |
| **Tema** | Métodos de acesso a variáveis GET e SET. Tratamento de erros através de exceções | |
| **Objetivos** | * Trabalhar os construtores e todas as nuances desta técnica; * Aquisição de competências de criação de métodos de acesso a variáveis GET e SET; * Dominar o tratamento de erros de forma a que os programas sejam executados sem sobressaltos através do mecanismo das exceções; | |
| **Exercícios** | * [**Faça Click Aqui**](#exercicios_semana_4) | |
|  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMANA 05** | **DATA INÍCIO** | | **DATA FIM** |
|  | |  |
| **Tema** | O mecanismo da Herança e sua enorme potencialidade. | |
| **Objetivos** | * Trabalhar os métodos e variáveis herdados, ficando com capacidade de criar métodos e mais tarde redefini-los. * Trabalhar os métodos de construção prioritária toString(), clone(), equals(); | |
| **Exercícios** | * [**Faça Click Aqui**](#exercicios_semana_5) | |
|  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMANA 06** | **DATA INÍCIO** | | **DATA FIM** |
|  | |  |
| **Tema** | Criação de variáveis e métodos de classe | |
| **Objetivos** | * Perceber a utilidade dos componentes de classe e sua visibilidade em todo o projeto; * Criar métodos e variáveis de classe e usá-los para facilitar o trabalho de dados que tem de ser transversais a qualquer instância; | |
| **Exercícios** | * [**Faça Click Aqui**](#exercicios_semana_6) | |
|  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMANA 07** | **DATA INÍCIO** | | **DATA FIM** |
|  | |  |
| **Tema** | Criação de aplicações no ambiente gráfico;  Aplicação das exceções no ambiente gráfico; | |
| **Objetivos** | * Dominar os principais mecanismos de manuseamento de componentes do ambiente gráfico; * Associar o mecanismo do tratamento dos erros ao manuseamento do ambiente gráfico; | |
| **Exercícios** | * [**Faça Click Aqui**](#exercicios_semana_7) | |
|  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **SEMANA 08** | **DATA INÍCIO** | | **DATA FIM** |
|  | |  |
| **Tema** | Produção | |
| **Objetivos** | * Avaliar o grau de aquisição de conhecimentos do formando. | |
| **Produção** | * [**Faça Click Aqui**](#exercicios_semana_8) | |
|  | | |

1. **Exercícios**

**Semana 1**

1. Sem recorrer à programação responda às seguintes questões:
   1. Qual a utilidade de uma variável?

Guarda temporariamente um valor

* 1. O que é uma constante?

É um recipiente onde é armazenado um valor imutável.

* 1. Fisicamente no computador, qual a diferença entre uma variável e uma constante?  
      A variável pode ser alterada depois da sua declaração, a constante não.
  2. O que é uma variável estruturada?  
      São variáveis compostas por outras variáveis.
  3. Qual a diferença entre uma variável simples e uma estruturada?  
      A variável estruturada pode conter outras variáveis.
  4. Em Java, que tipos de variáveis primitivos temos?  
      Byte, short, int, long, float, double, boolean, char
  5. Existe um tipo de dados que, embora não seja primitivo, é tratado como tal na maior parte dos casos. Qual é o tipo?  
      A string.
  6. Quantos bits ocupa na memória do computador, uma variável do tipo inteira?  
      32 bits (4 bytes).
  7. Se tivesse de guardar os seguintes dados, que tipo(s) primitivo(s) deveria usar?
     1. Nome de um gato;  
         String.
     2. Nome de uma rua;  
         String.
     3. Idade de uma criança com menos de um ano de idade;  
         Int.
     4. As coordenadas GPS de uma loja em Lisboa;  
         Int
     5. O preço de uma vaca;  
         Float.
     6. A cor do chapéu de uma senhora;  
         String.
     7. O número de golos da seleção nacional;  
         Int.

1. Agora já em ambiente de programação, crie um projeto chamado “semana1”.
   1. Crie um package chamado “variaveis”.
   2. Crie uma classe “TreinoBaseVariaveis”.
   3. Crie nesta classe um método “main” (que torna esta classe executável).
   4. Dentro do método “main”, crie uma variável para guardar a idade de uma pessoa;
   5. Posso nesta variável colocar a idade 34.0 anos? Teste e tire uma conclusão.  
       Não porque 34.0 é do tipo double.
   6. Crie uma variável para guardar o salário de uma pessoa;
   7. Verifique se o jovem é maior de idade;
   8. Mostre quantos anos faltam/passaram da idade de maior idade;
   9. Crie no fim do método main o seguinte código:

int i = 3;

int j = 12;

if(i > j){

double t = 0;

t = i + 3;

}else{

double t = 0;

t = j + 3;

}

System.out.println(t+“”);

* 1. O que está a causar os erros no código?  
      A variável t está fora do âmbito.

