
CES 22 – Programação Orientada a Objetos - POO

Aula 1

Introdução

- ▶ Motivação
- ▶ Linguagem Python
- ▶ Avaliação



Motivação para POO

- ▶ Desenvolvimento de programas é uma atividade complexa e arriscada.
- ▶ Riscos:
 - ▶ Programa com defeitos.
 - ▶ Projetos frequentemente estouram prazos e orçamentos.
 - ▶ Modificações nos programas.

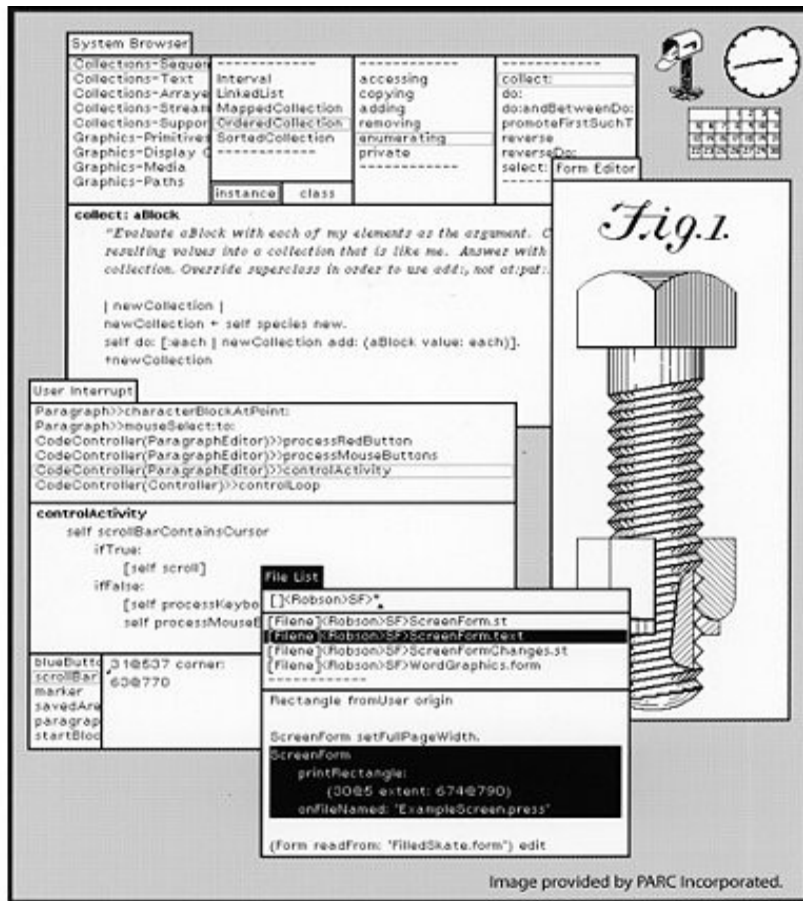


Origens da POO

- ▶ Linguagem Simula 67 (Norwegian Computing Cente - Ole Johan Dahl e Kristen Nygaard) foi a primeira linguagem de programação (focada para simulação) a apresentar os conceitos de objetos e classes de objetos.
- ▶ Simula influencia Alan Kay a desenvolver a linguagem Smalltalk no Xerox PARC nos anos 70.
- ▶ Smalltalk foi a primeira linguagem a ter um ambiente programação gráfico com Janelas, Icones e Mouse.
- ▶ Smalltalk continua ativo: Squeak, Pharo, Amber, ...



Smalltalk



Cont.

- ▶ Várias linguagens adotam hoje a POO:
 - ▶ ADA
 - ▶ C++
 - ▶ Java
 - ▶ C#
 - ▶ Python
 - ▶ Etc...



