|  |  |  |  |
| --- | --- | --- | --- |
| **PLANO DE ENSINO** | | | |
| **CURSO** | | **MÓDULO** | **Componente Curricular Sigla** |
| Técnico em Desenvolvimento de Sistemas | | 1º | FPOO |
| **COMPONENTE CURRICULAR** | **AULAS PREVISTAS** | **DOCENTE** | **TURMA(S)** |
| Fundamentos De Programação Orientada A Objeto | 85 | Reenye e Wellington | 1DES |
| **UNIDADE DE COMPETÊNCIA** | **OBJETIVO** | | |
|  | Fundamentos de Programação tem como objetivo proporcionar a aquisição de fundamentos técnicos e científicos relativos a lógica de programação, como ferramenta para desenvolvimento de aplicações, utilizando linguagem orientada a objetos, bem como o desenvolvimento de capacidades sociais, organizativas e metodológicas adequadas a diferentes situações profissionais. | | |

|  |
| --- |
| **FUNDAMENTOS TÉCNICOS E CIENTÍFICOS** |
| 1. Identificar as características e tipos de linguagem de programação  2. Utilizar lógica de programação para a resolução de problemas (7)  3. Declarar as variáveis e as constantes considerando os tipos de dados na elaboração do programa  4. Utilizar comandos de entrada e saída na elaboração de programas (3)  5. Utilizar operações aritméticas, relacionais e lógicas na elaboração de programas (6)  6. Utilizar estruturas de decisão na elaboração do programa (3)  7. Utilizar estruturas de repetição na elaboração do programa (3)  8. Utilizar vetores e matrizes na elaboração do programa (3)  9. Aplicar técnicas de código limpo (clean code) |

|  |
| --- |
| **CAPACIDADES SOCIAIS, ORGANIZATIVAS E METODOLÓGICAS.** |
| 1. Demonstrar atenção a detalhes (44)  2. Demonstrar capacidade de análise (2)  4. Demonstrar raciocínio lógico na organização das informações (24) |

|  |
| --- |
| **CONHECIMENTOS** |
| 1. Linguagem de programação  1.1. Princípios  1.2. Caraterísticas  1.3. Tipos  1.3.1. Compilada  1.3.2. Interpretada  2. Lógica de programação orientada a objeto  2.1. Definição  2.2. Aplicação  2.3. Algoritmo  2.3.1. Entrada, processamento e saída  2.3.2. Variáveis  2.3.3. Constantes  2.3.4. Operadores lógicos  2.3.5. Operadores aritméticos  2.3.6. Operadores relacionais  2.3.7. Estrutura de decisão  2.3.8. Estrutura de repetição  2.3.9. Vetores  2.3.10. Matrizes  2.4. Técnicas de código limpo (clean code) |

