



Programação de Computadores e seus conceitos básicos
Materiais de Formação
Subcapítulo 1.5. – Programas e Linguagens de Programação

WP3: Materiais de Formação Code4SP

Preparado por:



CITIZENS
IN POWER



Center for Social
Innovation



ZAUG
gGmbH



Subcapítulo 1.5.: Programas e Linguagens de Programação



Programas e Linguagens de Programação

As linguagens de programação são usadas para criar programas, que são usados para controlar o comportamento de uma máquina, tipicamente um computador. Uma linguagem de programação fornece uma estrutura para o programador dar instruções à máquina, e uma forma de comunicar essas instruções a outros programadores. Existem muitas linguagens de programação em uso atualmente. As mais populares são C, Java, Python, e JavaScript.





Programs and Program Languages

Tipos de linguagens

- Existem dezenas de linguagens de programação em uso hoje em dia, podendo ser sumariamente classificadas em cinco categorias:
- **Linguagens de programação de baixo nível:** Estas linguagens de programação são muito próximas do *hardware* e são utilizadas para programar microprocessadores e outros dispositivos de baixo nível. Não são fáceis de aprender e não são populares para a programação de uso geral. Exemplos: Linguagem de montagem, linguagem de programação C, e linguagem de montagem de baixo nível.
- **Linguagens de programação de alto nível:** Estas linguagens de programação foram concebidas para serem fáceis de aprender e utilizar. São populares para a programação de uso geral. Exemplos: Java, C++ e Python.
- **Linguagens de *scripting*:** são concebidos para serem fáceis de usar e são populares para de *scripting*. Exemplos: Python, Ruby e JavaScript.
- **Línguas específicas do domínio:** são concebidas para uma tarefa ou indústria específica. Não são fáceis de aprender e não são populares para programação de uso geral. Exemplos: MATLAB, SQL, and FORTRAN.
- **Linguagens de programação orientadas a objetos:** são baseadas no paradigma da programação orientada para objectos. Exemplos: Java, C++, and Python.





Programs and Program Languages

Do programa de alto nível ao ficheiro executável

Quando um programa de computador é escrito numa língua de alto nível, é primeiramente traduzido para uma língua de nível inferior, que é mais facilmente compreendida por máquinas. A língua de nível inferior é então compilada num ficheiro executável, que pode ser executado num computador.





Programs and Program Languages

IDEs (*Integrated Development Environments*)

Um Ambiente Integrado de Desenvolvimento (IDE) é uma aplicação de software que proporciona instalações abrangentes aos programadores de computadores para o desenvolvimento de software. Um IDE consiste tipicamente num editor de código fonte, ferramentas de automação de construção e um depurador. O editor de código fonte permite ao programador escrever o código, enquanto as ferramentas de automatização da construção automatizam o processo de compilação desse código numa forma que o computador pode executar. O depurador permite ao programador percorrer o código, examinando o estado do programa em cada ponto da sua execução. As IDEs são muitas vezes utilizadas em conjunto com um sistema de controlo de versões, o que permite a diferentes programadores que trabalham no mesmo projeto partilhar e fundir as suas alterações sem problemas.





Programs and Program Languages

Os elementos mais comuns em linguagens de programação

As linguagens de programação informática partilham uma série de elementos comuns, apesar das suas diferenças. Todas as linguagens de programação têm uma forma de representar instruções para o computador de uma forma que o computador possa compreender. Isto é normalmente chamado código, ou código fonte. Os programadores utilizam o código para criar programas e aplicações de *software*. Todas as linguagens de programação têm também uma forma de organizar instruções para que possam ser reutilizadas, modificadas ou partilhadas com outros programadores. A isto chamamos normalmente “biblioteca” ou “módulo”. As bibliotecas e os módulos permitem aos programadores criar programas complexos, baseando-se no trabalho de outros programadores. Por fim, todas as linguagens de programação têm uma forma de transmitir informação ao utilizador sobre o que o programa está a fazer e como está a funcionar. Isto é normalmente chamado de informação de saída ou *debug*. A informação de saída e *debug* ajuda os programadores a compreender e corrigir problemas dos seus programas.



Programs and Program Languages

Programação processual e orientada a objetos

Há dois tipos principais de programação: processual e orientada para objetos. A programação processual envolve um processo passo-a-passo, enquanto a programação orientada aos objetos envolve a criação de objetos que interagem uns com os outros. A programação processual é muitas vezes vista como mais simples do que a programação orientada a objetos. É fácil aprender os passos necessários para completar uma tarefa, e é fácil alterar a ordem desses passos sem afetar o resultado. Contudo, a programação de procedimentos pode ser menos eficiente porque pode ser difícil reutilizar o código que foi escrito para uma tarefa específica. A programação orientada a objetos é mais complexa que a programação processual, mas permite maior flexibilidade e reutilização do código. Os objetos podem ser criados para tarefas específicas e depois reutilizados conforme necessário. Além disso, o código orientado para objetos é muitas vezes mais fácil de ler e compreender do que o código processual. Contudo, a programação orientada aos objetos pode ser mais difícil de aprender e pode ser menos eficiente do que a programação processual.



OBRIGADO!

Co-funded by the
Erasmus+ Programme
of the European Union