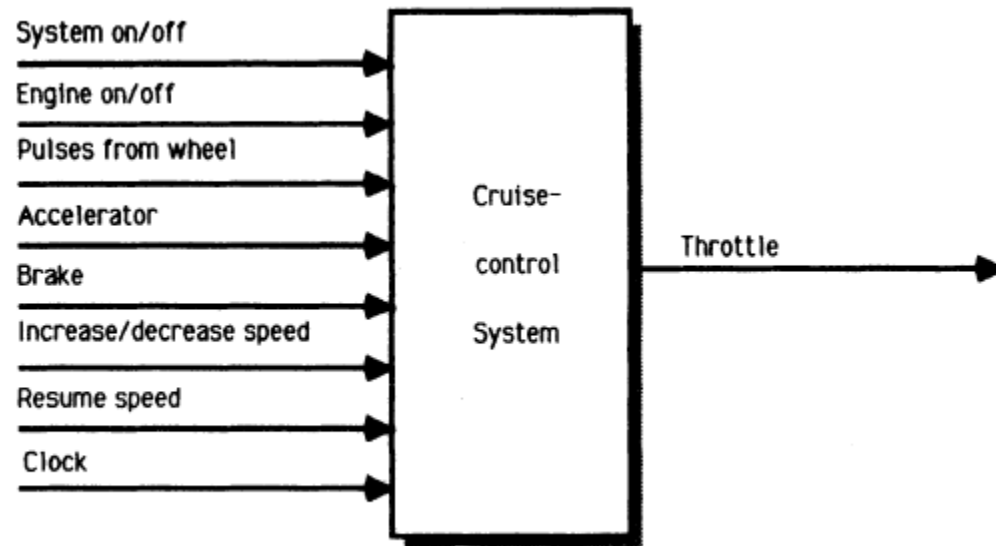


Fig. 1. Cruise-control system hardware block diagram.

