Orientada a Objetos

Encapsulamento

Profa.: Márcia Sampaio Lima

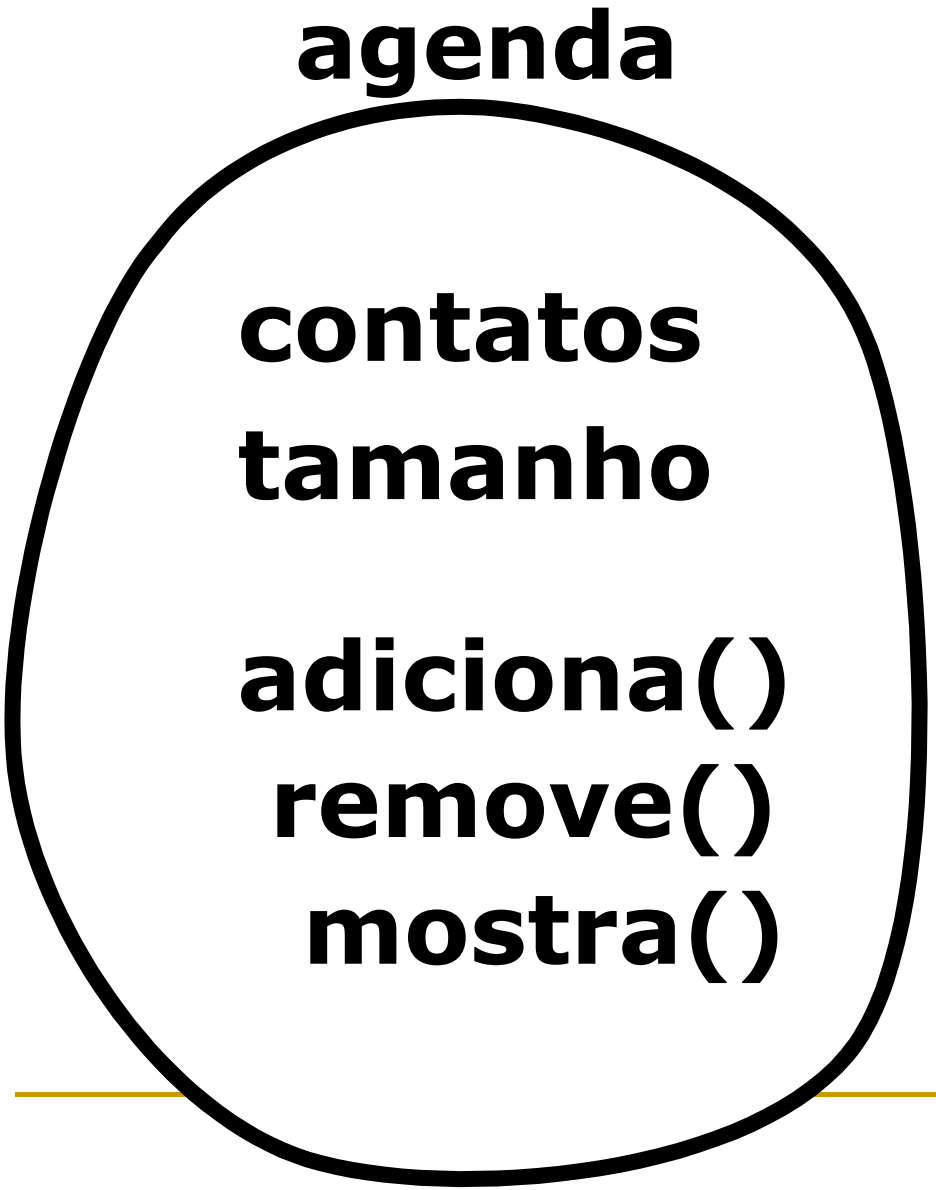
EST - UEA

Ref.: Slides Prof. Flávio José Mendes Coelho, M.Sc.