|  |  |
| --- | --- |
| **ATIVIDADE** | **SITUAÇÃO DE APRENDIZAGEM - Formativa** |
| **TEXTO** | |
| |  | | --- | | **Obs:** Esta situação de aprendizagem consiste em quatro situações problema com diferentes níveis de dificuldade. Deve ser resolvida em um arquivo único com uma estrutura de menu, onde o usuário selecione o exercício a ser executado (Utilizando Switch/Case); |   **Contextualização**:  Como estamos próximo aos feriados de carnaval em nosso país, exploraremos este tema:   |  | | --- | | Caso **1 –** A escola de samba unidos do JAVA está com dificuldade para gerenciar o pagamento de seus funcionários, devido ao aumento das horas extras. Para auxiliá-los crie um algoritmo que receba o total de horas e horas extras trabalhadas no mês e informe o valor do salário reajus-tado. Leve em consideração que todos recebem em média R$ 5,625 por hora trabalhada e que a hora extra vale 75% a mais do valor original.  Caso **2 –** A academia de artes carnavalescas precisa de um sistema que armazene o nome das es-colas de samba concorrentes e os pontos obtidos nas categorias: bateria, samba-enredo e fan-tasias, e no final da apuração informe o nome da escola, a média dos pontos obtidos e a escola campeã. Sabe-se que 5 escolas de samba estão competindo e que os pontos de cada catego-ria vão de 0 a 10.  Caso **3 –** A escola Rosas de Café está passando por problemas financeiros e precisa da sua ajuda. Atualmente foi verificado que estão gastando além do estimado com as matérias-primas utiliza-das nas fantasias. Antes de tomar qualquer decisão estratégica precisam identificar onde está o maior gasto. Receba do usuário a quantidade utilizada em quilo dos materiais, seu valor e o valor estimado para ser utilizado e informe a relação dos materiais e se o valor está excedendo o valor previsto. Sabe-se que as matérias-primas são: Lantejoulas, Penas de Avestruz do Hi-malaia e tecido de seda Húngara.  Caso **4 –** Para organizar o desfile da escola Imperatriz HTML, será necessário separar as pessoas nos blocos que irão se apresentar. Receba do usuário o total de pessoas que irão participar do desfile e informe a quantidade de pessoas necessária em cada bloco. Sabe-se que cada bloco possui uma porcentagem estimada ficando da seguinte maneira: Comissão de frente (15%), Abre-Alas (5%), Ala das baianas (15%), Carros alegóricos (20%), Mestre-sala e Porta-bandeira (3%), Bateria (20%), Madrinha de bateria (2%) e Recuo da bateria (20%). |   **Desafio**: Propor soluções computacionais que resolvam aos problemas propostos:  1 – Em um arquivo único .java em uma estrutura de menu.  2 – A quinta opção do menu deve sair do laço principal encerrando assim o programa.  3 – Escolha uma das situações propostas e desenhe um fluxograma da sua solução.  Este fluxograma pode ser desenhado no caderno ou em uma aplicação gráfica como MsPaint, draw.io ou outra de sua preferência.  **Entregas**:   * Código fonte com a solução dos quatro problemas. * Fluxograma de uma das situações propostas. | |

**INSTRUMENTO DE REGISTRO DE AVALIAÇÃO FORMATIVA**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Natureza dos Critérios** | **Fundamentos Técnicos e Científicos ou Capacidades Técnicas** | **Critérios de avaliação**   |  |  |  |  | | --- | --- | --- | --- | |  | Crítico |  | Desejável | | 0 | NÃO Atingiu | 1 | Atingiu | | F | Formativa | S | Somativa | | | **Alunos** | | | | | | | |
|  |  |  |  |  |  |  |  |
| **Competências Técnicas** | 1. Identificar as características e tipos de linguagem de programação | Codifica ao menos uma das soluções utilizando a linguagem de programação estudada C ou Java. | F |  |  |  |  |  |  |  |  |
| 2. Utilizar lógica de programação para a resolução de problemas (7) | Desenvolver o algoritmo de forma que resolva os problemas propostos. | F |  |  |  |  |  |  |  |  |
| 3. Declarar as variáveis e as constantes considerando os tipos de dados na elaboração do programa. | Utilizar a classe Scanner de forma que processe efetivamente as entradas de dados através do teclado. | F |  |  |  |  |  |  |  |  |
| 4. Utilizar comandos de entrada e saída na elaboração de programas (3) | Aplicação da sintaxe da linguagem estudada, classes Scanner e System.out.println. | F |  |  |  |  |  |  |  |  |
| 5. Utilizar operações aritméticas, relacionais e lógicas na elaboração de programas (6) | Resolver os problemas que envolvam cálculos. | F |  |  |  |  |  |  |  |  |
| 6. Utilizar estruturas de decisão na elaboração do programa (3) | Resolver os problemas que envolvam decisão. | F |  |  |  |  |  |  |  |  |
| 7. Utilizar estruturas de repetição na elaboração do programa (3) | Resolver os problemas que envolvam laços, for, while ou do while. | F |  |  |  |  |  |  |  |  |
| 9. Aplicar técnicas de código limpo (clean code) | Apresentar o código fonte de maneira adequada a técnica estudada. | F |  |  |  |  |  |  |  |  |
| **Competências de Gestão** | 1. Demonstrar atenção a detalhes (44) | Utilizar endentação e convenções facilitando o entendimento de outros programadores. | F |  |  |  |  |  |  |  |  |
| 2. Demonstrar capacidade de análise (2) | Compreensão dos códigos através de testes de mesa e validação de resultados. | F |  |  |  |  |  |  |  |  |
| 4. Demonstrar raciocínio lógico na organização das informações (24) | Definição de variáveis e constantes pertinentes aos problemas propostos, resolução de problemas. | F |  |  |  |  |  |  |  |  |
| **Nível de Desempenho** | | | |  |  |  |  |  |  |  |  |
| **Nota** | | | |  |  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **PLANO DE AULA** | | |
| **CONHECIMENTOS** | **ESTRATÉGIAS DE ENSINO** | **INTERVENÇÕES MEDIADORAS** |
| 1. Linguagem de programação  1.1. Princípios  1.2. Caraterísticas | Exposição dialogada, trabalho em grupo utilizando a internet. | Quais as principais linguagens de programação utilizadas na atualidade? |
| 1.3. Tipos  1.3.1. Compilada  1.3.2. Interpretada | Apresentação realizada pelos alunos em forma de mini seminário e debate. | Explique a diferença entre uma linguagem compilada e uma interpretada? Em uma aplicação Web qual tipo de linguagem você utilizaria? |
| 2. Lógica de programação orientada a objeto  2.1. Definição  2.2. Aplicação | Exposição dialogada com auxílio de quadro branco e projetor multimídia. | O que é lógica de programação? Qual a diferença entre lógica e sintaxe? |
| 2.3. Algoritmo  2.3.1. Entrada, processamento e saída  2.3.2. Variáveis  2.3.3. Constantes | Exposição dialogada, Atividades práticas, Demonstrações. | Como você definiria algoritmo?  Qual a diferença para um fluxograma? |
| 2.3.4. Operadores lógicos  2.3.5. Operadores aritméticos  2.3.6. Operadores relacionais  2.3.7. Estrutura de decisão  2.3.8. Estrutura de repetição | Demonstrações e Atividades Práticas | Como a computação pode ajudar um comerciante a vender seus produtos? Estes operadores lógicos, aritméticos são aplicáveis a uma solução prática do dia a dia? |
| 2.3.9. Vetores  2.3.10. Matrizes | Exposição dialogada, exemplos de casos de uso, e Atividades Práticas. | Como a memória do computador se comporta diante de várias informações? |
| 2.4. Técnicas de código limpo (clean code). | Demonstrações e Atividades Práticas | Se eu pedisse para outro aluno continuar o código que você iniciou ele conseguiria concluí-lo? |