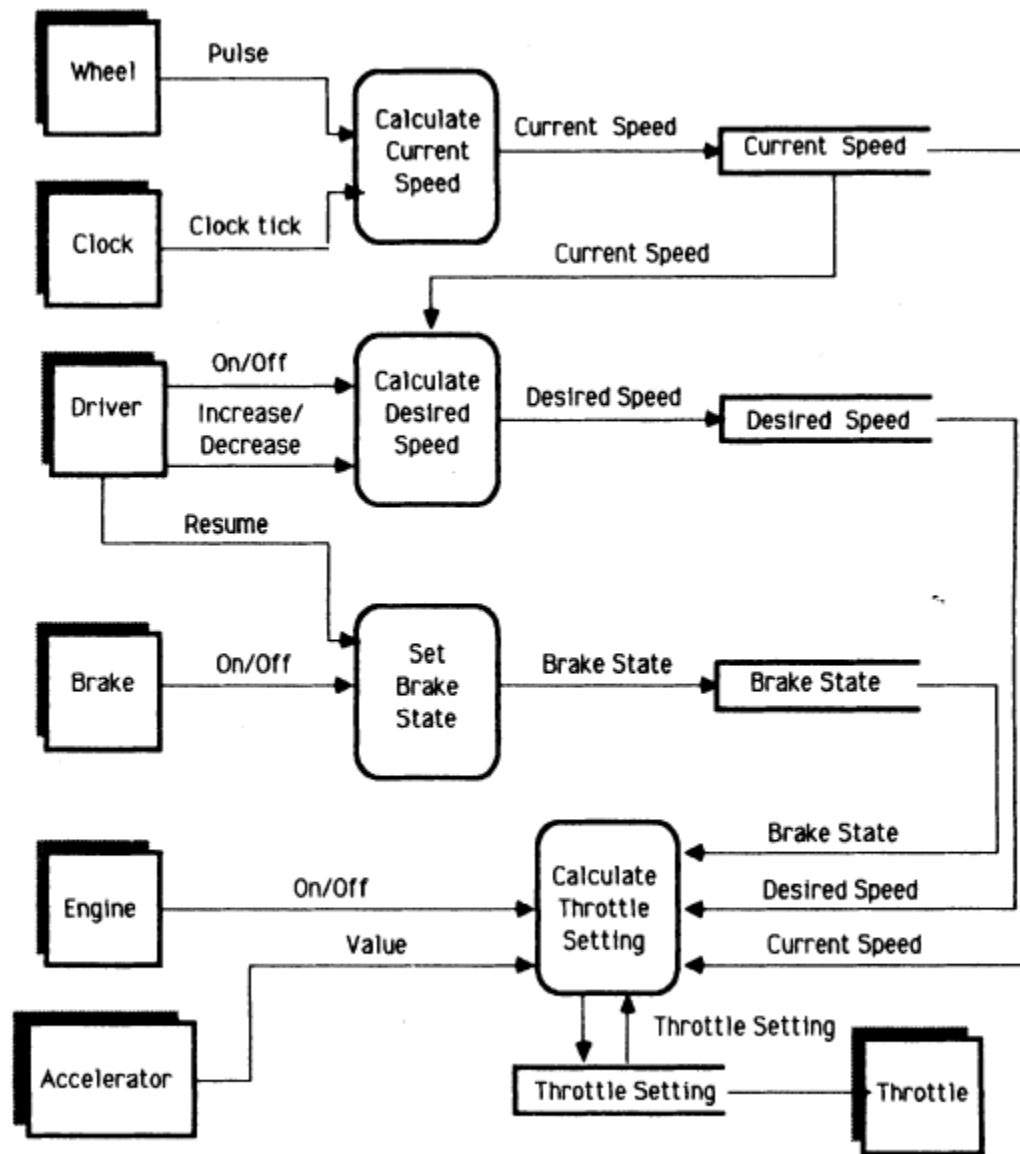


Fig. 2. Cruise-control system data flow diagram.

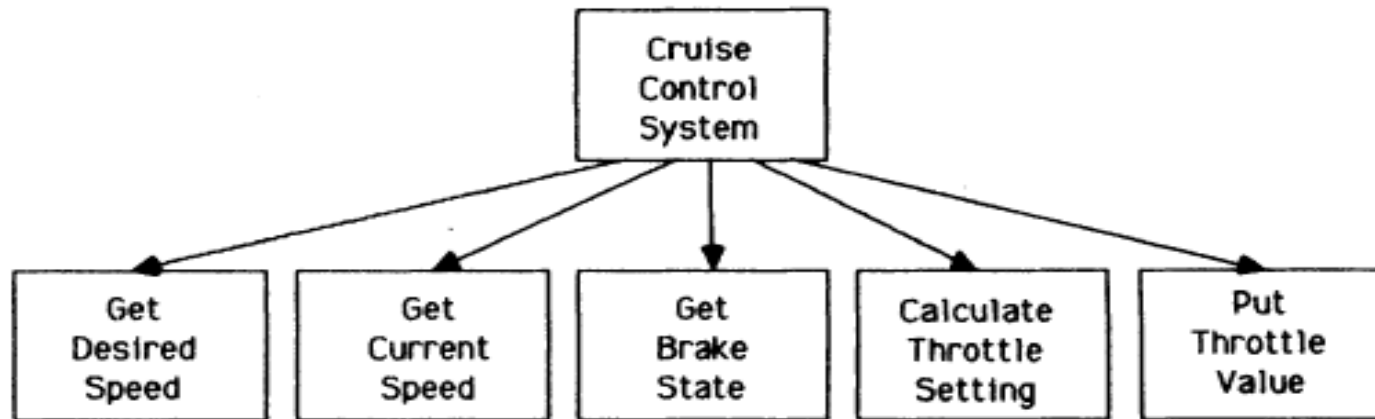


Fig. 3. Functional decomposition.