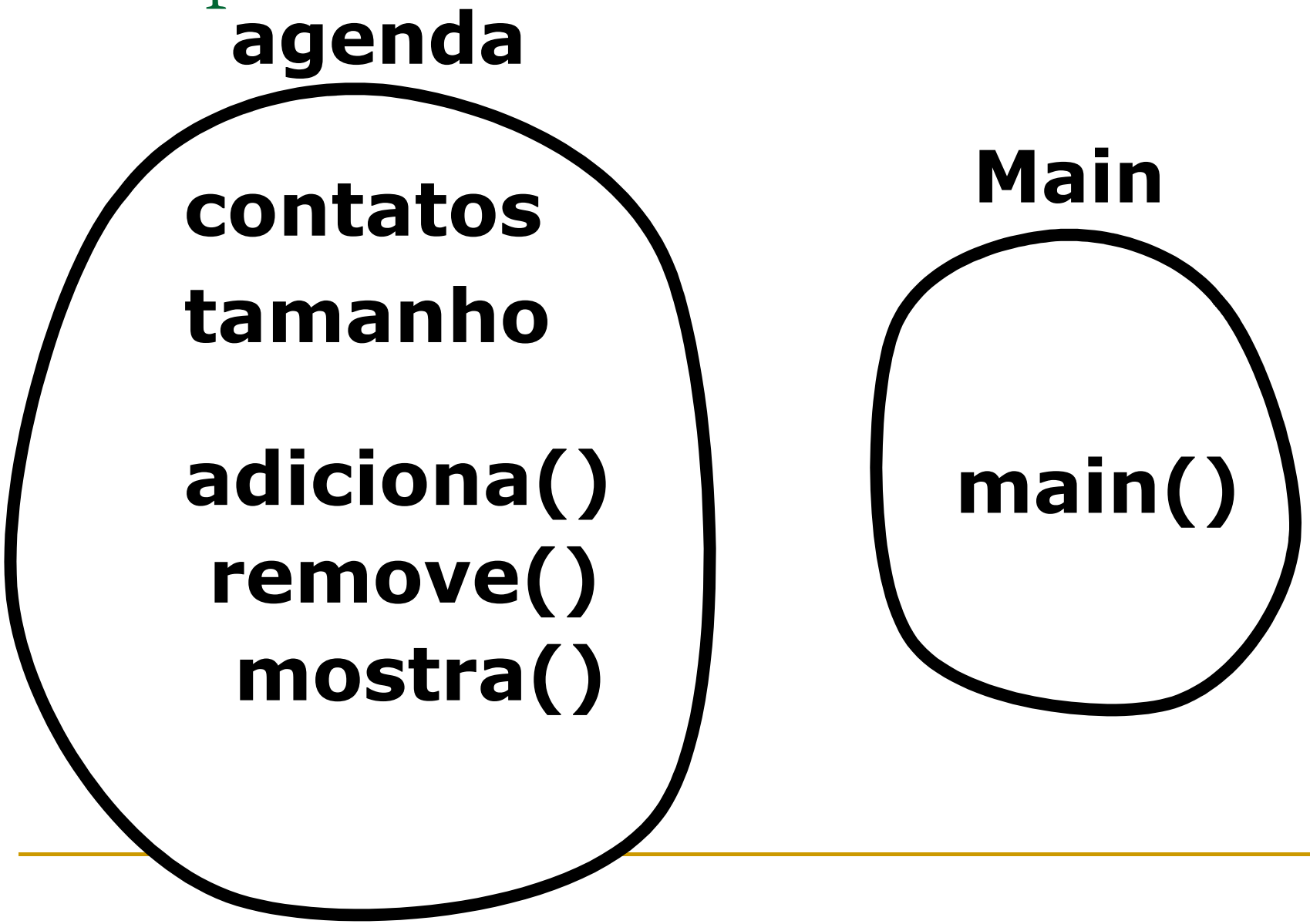
Encapsulamento

- **Estudo de caso: Agenda**
- **Vejamos graficamente como a classe Main acessa a classe Agenda.**