|  |  |
| --- | --- |
| **ATIVIDADE** | **SITUAÇÃO DE APRENDIZAGEM - Somativa** |
| **TEXTO** | |
| |  | | --- | | **Obs:** Esta situação de aprendizagem consiste em quatro situações problema com diferentes níveis de dificuldade. Deve ser resolvida em um arquivo único com uma estrutura de menu, onde o usuário selecione o exercício a ser executado (Utilizando Switch/Case); |   **Contextualização**:  Como estamos próximo aos feriados de carnaval em nosso país, exploraremos este tema:   |  | | --- | | Caso **1 –** Crie um algoritmo que seja capaz de calcular o valor da multa a ser paga pelo cliente ao atrasar o pagamento da fatura do cartão de crédito. Sabe-se que, a taxa de juros é de 1.3% ao dia. Receba do usuário o valor atual da fatura e os dias de atraso.  Caso **2 –** Desenvolva um sistema capaz de calcular a área e o volume de um cubo.  (Ac = 6 \* a^2) (Vc = a^3).  Caso **3 –** Uma cervejaria precisa de ajuda para gerenciar a entrega de cestas básicas para seus funcionários. Foi pedido para você, criar um projeto piloto que administre 10 funcionários, o sistema deverá possuir um menu com as seguintes opções:   * Registrar Coleta (O usuário irá informar o nome do funcionário e o sistema deverá registrar a retirada da cesta); * Lista das Retiradas (O sistema deverá exibir uma lista com os nomes dos funcionários que já fizeram a retirada); * Lista dos Pendentes (O sistema deverá exibir uma lista com os nomes dos funcionários que ainda não fizeram a retirada); * Sair;   Sabe-se que o nome dos funcionários são: Alice, Miguel, Sophia, Arthur, Helena, Bernardo, Valentina, Heitor, Laura, Davi.  Caso **4 –** Uma pizzaria está criando um sistema de fidelidade, para isso precisa de um sistema que cadastre clientes (apenas nome), contabilize os pedidos feitos e libere um cupom de desconto (20%) a cada 10 pedidos. O sistema deverá armazenar no máximo 20 clientes. |   **Desafio**: Propor soluções computacionais que resolvam os quatro problemas propostos:  1 – Em um arquivo único .java em uma estrutura de menu.  2 – A quinta opção do menu deve sair do laço principal encerrando assim o programa.  3 – Escolha uma das situações propostas e desenhe um fluxograma da sua solução.  Este fluxograma pode ser desenhado no caderno ou em uma aplicação gráfica como MsPaint, draw.io ou outra de sua preferência.  **Entregas**:   * Código fonte com a solução dos quatro problemas. * Fluxograma de uma das situações propostas. | |