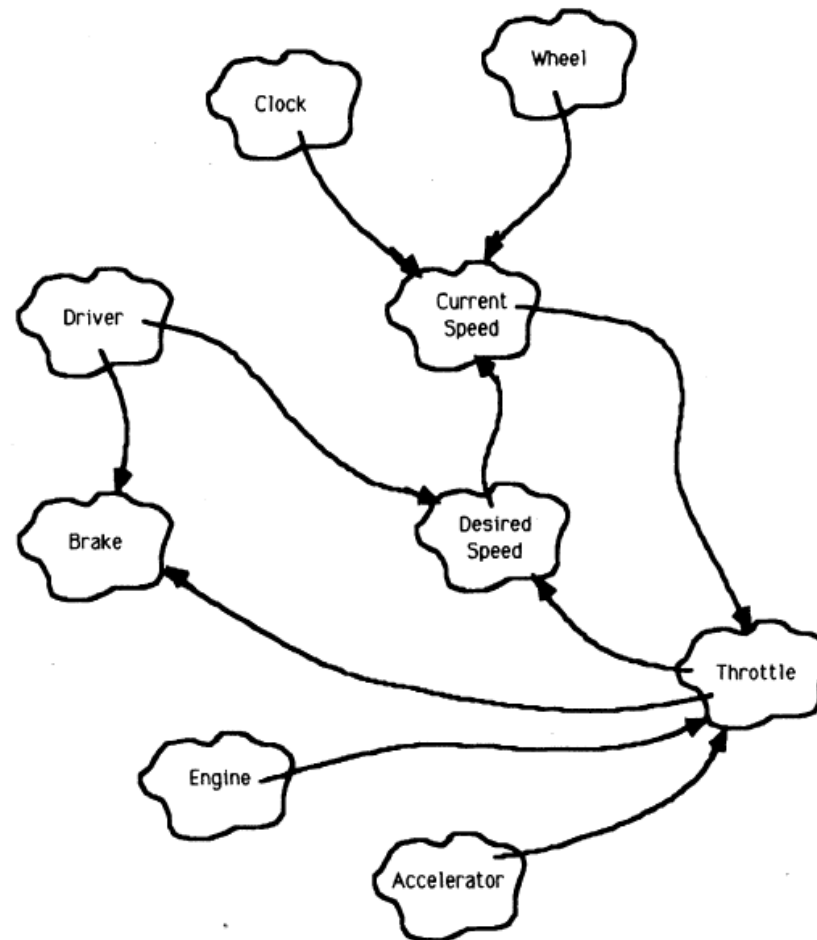


Fig. 4. Object-oriented decomposition.

Vantagens do modelo Orientado a Objetos

- ▶ O modelo OO é mais próximo do mundo real. A decomposição funcional (procedimental) desmembra o problema em sub-rotinas desconexas da realidade.
- ▶ O modelo OO facilita as modificações. No modelo funcional as variáveis que descrevem o problema são transformadas em variáveis globais e qualquer modificação tende a afetar todos os módulos envolvidos. Exemplo: Se for necessário introduzir um display de velocidades, esta modificação afetará apenas os objetos associados a velocidade.



- A Orientação a Objetos facilita também o desenvolvimento de sistemas.

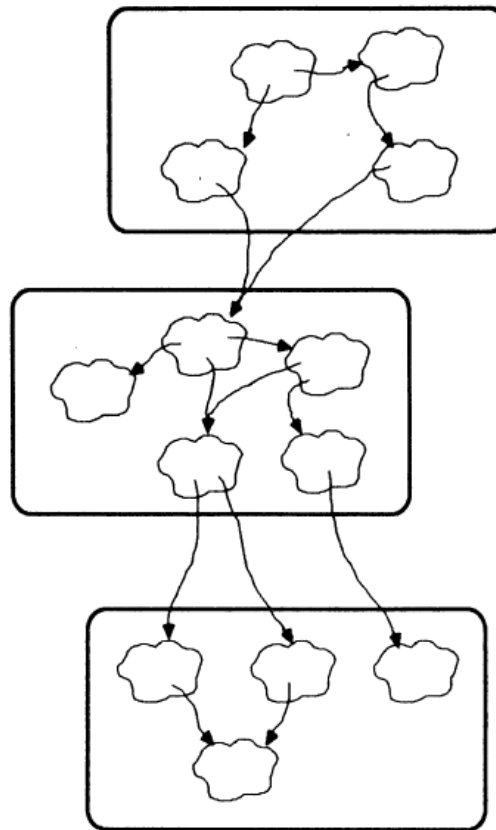


Fig. 5. Canonical structure of large object-oriented systems.