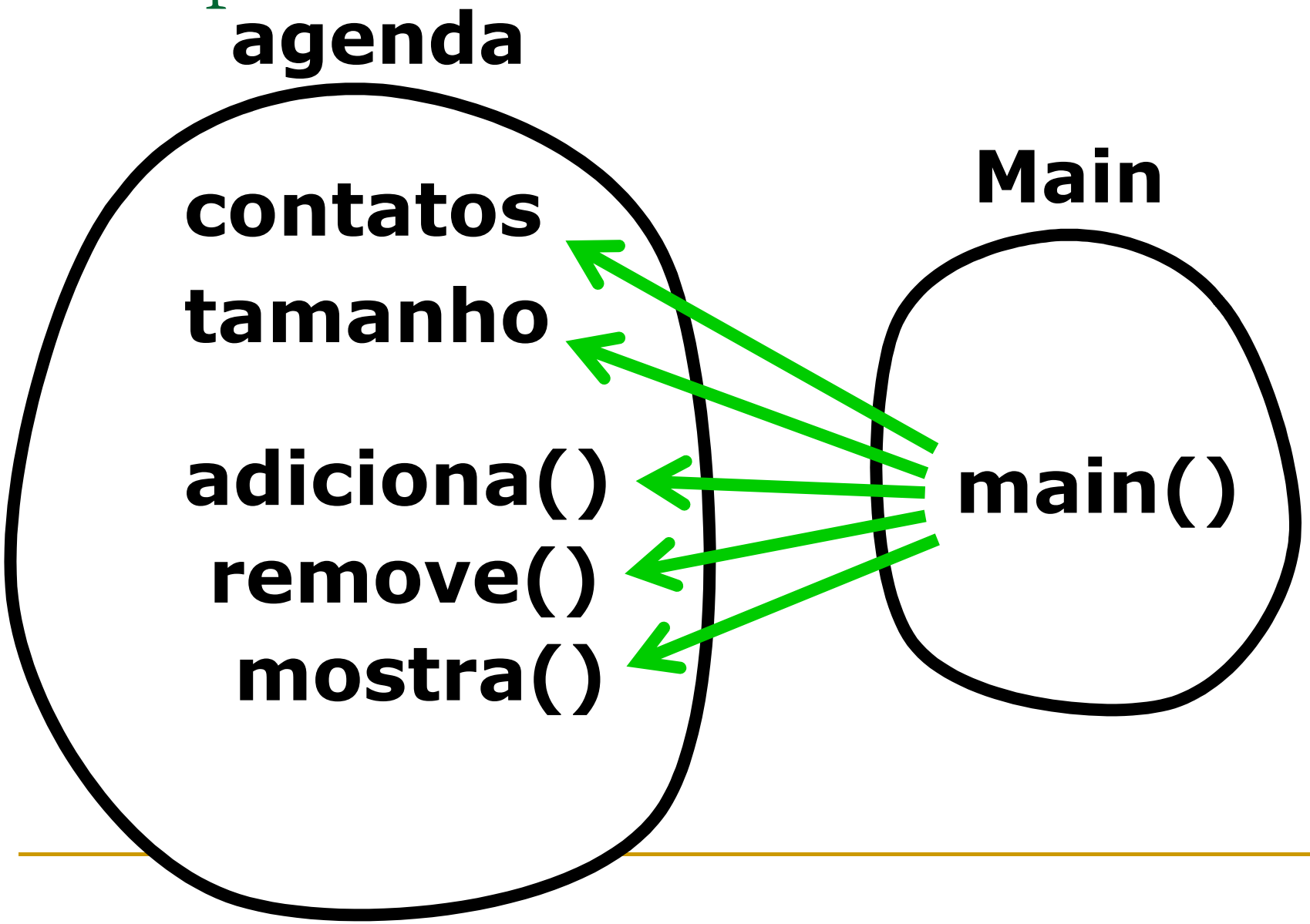
Encapsulamento



Encapsulamento



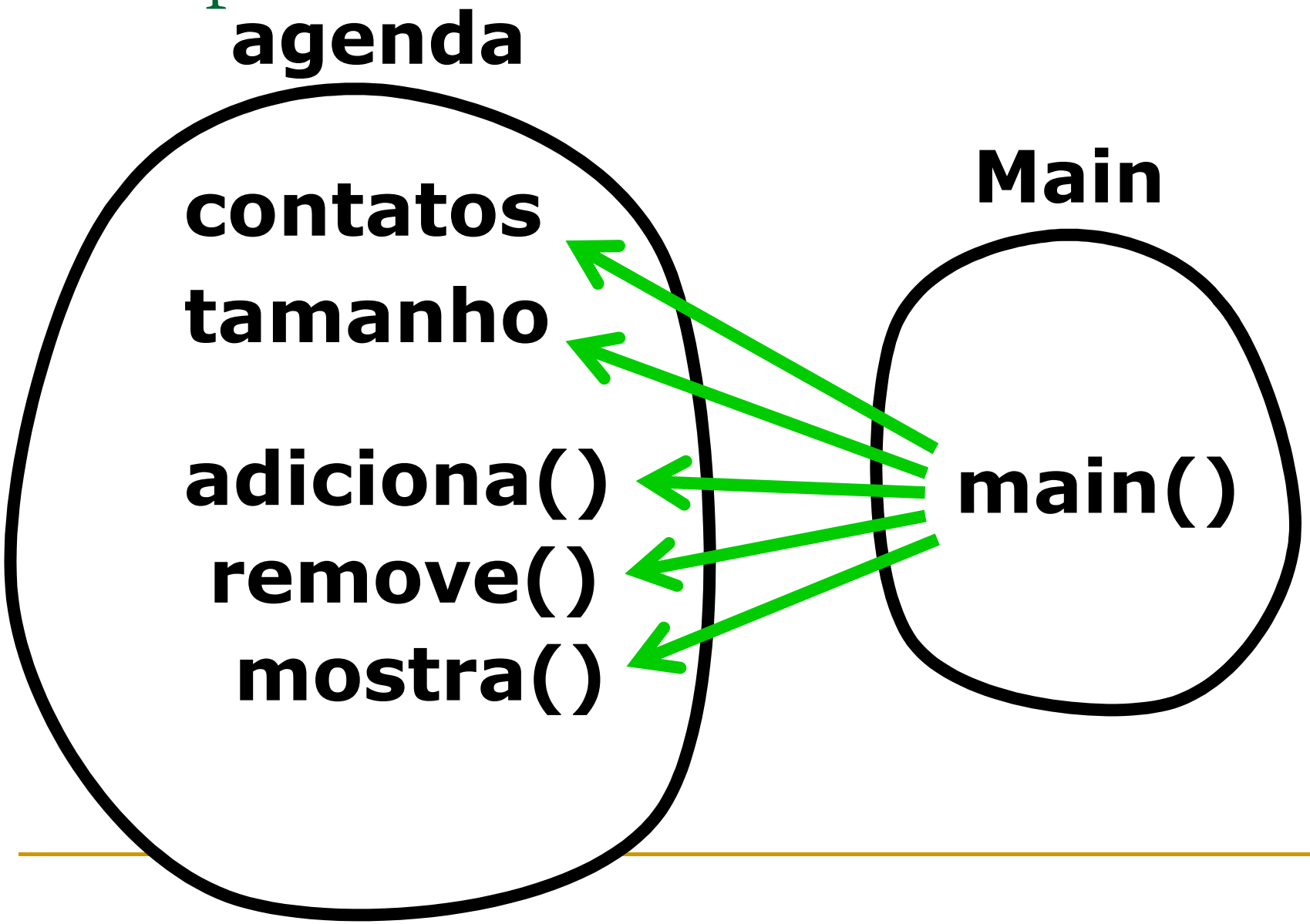
Encapsulamento



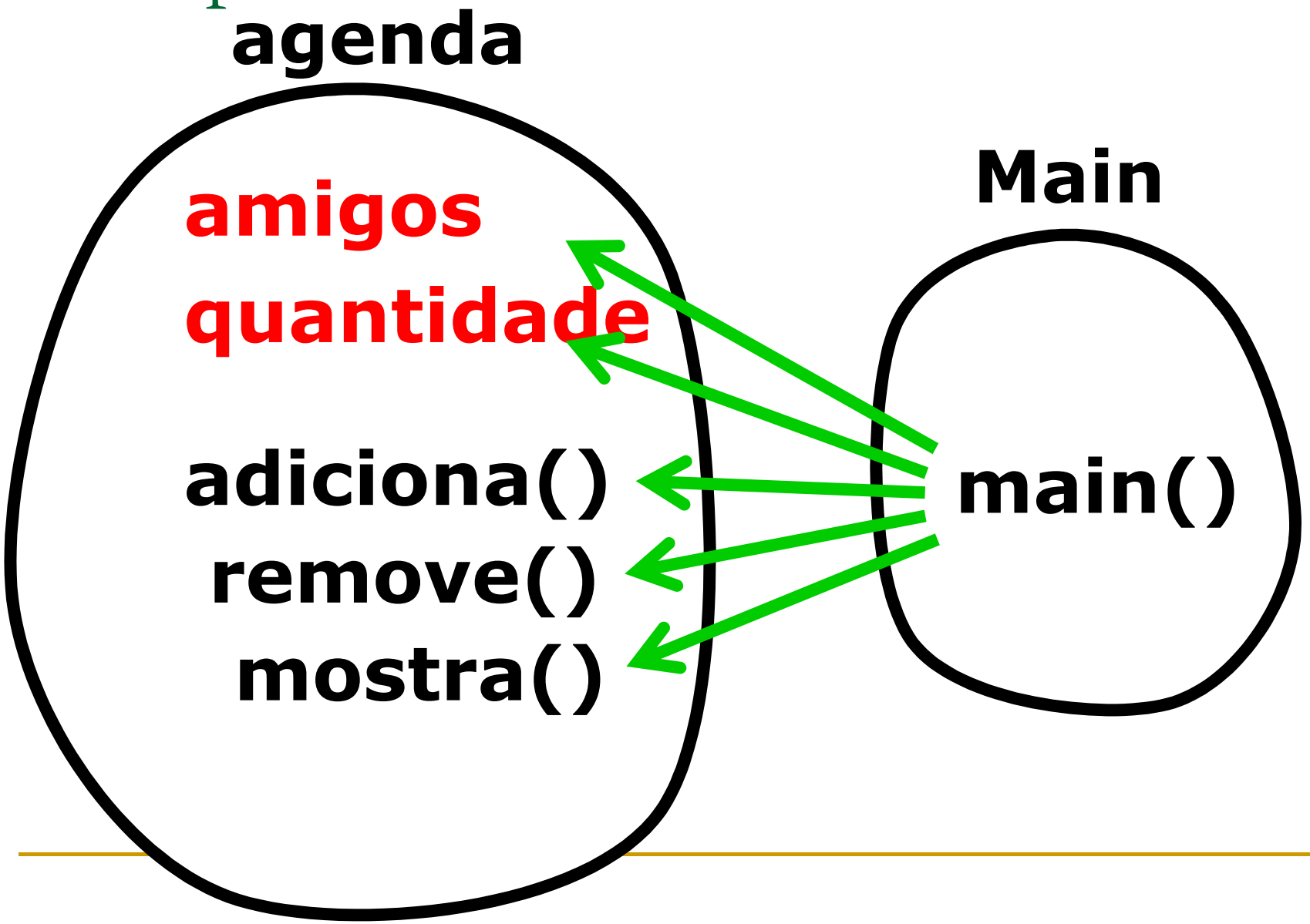
Encapsulamento

- Agenda sem Encapsulamento:
  - **Main tem acesso total aos membros de Agenda.**
  - Causando:
    - **Dependência de implementação.**
    - **Se atributos de Agenda mudam, Main muda!**

