

	Delegação Regional do Centro	
	Centro de Emprego e Formação Profissional de Viseu	
	Programador/a de Informática (AÇÃO 24/2023)	
	Formando/a: Judite da Silva Miguel	N.º: 3824316
		Data: 28/11 /2023

Portefólio Reflexivo de Aprendizagens (PRA)

UFCD: 0806: Princípios metodológicos de programação

Número de Horas: 25 horas

Formador/a: António Inácio Mesquita Fonseca

Assinatura: _____

Mediadora: Carla David

Assinatura: _____

Objetivos da UFCD:

- Reconhecer as vantagens e desvantagens da utilização de uma metodologia.

Nesta UFCD, começamos por explorar o site <https://studio.code.org>, onde se exploram os fundamentos de ciência de computação, criando desenhos. A ciência de computação é a base para muito o que se desenvolverá nos próximos anos em termos de programação. Começamos por explorar a Frozen onde fizemos ambientes de gelo e padrões de patinar no gelo. Aqui aprendemos as noções básicas de como programar, através de blocos visuais que arrastamos e soltamos para escrever programas. Um programa é um conjunto de instruções que dizem ao computador o que fazer. Cada bloco é uma ação do que o boneco (Frozen) pode fazer. Nos mesmos moldes trabalhamos formas geométricas, fizemos outros jogos. Também e em contexto dos jogos feitos introduzimos estruturas de repetição, estruturas de decisão.

De seguida fizemos o jogo interativo “Flexbox Froggi”, que é um jogo que ajuda a escrever código CSS.

Posteriormente foram introduzidos conceitos sobre boas práticas de programação, construção de identificadores, lista de identificadores, o que são erros de sintaxe, erros de lógica e erros de semântica, o que é a geração automática de documentação, código anotado ou comentado e layout do programa/projeto. Vimos alguns extratos de código em java escrevemos texto de código e visualizamos alguns dos erros mais comuns que se fazem e corrigimos. Compilamos o código, executamos e removemos os erros do programa. Aprendemos a comentar o código.

Abordámos os paradigmas de programação, a programação procedimental é um paradigma de programação que tem as suas raízes na programação estruturada. Na programação estruturada, os programas são divididos em várias sub-rotinas (e que podem ser de dois tipos: procedimentos e funções). Nos procedimentos um bloco de código nomeado que pode ser chamado para executar uma tarefa específica e nas funções um bloco de código nomeado que pode ser chamado para executar uma tarefa específica e retorna um valor.

A programação orientada para eventos (acontecimentos) é uma abordagem em que o código é escrito para responder a determinados eventos, sejam eles produzidos por um utilizador (clique do rato, por exemplo), sejam obtidos por um sensor (de temperatura, por exemplo). As sub-rotinas que respondem a eventos são designadas por manipuladores de eventos. O manipulador de eventos assegurará que o processamento adequado seja efetuado em resposta ao evento que o desencadeou. Exemplos: aplicações de telemóvel, esperam que aconteça uma determinada coisa, criar interfaces de utilizador, eventos em páginas web.

Nas aulas demos exemplo de uma função (subprograma) `Le()` que está fora do âmbito do programa principal. Um subprograma é composto por um conjunto de comandos que, ao serem executados, realizam uma ou mais tarefas específicas e, no final, devolvem o controlo ao programa (ou a outro subprograma) que despoletou o pedido. Os subprogramas podem ser do tipo implementados na linguagem de programação utilizada (as bibliotecas) ou desenvolvidos e implementados especificamente para um projeto (e eventualmente podem ser usados em outro projeto), esta função está na mesma pasta do programa principal.

Por fim abordámos a programação orientada a objetos. A programação orientada para objetos [*Object-oriented programming -OOP*] é uma metodologia de desenvolvimento de aplicações que se concentra nos objetos e na forma como estes se relacionam. Um objeto é uma qualquer entidade que exista no mundo real e com a qual se pretende implementar num programa de computador. Uma classe é uma especificidade que define todos os objetos que sejam do mesmo tipo.

A herança permite definir uma nova classe que se baseia numa classe principal (também designada por superclasse); a classe descendente (também designada por subclasse) herdará todos os atributos e métodos da classe ascendente, mas alargará a classe descendente adicionando atributos e métodos adicionais ou substituindo os métodos herdados por novo código.

Também abordámos o conceito de polimorfismo que é a capacidade de usar o mesmo método para identificar objetos diferentes.

Por fim fizemos alguns trabalhos onde aplicamos, funções, procedimentos, classe principal, subclasses, heranças, polimorfismo.

Gostei da maneira como o formador pensou e estruturou a ação de formação, gostei dos pontos que abordou, acho que conciliou muito bem a teórica com a prática. Acho que foi uma boa introdução ao que se vai abordar futuramente.