Happy End?



Limitações

- ▶ Os problemas reais são complexos e as construções de linguagens OO não são suficientes para abordar adequadamente todas as nuances.
- ▶ Exemplo: o conceito de herança é hoje usado com restrições.



- ▶ **Eric Raymond** (2005)

The Art of UNIX Programming

"The OO design concept initially proved valuable in the design of graphics systems, graphical user interfaces, and certain kinds of simulation. To the surprise and gradual disillusionment of many, it has proven difficult to demonstrate significant benefits of OO outside those areas."

- ▶ **Linus Torvalds** (2007)

this email

"C++ is a horrible language. ... C++ leads to really, really bad design choices. ... In other words, the only way to do good, efficient, and system-level and portable C++ ends up to limit yourself to all the things that are basically available in C. And limiting your project to C means that people don't screw that up, and also means that you get a lot of programmers that do actually understand low-level issues and don't screw things up with any idiotic 'object model' crap."

- ▶ **Eric Allman** (2011)

Programming Isn't Fun Any More

"I used to be enamored of object-oriented programming. I'm now finding myself leaning toward believing that it is a plot designed to destroy joy. The methodology looks clean and elegant at first, but when you actually get into real programs they rapidly turn into horrid messes."



Quem não tem um cão, caça com um gato!



Python

MONTY PYTHON'S FLYING CIRCUS



Histórico

- ▶ Criado por Guido Van Rossum em 1991. Um programador holandês que trabalhou na Google entre 2005 e 2012 desenvolvendo a linguagem Python e atualmente está na Dropbox.



Características

- ▶ Linguagem Interpretada por uma máquina virtual com coletor de lixo.
- ▶ Fácil leitura e sintaxe concisa.
- ▶ Suporta Múltiplos paradigmas.
- ▶ Free e Open Source
- ▶ Portavel para diferentes plataformas.



Usos

- ▶ Análise de dados.
- ▶ Programação de sistemas.
- ▶ Programação para a Internet e Web.



Python 2 ou 3?

- ▶ A versão 3 possui diferenças significativas frente a versão 2. As modificações foram introduzidas de modo a prover maior consistência à linguagem.
- ▶ Principais diferenças:
 - ▶ Comando print
 - ▶ Comportamento de tratamento de exceções.
 - ▶ Adoção do código Unicode (Permite 107 mil caracteres).
- ▶ No curso usaremos a versão 3. (3.6)
- ▶ Entretanto, vários ambientes requerem o uso da versão 2. Exemplo: Google AppEngine.



Literatura recomendada

1. How to think like a Computer Scientist
 1. <http://openbookproject.net/thinkcs/python/english3e/index.html>
2. Livros gratuitos:
 - ▶ <http://pythonbooks.revolunet.com/>



Avaliação

- ▶ Avaliação bimestral
 - ▶ Projetos:
 - ▶ Projeto de grupo (até 3 alunos). 45%
 - ▶ Exercícios. 45%
 - ▶ Colaboração do aluno. 10%
- ▶ Final do semestre
 - ▶ Exame teórico
 - ▶



Cont.

- ▶ Projeto de Grupo
 - ▶ Tema 1.o Bimestre
 - ▶ Desenvolvimento de game usando PyGame.
 - ▶ Tema 2.o Bimestre
 - ▶ Aplicação usando a Web.
 - ▶ Avaliação:
 - ▶ Apresentações.
 - ▶ Criatividade
 - ▶ Complexidade
 - ▶ Uso adequado da POO.
 - ▶ Documentação no Github.
 - ▶ Envolvimento de todos os membros registrado no Github.