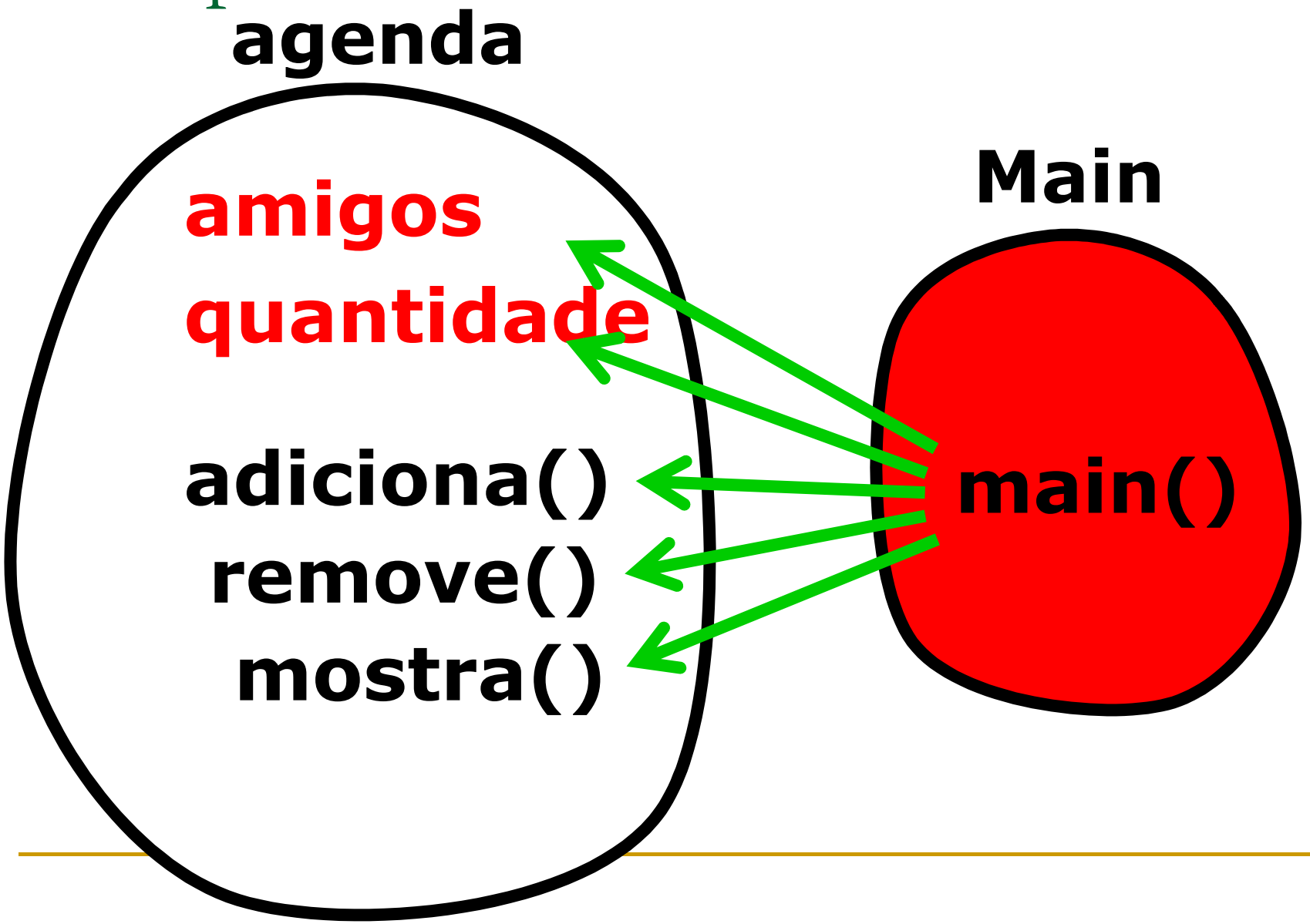
Encapsulamento



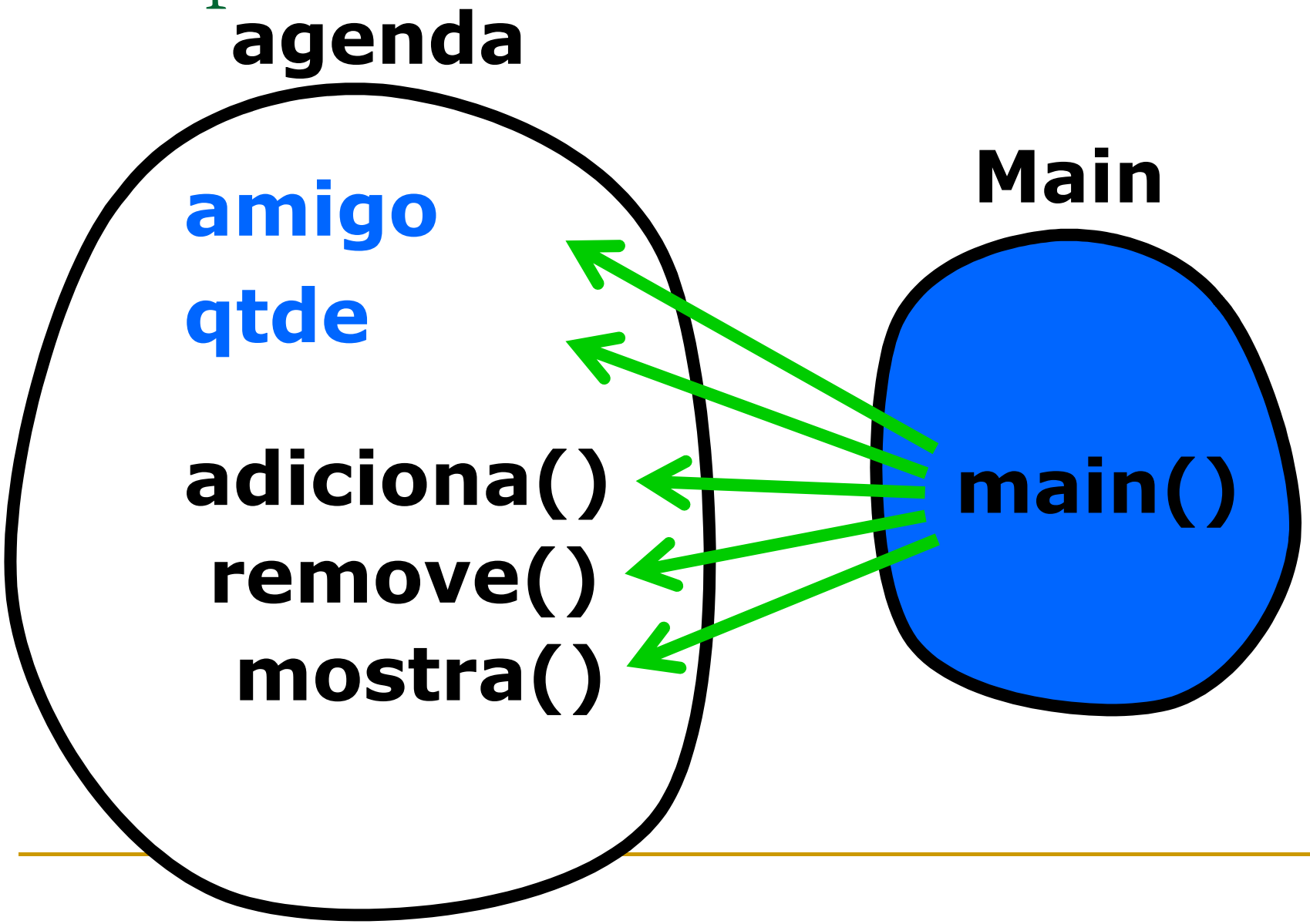
Encapsulamento



Encapsulamento



Encapsulamento

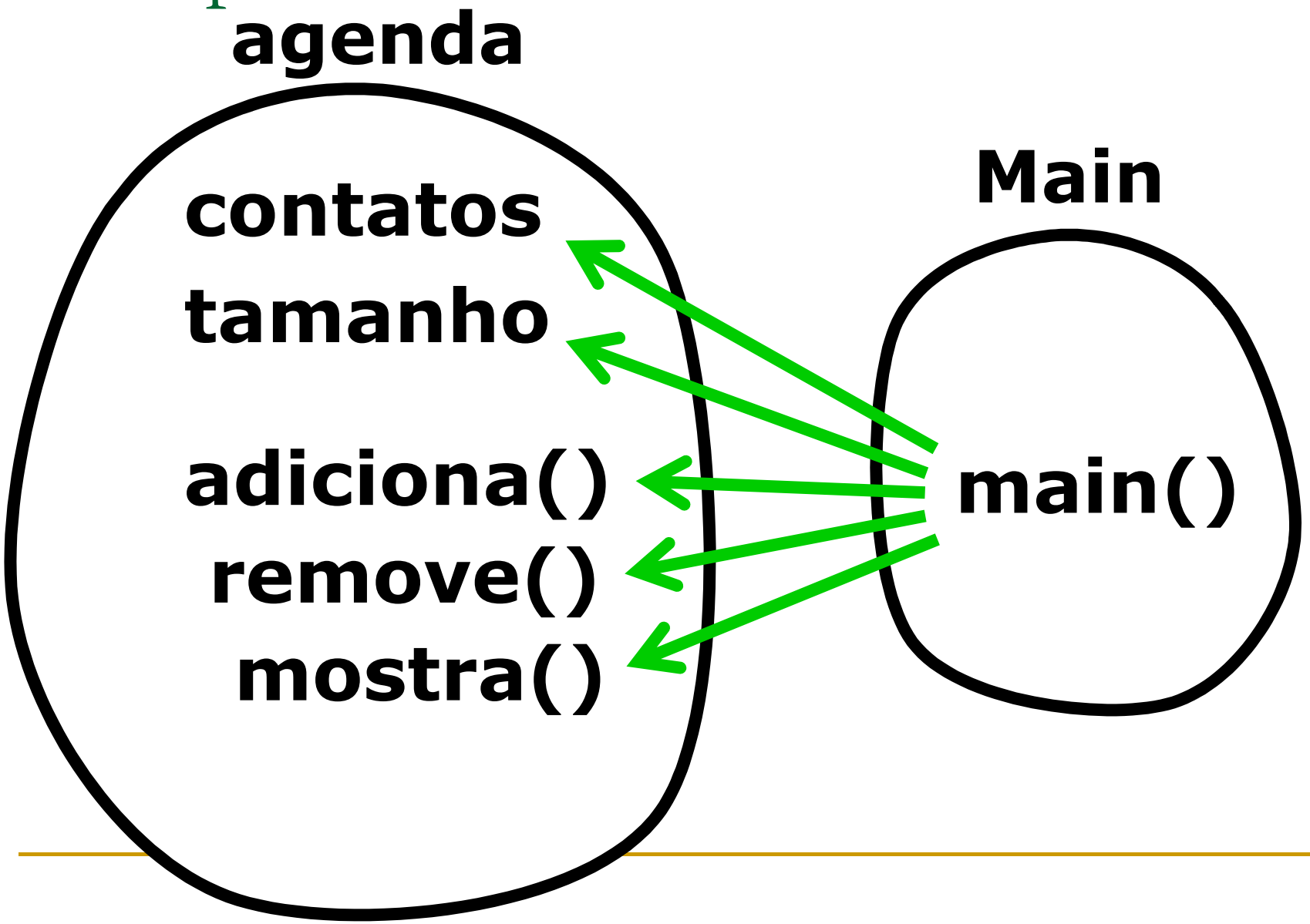