**Semana 2**

1. Crie um projeto chamado “semana2”;
2. Crie uma classe executável com o nome “TesteVariaveisEstruturadas”?
3. No método “main”, crie variáveis para representar o salário de três pessoas (1 variável por cada pessoa);
4. Preencha as três variáveis com salários fictícios;
5. Calcule a média dos salários e mostre no ecrã. Confirme os valores calculados;
6. Mostre no ecrã o salário mais alto. Confirme se está correto;
7. Crie agora uma variável para introduzir o nome do administrador da empresa e preencha-o.
8. Mostre no ecrã quantas letras tem o nome do administrador;
9. Verifique se o número de letras do nome do administrador, é maior que 20 vezes o salario médio.
10. Os funcionários desta empresa são 5. Vamos criar o código para dar os bons-dias quando os funcionários entrarem de manhã. Desta forma, crie uma estrutura cíclica, que imprima no ecrã a frase “Bom dia” a cada funcionário;
11. Crie agora uma estrutura que permita guardar 5 salários;
12. Crie agora uma estrutura que permita guardar 5 nomes de funcionários;
13. Preencha os dois “arrays” anteriores com os nomes e os salários;
14. Calcule de novo o salário médio, o máximo e mínimo;
15. Crie mais um array com cinco posições do tipo booleano para guardar os dados relativos ao pagamento dos funcionários;
16. Faça com que dois funcionários tenham o registo neste array como falso, isto é, ainda não tenham recebido o salario do mês.
17. Calcule o total de salários por receber.
18. Crie código para contar quantos funcionários receberam salário;
19. Imprima no ecrã quais os funcionários que receberam salário. Deve procurar no array que tem o registo se o funcionário recebeu, ou não e mostrar o respetivo nome no ecrã.

**Semana 3**

1. Crie um projeto chamado “semana3”;
2. Crie uma classe executável com o nome “TesteVariaveisEstruturadas”?
3. Para os exercícios que se seguem, deve usar uma estrutura de dados do tipo “ArrayList”. Crie uma estrutura que permita guardar 5 salários;
4. Crie agora uma estrutura que permita guardar 5 nomes de funcionários;
5. Preencha os dois “ArrayList” anteriores com os nomes e os salários;
6. Calcule de novo o salário médio, o máximo e mínimo;
7. Crie mais um “ArrayList” com cinco posições do tipo booleano para guardar os dados relativos ao pagamento dos funcionários;
8. Faça com que dois funcionários tenham o registo neste “array” como falso, isto é, ainda não tenham recebido o salario do mês.
9. Calcule o total de salários por receber.
10. Crie código para contar quantos funcionários receberam salário;
11. Imprima no ecrã quais os funcionários que receberam salário. Deve procurar no “array” que tem o registo se o funcionário recebeu, ou não e mostrar o respetivo nome no ecrã.
12. Como pode ver, é complicado armazenar valores respeitantes à mesma pessoa em três “arrays” diferentes. Ideal seria, contruir algo que conseguisse armazenar todas as variáveis respeitantes a cada pessoa, numa só estrutura. Para isso, vamos construir uma classe com o nome “Funcionário”. Esta classe terá três variáveis privadas. Uma para guardar o nome, outra para guardar o salário e outra para registar se o salário do mês atual já foi recebido.

**NOTA**: verifique a funcionalidade de uma classe/objeto. Possibilita o armazenamento de vários valores, numa só estrutura, com a possibilidade de apenas ele próprio manusear essas varáveis, isto é, nenhuma outra classe tem acesso às suas variáveis.

1. Vamos agora criar uma classe executável com o nome “TesteEmpresa”. Desta forma, se é executável tem um método “main”.
2. Dentro do método “main”, vamos agora criar a primeira instância da classe Funcionário. Vamos chamar a essa variável funcionário1.
3. Como é sabido para invocarmos qualquer método de um objeto, devemos primeiro colocar o nome do objeto seguido de um ponto. Vamos tentar mostrar o valor do salário desse funcionário através do código seguinte:

System.out.println(funcionario1.salario);

Funcionou? Porquê?

1. Vamos então criar um método para podermos aceder ao valor da variável “salario”. Crie um método chamado getSalario() que devolve o valor do salário da pessoa.
2. Agora tente imprimir no ecrã o salário desse funcionário através do código

System.out.println(funcionario1.getSalario());

Funcionou? Porquê?

1. Vamos agora tentar modificar o valor do salário deste funcionário. Coloque para isso o seguinte código:

Funcionario1.salario = 1000;

System.out.println(funcionario1.getSalario());

Funcionou? Porquê?