Grupo do curso

<https://groups.google.com/forum/#!forum/ita-ces-22-2018>

- ▣ Alunos devem se inscrever para postar mensagens.
- ▣ Os posts (relevantes) serão considerados para avaliação da colaboração.



Primeiro Programa

- ▶ Hello World
- ▶ `print("Hello World")`



Identação para marcar blocos

```
if True:
```

```
    print("true")
```

```
else:
```

```
    print("false")
```



Todos os valores são objetos

- ▶ Exemplos de valores
 - ▶ 54
 - ▶ 43.3
 - ▶ “abc”
- ▶ Todos os valores possuem um tipo ou classe
 - ▶ `type(54)`
 - ▶ `<class 'int'>`



Variáveis

- ▶ Variáveis são referencias para objetos.
- ▶ `message = "What's up, Doc?"`
- ▶ `n = 17`
- ▶ `pi = 3.14159`
- ▶ `n = message`



Turtle

import turtle

Allows us to use turtles

`wn = turtle.Screen()`

Creates a playground for turtles

`alex = turtle.Turtle()`

Create a turtle, assign to alex

`alex.forward(50)`

Tell alex to move forward by 50 units

`alex.left(90)`

Tell alex to turn by 90

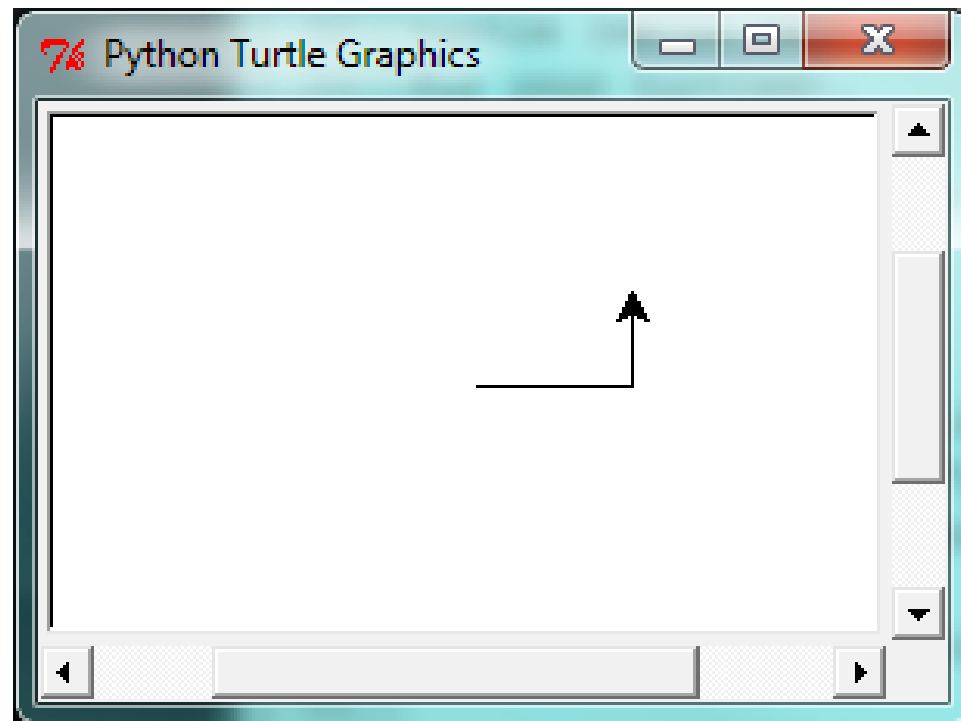
degrees `alex.forward(30)`

Complete the second side of a rectangle

`wn.mainloop()`

Wait for user to close window





Histórico

- ▶ Turtle vem da linguagem Logo desenvolvida por Seymour Pappert e Daniel Bobrow no MIT em 1967.
- ▶ A ideia é ser um ambiente visual para o desenvolvimento de aplicações educacionais.



Exercício 1

- ▶ Explorar turtle.
- ▶ Fazer os exercícios 6, 11, 12 e 13 do livro 1.