Encapsulamento

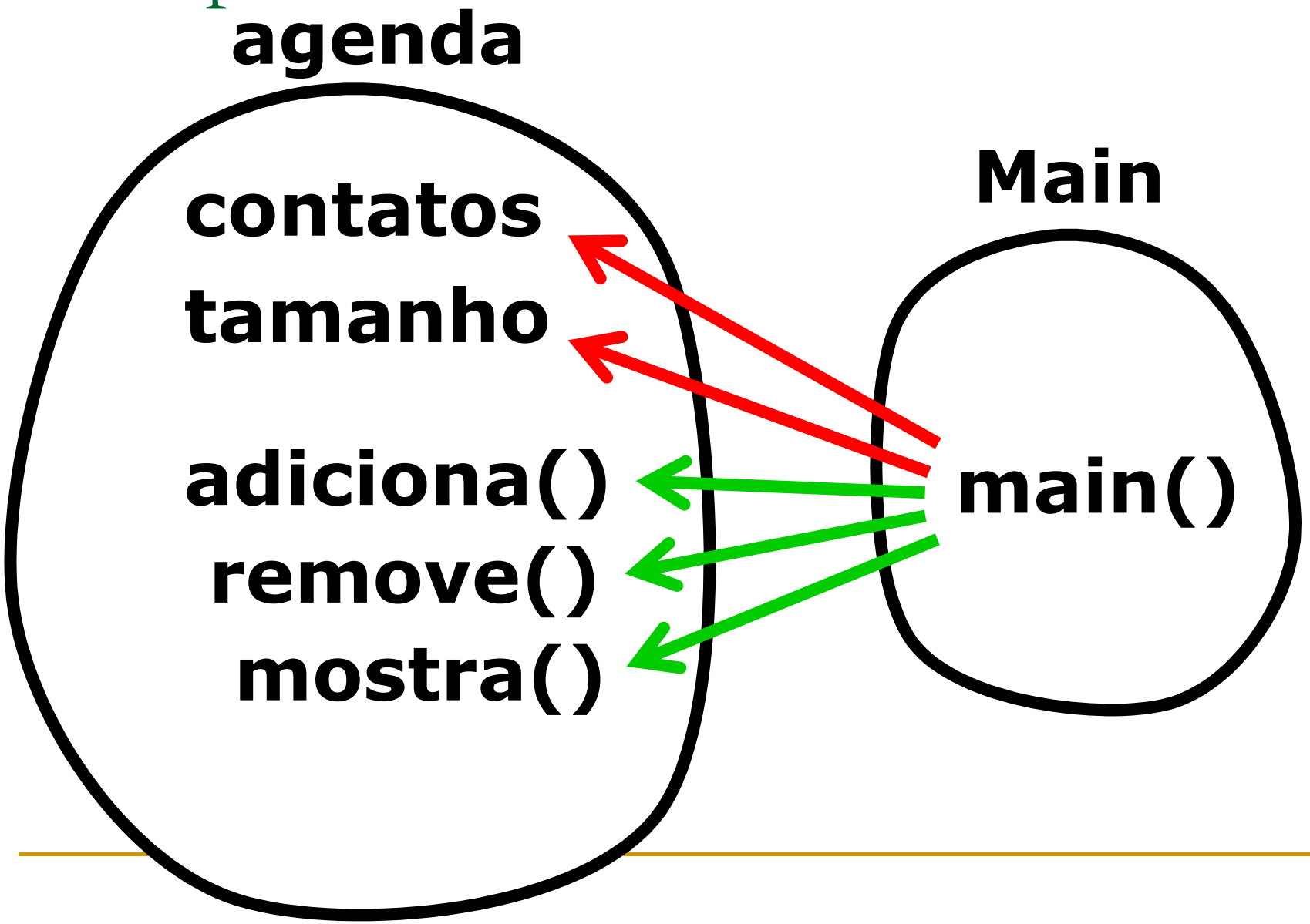
- Agenda sem encapsulamento:

- **Como resolver o problema da dependência de implementação?**
- **Encapsulando a classe Agenda!**

