**Linguagem de Programação Python**

**Capítulo 2 – Tópico 2: Estruturas de Repetição**

**Estruturas de Repetição**

As estruturas de repetição são necessárias, úteis e importantes no desenvolvimento de rotinas de programação. Neste tópico, vamos apresentar as condições que permitem a realização de rotinas repetitivas e fornecem mais sofisticação e inteligência ao seu código.

Em certos momentos, surgem circunstâncias específicas em programação que exigem que uma ação ou um grupo de ações seja realizado várias vezes. Para resolver esses casos de forma lógica foram criadas as chamadas “Estruturas de Repetição”, que permitem criar blocos de repetição até que determinada condição se torne verdadeira ou por um número predeterminado de vezes.

Diagrama

Descrição gerada automaticamente

Quem está se preparando para entrar na universidade, deve estudar até passar no vestibular, não é mesmo? Com isso, podemos dizer que os estudantes repetem ações (estudar bastante; prestar vestibular) até que uma condição (conquistou uma vaga?) seja conquistada. Note que, no fluxograma, o bloco de ações “Estudar bastante” e “Prestar vestibulares” deve repetir-se por tempo indeterminado, até que a condição “Conquistar uma vaga” se torne verdadeira. Este exemplo simples permite perceber que a repetição é algo que faz parte da vida cotidiana. Tanto é assim que ela é parte crucial em praticamente todos os processos produtivos.

Que tal pensarmos nos processos de repetição no contexto da informática?

Com o advento do computador, passou a ser possível automatizar a produção de artefatos que até então eram feitos manualmente. Com os avanços tecnológicos do século XXI, isso tornou-se cada vez mais frequente. De modo geral, transformar algo em automático é prever a repetição de eventos durante determinado número de vezes, até que uma condição seja alcançada ou até que o usuário decida encerrar o processo. Por isso, a repetição é algo muito presente na programação de computadores.

Como as repetições são programadas no computador?

Para resolver situações envolvendo repetição de forma lógica e inteligente, foram criadas as chamadas estruturas de repetição ou laços de repetição. As estruturas de repetição permitem programar o loop (laço, ciclo) de blocos de instrução a partir de parâmetros previamente estabelecidos em algoritmos ou códigos de programação.

Texto

Descrição gerada automaticamente

Para o uso correto de estruturas de repetição, um item fundamental que deve ser compreendido é o controle do número de vezes que a estrutura será repetida. Desse modo, as repetições podem ser controladas a partir de alguns recursos.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Para construir estruturas de repetição, podemos utilizar dois tipos de recursos diferentes.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente Uma imagem contendo Diagrama

Descrição gerada automaticamente

**Estrutura ENQUANTO...FAÇA (WHILE)**

A instrução ENQUANTO...FAÇA (WHILE) indica o início de um loop e recebe uma condição como parâmetro. Tal condição é chamada de condição de parada, já que o loop é encerrado quando a condição retornar um valor lógico falso ou o loop nem é iniciado se a verificação condicional resultar em um valor lógico falso assim que a estrutura de repetição for iniciada.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Nessa estrutura, a rotina de repetição só é executada caso a condição seja verdadeira. Se a condição for falsa, a estrutura é ignorada e finalizada.

*Exemplo:*

Acompanhe três condições diferentes em que a instrução ENQUANTO...FAÇA está representada.

Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente



Interface gráfica do usuário

Descrição gerada automaticamente



**Estrutura PARA PRÓXIMO (FOR NEXT)**

A estrutura PARA PRÓXIMO tem uma estrutura que lhe é particular.

Interface gráfica do usuário, Diagrama

Descrição gerada automaticamente

Nesse tipo de construção, a rotina sempre entra na estrutura na primeira vez. Como temos a atribuição VAR = vi (valor inicial) no início da estrutura, tudo o que estiver dentro dela se repetirá até encontrar a palavra PRÓXIMO.

Quando encontra a palavra PRÓXIMO, a rotina soma o valor do passo (somar 2) à variável de controle (A) e realiza um teste com o valor final (vf). Enquanto o valor da variável de controle não atender a condição determinada para o valor final (vf), a rotina continuará se repetindo.

**Exemplos:**

*Exemplo 1 – contador fixo*

Vamos construir um algoritmo em pseudocódigo para ler o nome e duas notas de cada aluno de uma turma de 15 alunos. Além disso, precisaremos calcular a média, bem como mostrar o valor das duas notas e a média de cada aluno.

Interface gráfica do usuário

Descrição gerada automaticamente

*Exemplo 2 – flag de resposta*

Vamos construir um algoritmo em pseudocódigo para ler o nome e duas notas de diversos alunos. Além disso, é preciso calcular a média, bem como mostrar o nome, as duas notas e a média de cada aluno. Após processar cada aluno, deve-se perguntar se o usuário deseja continuar.

Quando o usuário responder “sim” (S), o programa continuará. Quando a resposta for “não” (N), o programa terminará.

Texto

Descrição gerada automaticamente

*Exemplo 3 – flag predeterminado*

Vamos construir um algoritmo em pseudocódigo para ler o nome e duas notas de uma turma de alunos. Além disso, precisaremos calcular a média, bem como mostrar o nome, as duas notas e a média de cada aluno.

O programa vai finalizar quando o usuário digitar a palavra FIM como conteúdo da variável Nome, em vez de um nome válido. Desse modo, predeterminamos que o programa irá terminar quando o usuário digitar a palavra FIM no lugar do nome de um aluno.

Texto

Descrição gerada automaticamente