**INSTRUMENTO DE REGISTRO DE AVALIAÇÃO SOMATIVA**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Natureza dos Critérios** | **Fundamentos Técnicos e Científicos ou Capacidades Técnicas** | **Critérios de avaliação**   |  |  |  |  | | --- | --- | --- | --- | |  | Crítico |  | Desejável | | 0 | NÃO Atingiu | 1 | Atingiu | | F | Formativa | S | Somativa | | | **Alunos** | | | | | | | |
|  |  |  |  |  |  |  |  |
| **Competências Técnicas** | 1. Identificar as características e tipos de linguagem de programação | Codifica ao menos uma das soluções utilizando a linguagem de programação estudada C ou Java. | S |  |  |  |  |  |  |  |  |
| 2. Utilizar lógica de programação para a resolução de problemas (7) | Desenvolver o algoritmo de forma que resolva os problemas propostos. | S |  |  |  |  |  |  |  |  |
| 3. Declarar as variáveis e as constantes considerando os tipos de dados na elaboração do programa. | Utilizar a classe Scanner de forma que processe efetivamente as entradas de dados através do teclado. | S |  |  |  |  |  |  |  |  |
| 4. Utilizar comandos de entrada e saída na elaboração de programas (3) | Aplicação da sintaxe da linguagem estudada, classes Scanner e System.out.println. | S |  |  |  |  |  |  |  |  |
| 5. Utilizar operações aritméticas, relacionais e lógicas na elaboração de programas (6) | Resolver os problemas que envolvam cálculos. | S |  |  |  |  |  |  |  |  |
| 6. Utilizar estruturas de decisão na elaboração do programa (3) | Resolver os problemas que envolvam decisão. | S |  |  |  |  |  |  |  |  |
| 7. Utilizar estruturas de repetição na elaboração do programa (3) | Resolver os problemas que envolvam laços, for, while ou do while. | S |  |  |  |  |  |  |  |  |
| 9. Aplicar técnicas de código limpo (clean code) | Apresentar o código fonte de maneira adequada a técnica estudada. | S |  |  |  |  |  |  |  |  |
| **Competências de Gestão** | 1. Demonstrar atenção a detalhes (44) | Utilizar endentação e convenções facilitando o entendimento de outros programadores. | S |  |  |  |  |  |  |  |  |
| 2. Demonstrar capacidade de análise (2) | Compreensão dos códigos através de testes de mesa e validação de resultados. | S |  |  |  |  |  |  |  |  |
| 4. Demonstrar raciocínio lógico na organização das informações (24) | Definição de variáveis e constantes pertinentes aos problemas propostos, resolução de problemas. | S |  |  |  |  |  |  |  |  |
| **Nível de Desempenho** | | | |  |  |  |  |  |  |  |  |
| **Nota** | | | |  |  |  |  |  |  |  |  |

Instrumento de avaliação / exercícios / Atividades.