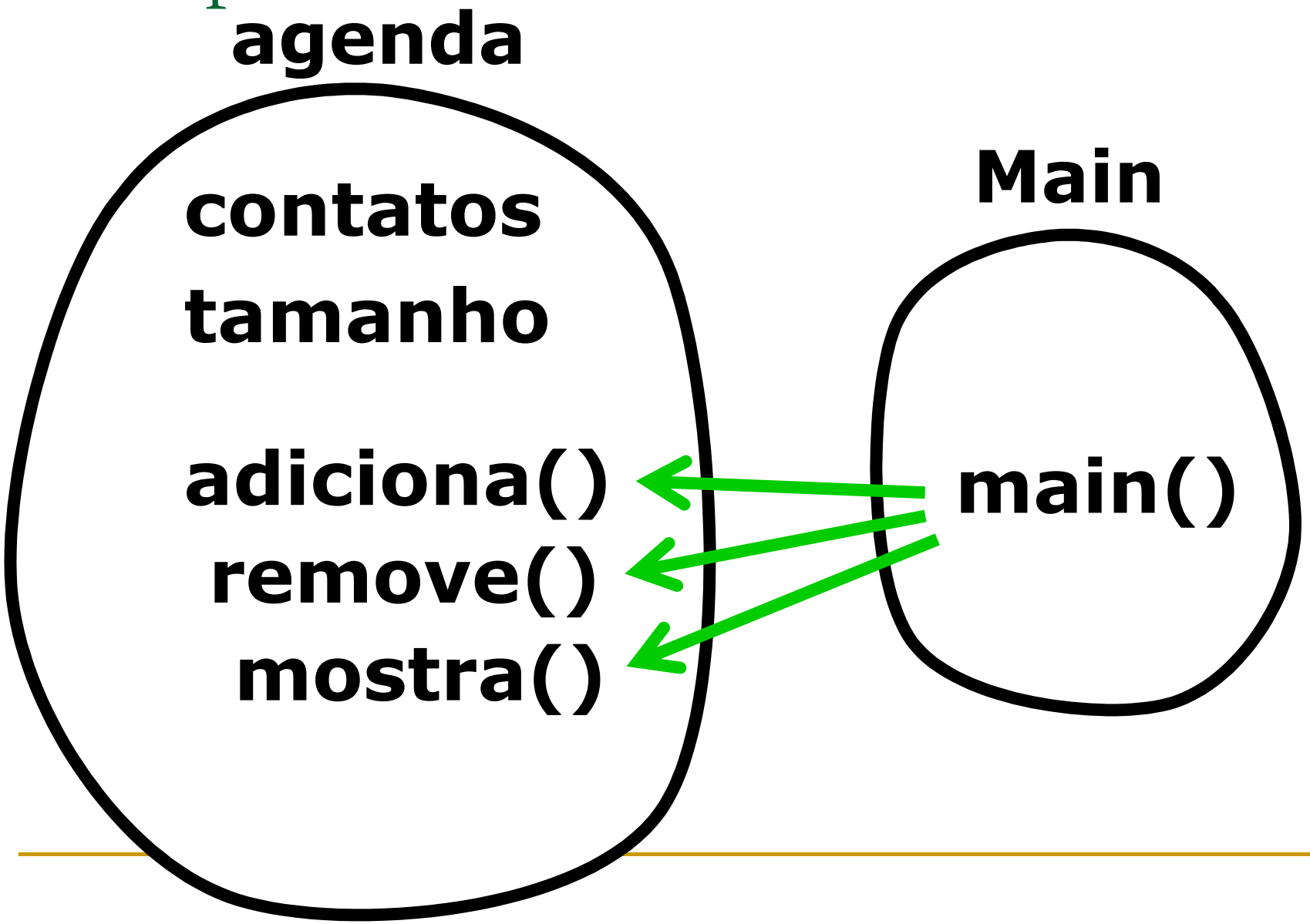
Encapsulamento



Encapsulamento



Encapsulamento



Encapsulamento

- Os métodos:
  - Adiciona();
  - Remove();
  - Mostra();

Passam a ser os únicos membros visíveis por outras classes!!!

Encapsulamento

- Os métodos:
  - Adiciona();
  - Remove();
  - Mostra();

Formam a interface da classe Agenda!!!

Encapsulamento

- **Interface:**
  - **É o conjunto dos métodos visíveis (acessíveis) de uma classe.**

- **Classe Encapsulada:**
  - **Uma classe encapsulada só pode ser utilizada por meio de sua interface.**

Encapsulamento

- **Encapsular uma classe é:**
  - **Ocultar os atributos, e**
  - **Fornecer uma interface para outras classes.**

Encapsulamento

- **Como encapsular uma classe?**
  - **Como ocultar os atributos e métodos?**
  - **Como fornecer uma interface para outras classes?**

    - **Por meio dos modificadores de visibilidade public e private.**

Encapsulamento

- Modificadores de visibilidade:
- **Public**
  - **Publica o membro.**
  - **O membro publicado poderá ser acessado por qualquer método de qualquer classe.**

Encapsulamento

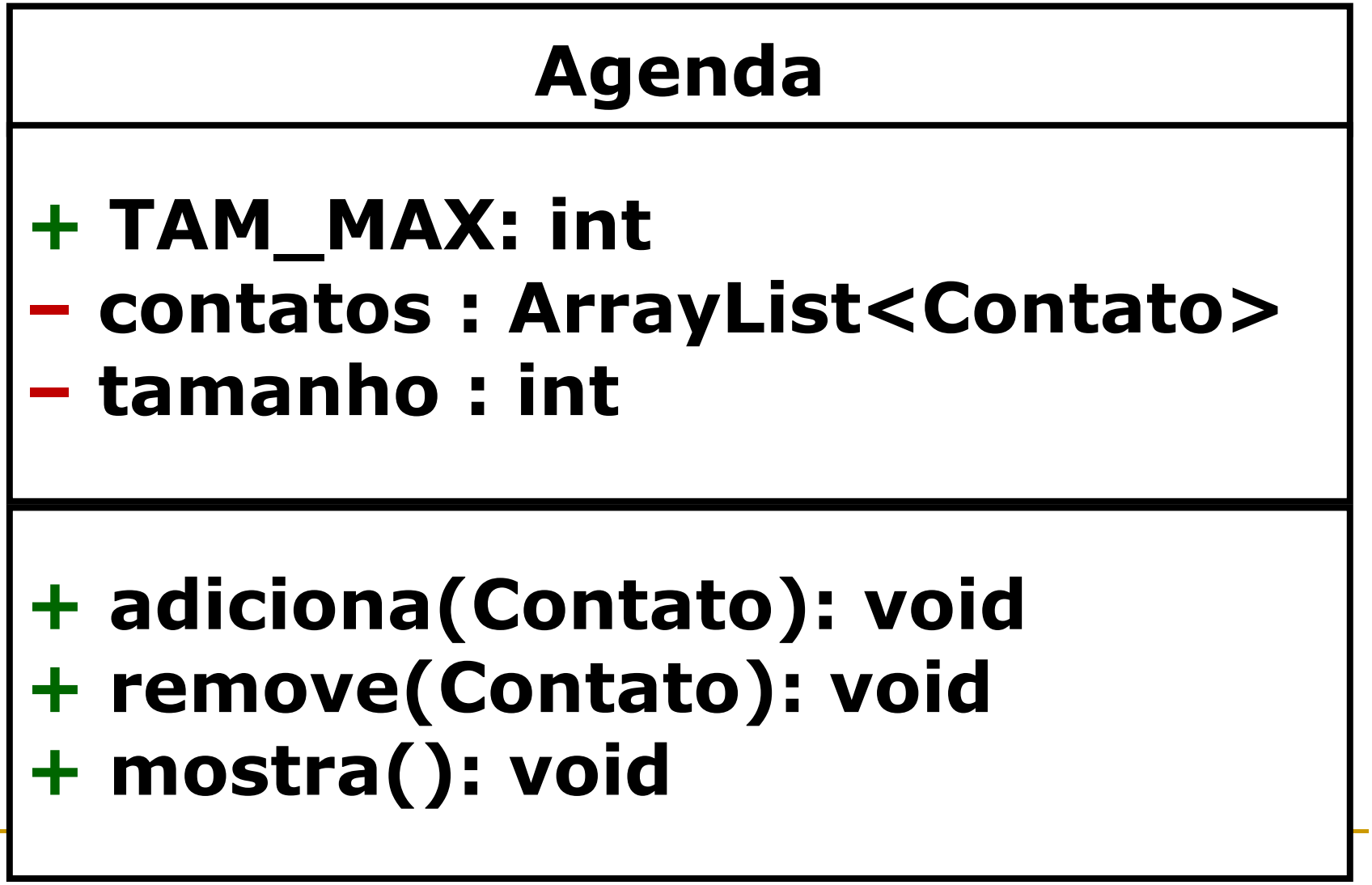
- Modificadores de visibilidade:
- **Private**
  - **Oculta o membro.**
  - **O membro privado poderá ser acessado por métodos da própria classe.**