1. Crie então um método próprio para alterar o salário;
2. Crie métodos para obter o valor das variáveis;
3. Crie métodos para modificar o valor das variáveis;
4. Crie uma nova classe executável com o nome “TesteMultidão”;
5. Crie dentro do método “main” um arraylist para guardar registos do tipo da classe que definimos anteriormente “Funcionário”;
6. Preencha o arraylist com cinco pessoas, dando-lhes nome, salario;
7. Faça com que duas dessas pessoas não tenham ainda recebido o salário deste mês;
8. Calcule o salário médio;
9. Das pessoas que receberam salário, qual foi o salário mais alto?
10. Das pessoas que não receberam salário, qual o menor salário?

**Semana 4**

1. Crie um projeto chamado “semana4”;
2. Crie uma classe com o nome “Paiis” (país). Não se esqueça que, **não** deve colocar o acento no nome da classe;
3. Crie uma variável nesta classe, para guardar a língua falada nesse país;
4. Crie os métodos GET e SET para esta variável (crie manualmente, isto é, não utilize a ferramenta de geração de código “ALT + INSERT”);
5. Crie agora uma classe de teste para a classe “Paiis”;
6. Na classe de teste, crie um objeto do tipo do “Paiis”, com o nome “portugal”;
7. Mude o nome da língua falada nesse país para português;
8. Imprima para o ecrã, a língua falada nesse país;
9. Funcionou de acordo com o esperado? Se não, corrija.
10. Os construtores, são úteis para passar valores a um objeto, aquando da sua construção (instanciação). Os próximos exercícios, vêm exercitar o manuseamento deste tópico. Assegure-se que, no final compreendeu a finalidade, modo de construção e aplicação de um construtor (é muito importante). Um construtor é, um método muito especial. Quando é criado um objeto a partir de uma classe (criação de uma instância), este é o primeiro método a ser invocado, e isto acontece apenas uma vez. Este deve conter as instruções a serem executadas, quando o objeto está a ser criado. Se o programador não elaborar nenhum construtor, o Java por omissão fornece um sem parâmetros. Assim que um construtor é definido por nós, o que existe por omissão desaparece. Variáveis e constantes, devem ser inicializadas dentro do construtor, quer com valores locais, quer com valores passados como parâmetro.

Crie um construtor sem parâmetros e inclua uma instrução que inicialize a variável da língua falada no país, com a frase vazia;

1. Crie na classe “Paiis” as seguintes propriedades variáveis:
   1. Número da população;
   2. Nome presidente/rei (se o modo de governação for uma monarquia);
   3. Nome primeiro ministro;
   4. Valor do Produto Interno Bruto;
2. Deve ainda incluir as propriedades constantes:
   1. Nome;
   2. Continente;
3. Crie métodos GET e SET para as variáveis recém-criadas;
4. Como temos constantes e estas devem ser inicializadas no construtor com os valores passados como parâmetro, deve remover agora o construtor sem parâmetros, para dar lugar a um outro. Transfira todas as instruções do construtor a apagar para o novo;
5. Implemente agora um construtor, desta desta vez com parâmetros. Todas as constantes devem ser inicializadas com os valores passados como parâmetro neste construtor;
6. A sua classe deverá agora estar com alguns erros, uma vez que adicionou à sua classe constantes e estas não foram inicializadas.
7. Deverá agora fazer ajustes na classe de teste. Lembre-se que deixou de existir o construtor sem parâmetros. Agora tem um construtor cujos parâmetros são os valores a atribuir às constantes. Corrija a situação;
8. Na classe de teste, atribua valores às outras variáveis do objeto “portugal”, através dos métodos definidos para o efeito;
9. Imprima agora todos os dados do objeto presente na classe de teste “portugal”. Os valores estão corretos? Se não, corrija-os;
10. Crie na classe um método que permita calcular o rendimento por cada habitante (produto interno bruto / numero de habitantes);
11. O mecanismo de tratamento de erros, é um instrumento que permite a um programa continuar a sua execução, mesmo depois de ter acontecido um erro. Crie uma classe de Exceção com o nome “ValorNumericoInvalidoException” a partir da classe Exception, onde vai buscar todos os métodos e variáveis.
12. Agora, nos métodos que fazem a alteração dos valores da área geográfica e do numero da população, implemente o lançamento de uma exceção se o valor passado por parâmetro for inferior a zero;
13. Este procedimento originou um erro na sua classe de teste, uma vez que previmos a possibilidade de lançamento de uma exceção na classe “Paiis” e ainda não fizemos o seu tratamento. Implemente o tratamento da possibilidade dos métodos que alteram a área geográfica e o número da população originarem erro, pela introdução de valores negativos;
14. Crie, agora uma exceção para ser lançada quando o nome do primeiro ministro for uma frase vazia;
15. Agora coloque o tratamento do erro de pode ser lançado quando usar os métodos para alteração do nome do primeiro ministro;