|  |  |  |  |
| --- | --- | --- | --- |
| **Componente curricular:** Fundamentos De Programação Orientada A Objeto | **Atividade:** Listas de Exercícios | | **Nota:** |
| **Unidade:** SENAI Jaguariúna | **Data: / /** | |  |
| **Aluno:** | **N°:** | **Turma:** |
| **Exercícios Vetores**  (1) Crie dois vetores numéricos de dimensão 5 com valores informados pelo usuário. Apresente a soma do primeiro vetor com o inverso do segundo.  (2) Crie um vetor de tamanho 10, permita que o usuário o preencha e informe :  A soma e média dos pares;  A soma e média dos impares;  O total de números pares;  O total de números impares;  (3) Utilizando apenas um vetor de tamanho 6, troque os valores de maneira inversa;  (4) Localize um valor informado pelo usuário dentro de um vetor com dimensão 10;  (5) Faça um programa que leia 10 valores inteiros em qualquer ordem e os classifique em ordem crescente.  (6) Faça um programa que gere 100 valores inteiros aleatorios e os classifique em ordem crescente.  (7) Faça um programa que leia 5 nomes aleatorios e os classifique em ordem decrescente.  (8) Faça um programa que leia 5 números inteiros aleatorios e os classifique em ordem crescente.  **Exercícios Funções**   1. Crie uma função que receba um número decimal e imprima sua representação em binário. 2. Crie uma função que receba um valor em Celsius e retorne em Fahrenheit. 3. Crie uma função que receba um valor em Fahrenheit e retorne em Celsius. 4. Crie uma função que receba dois valores e retorne o maior entre eles. 5. Crie uma função que compare se dois valores são iguais e retorne um booleano. 6. Crie uma função que desenhe um retângulo na tela com caracteres, recebendo como parâmetros (largura, altura e caractere). 7. Crie uma função que calcule o fatorial de um número. | | | |

|  |  |  |
| --- | --- | --- |
| **NÍVEIS DE DESEMPENHO** | **NÍVEIS** | **NOTA** |
| Atingiu todos os critérios críticos e desejáveis | 1 | 100 |
| Atingiu todos os critérios críticos e 4 desejáveis | 2 | 90 |
| Atingiu todos os critérios críticos e 3 desejáveis | 3 | 80 |
| Atingiu todos os critérios críticos e 2 desejáveis | 4 | 65 |
| Atingiu todos os critérios críticos | 5 | 50 |
| Atingiu 5 critérios críticos | 6 | 40 |
| Atingiu 4 critérios críticos | 7 | 30 |
| Atingiu 3 critérios críticos | 8 | 20 |
| Atingiu 1 ou 2 critérios críticos | 9 | 10 |

|  |  |
| --- | --- |
| **NÍVEL MÍNIMO DE DESEMPENHO ESPERADO** | **5** |

|  |  |  |  |
| --- | --- | --- | --- |
| ELABORAÇÃO | DATA | APROVAÇÃO | DATA |
| **Júlio** | **/ /** |  | **/ /** |

***ANEXOS:***

Exercícios ou qualquer material utilizado no dia a dia:

Atividades\_1\_a\_5\_Logica\_de\_Programacao.doc;

CRONOGRAMA (deixar por ultimo) O cronograma deve ser atualizado a cada turma nova.

**Cronograma e Acompanhamento de Distribuição de Aulas**

**Curso**: Técnico em Desenvolvimento de Sistemas

**Componente Curricular:** Fundamentos De Programação Orientada A Objeto

**Turma:** 1DES

**Professor**: Reenye e Wellington 2º Sem. 2023

|  |  |  |
| --- | --- | --- |
| Programa Analítico | Dia/Mês/Ano  (Previsto) | Dia/Mês/Ano  (Realizado) |
| 1. Linguagem de programação: (Princípios, Caraterísticas, Tipos: Compilada, Interpretada) |  |  |
| 2. Lógica de programação orientada a objeto (Definição, Aplicação, Algoritmo) |  |  |
| 2.3.1. Entrada, processamento e saída (Variáveis, Constantes) |  |  |
| 2.3.4. Operadores lógicos |  |  |
| 2.3.5. Operadores aritméticos |  |  |
| 2.3.6. Operadores relacionais |  |  |
| 2.3.7. Estrutura de decisão |  |  |
| 2.3.8. Estrutura de repetição |  |  |
| 2.3.9. Vetores |  |  |
| Atividades com vetores |  |  |
| 2.3.10. Matrizes |  |  |
| Atividades com matrizes |  |  |
| 2.4. Técnicas de código limpo (clean code) |  |  |
| Avaliação Formativa |  |  |
| Correção da Avaliação Formativa |  |  |
| Avaliação Somativa |  |  |
| Atividades de Recuperação |  |  |
| Elaborado por: Reenye e Wellington  Data: 18/07/2023 | | |