**Semana 5**

1. Crie um projeto chamado “semana5”;
2. Copie as classes da semana 4 para este novo projeto;
3. Crie uma classe com o nome “Cidade”;
4. Crie as variáveis dentro desta classe
   1. Nome;
   2. Presidente da camara;
5. Transferira as variáveis da classe “Paiis” para a classe recém-criada:
   1. Numero de população;
6. Transfira também seguintes as constantes:
   1. Área geográfica;
   2. Tem costa marítima (Sim/Não);
   3. Conjunto dos nomes dos rios que passam no país;
7. Transfira os métodos GET e SET para a classe “Cidade”;
8. Refaça o construtor de acordo com as alterações produzidas, com a transferência de variáveis e constantes, para a classe “Cidade”;
9. Crie na classe” Paiis”, um arraylist de cidades;
10. Crie um método para adicionar novas cidades ao país;
11. Crie um método para remover uma cidade;
12. Crie um método para substituir uma cidade que está numa determinada posição do arraylist por outra;
13. Crie um método no país para determinar qual a população total;
14. Crie um método para calcular qual a população média de cada cidade;
15. Crie um método para calcular qual é a cidade mais populosa;
16. Crie um método para calcular qual é a cidade menos populosa;
17. Qual a cidade que é atravessada por mais rios;
18. Quais os rios que atravessam o país;
19. O país tem costa marítima?
20. Crie o projeto “semana5\_heranca”. Neste projeto, irá trabalhar o mecanismo de herança;
21. Crie uma classe “Formando”, que irá ser uma superclasse;
22. Crie as variáveis dentro desta classe:
    1. Nome;
    2. Número de formando;
    3. Nota UFCDs;
23. Crie métodos GET e SET, para as variáveis que o justifiquem;
24. Crie agora, uma subclasse de formando, com o nome “FormandoEFA”;
25. Acrescente as variáveis a esta classe:
    1. Nota Pra;
26. Crie métodos GET e SET, para as variáveis que o justifiquem;
27. Crie agora, a classe “FormandoAprendizagem”;
28. Acrescente as variáveis a esta classe:
    1. Nota PAF;
29. Crie métodos GET e SET, para as variáveis que o justifiquem;
30. Crie agora, a classe “FormandoModular”;
31. Crie métodos GET e SET, para as variáveis que o justifiquem;
32. Crie em cada uma das classes anteriores, um método com o nome “calculaNotaFinal”, que calcule a nota final de cada formando, segundo a seguinte regra:
    1. Formandos EFA: a nota é calculada usando 90% da nota das UFCDs, mais 10% da nota de PRA;
    2. Formando Aprendizagem: a nota é calculada usando 70% das UFCDs, mais 30% da nota da PAF;
    3. Formando Modular: a nota é calculada usando 100% da nota das UFCDs;
33. Crie uma classe “Turma” e acrescente as variáveis:
    1. Nome Turma;
    2. Número Turma;
    3. Arraylist de Formandos;
34. Crie métodos para adicionar formandos à turma;
35. Crie uma classe de teste para a classe turma;
36. Preencha a turma com quatro formandos de tipos diferentes (preencha os dados dos formandos);
37. Mostre no ecrã a nota de cada formando;

**Semana 6**

1. Crie um projeto chamado “semana6”;
2. Variáveis e métodos de classe são elementos que estão acessíveis e partilhados por todos os objetos de determinada classe. Copie todos os ficheiros do projeto semana5\_heranca, para o projeto desta semana;
3. Crie uma **variável de classe** chamada contador (lembre-se que esta variável vai ser partilhada por todos as instância da classe formando);
4. Crie um **método de classe** que incremente o valor deste contador e o devolva depois de incrementado;
5. Através do construtor, faça com que o número do formando seja atribuído pelo método de classe criado;
6. Teste agora a turma;

**Semana 7**

1. Crie um projeto chamado “semana7”;
2. Copie todos os ficheiros do projeto “semana6”, para o projeto desta semana;
3. Crie uma classe do tipo “JFrame”;
4. Dentro desta classe, crie uma variável do tipo “Turma”;
5. Coloque nesta classe um “JTextArea” e um botão. Quando acionado o botão abrirá um “JDialog” com o formulário para criação de um novo formando. Quando for criado um novo formando, este deverá ser mostrado na “JTextArea”;
6. Crie agora um “JDialog”, com um formulário para criar novos formandos;
7. Teste para confirmar que tudo funciona corretamente;

**Semana 8**