Encapsulamento

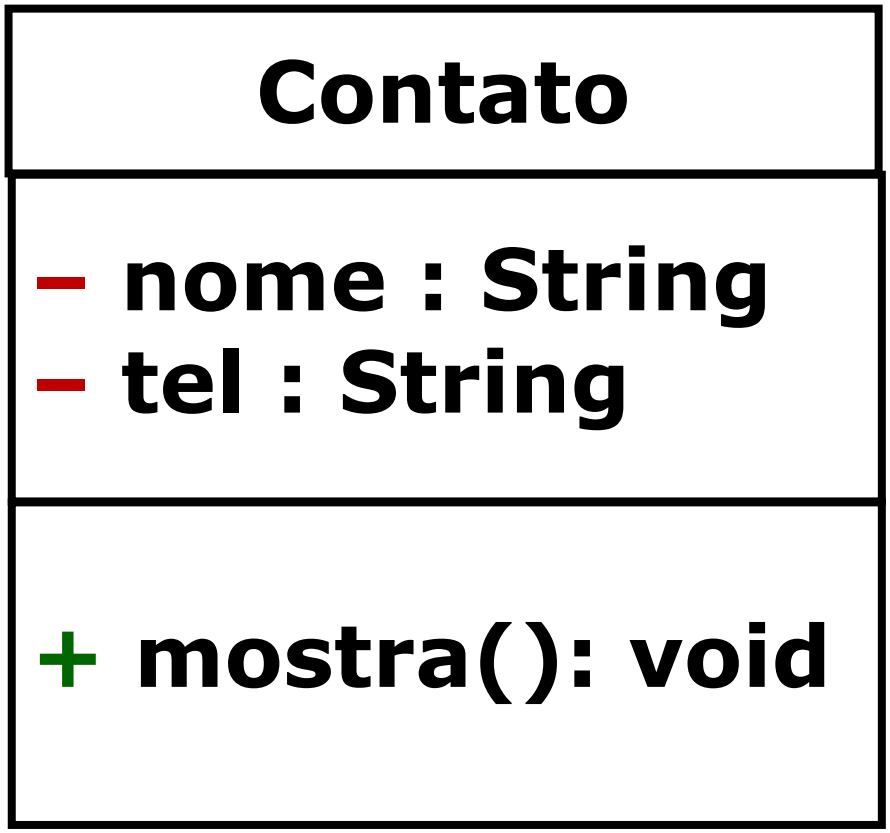
- Representação Java e UML:

Visibilidade	Java	UML
Pública	**public**	**+**
Privada	**private**	**–**

Encapsulamento



Encapsulamento



Encapsulamento

- **Depois do encapsulamento Main só sabe utilizar a classe Agenda por meio da interface de Agenda.**

Encapsulamento

- **Seja a classe encapsulada E e a classe X que utiliza E.**
  - **Somente os próprios métodos de E podem acessar seus atributos.**
- **Seja a classe encapsulada E e a classe X que utiliza E.**
  - **A classe X só sabe utilizar E por meio da interface de E.**

Ref.: Slides Prof. Flávio José Mendes Coelho, M.Sc.

Encapsulamento

- **Seja a classe encapsulada E e a classe X que utiliza E.**
  - **Se E for modificada, mas sua interface não, então estas mudanças não afetam X.**

Encapsulamento

- **Acoplamento:**

  mede o nível de dependência existente entre duas classes.

  - ✓ **Duas classes com alto acoplamento são muito dependentes entre si.**
  - ✓ **Duas classes com baixo acoplamento são menos dependentes entre si.**

Encapsulamento

Durante o projeto de classes, todo esforço deve ser empregado para tornar as classes menos acopladas.

Encapsulamento

- **O encapsulemento ajuda neste esforço pois diminui o acoplamento entre duas classes.**

- **Conseqüência: o código torna-se mais flexível a mudanças.**

Encapsulamento

- Métodos Get / Set
  - **O que fazer quando precisamos:**
    - **Obter o valor de um atributo privado?**
    - **Modificar o valor um atributo privado?**

Encapsulamento – método get()

- Métodos Get ()
  - **Também chamado de getter.**
    - **Obtém o valor de um atributo privado.**
    - **Seu uso é irrestrito pois não modifica o atributo consultado.**

Encapsulamento – método get()

```
private Tipo atributo;


public Tipo getAtributo() {
    return this.atributo;
}
```

Encapsulamento – método get()

- Exemplo: definição do método get() na classe:

```
private String nome;

public String getNome() {
   return this.nome;
}
```

Encapsulamento – método get()

- Exemplo: chamada ao método get()

```
Contato c = new Contato("Ana", 123);
String nome = c.getNome();
```

Encapsulamento – método set()

- Métodos Set ()
  - **Também chamado de setter.**
    - **Modifica o valor de um atributo privado.**

Encapsulamento – método set()

```
private Tipo atributo;

Public void setAtributo(Tipo atributo){
        this.atributo = atributo;
   }
```

Encapsulamento – método set()

- Implementação na classe:

```
private String nome;

Public void setNome(String nome) {
   this.nome = nome;
}
```

Encapsulamento – método set()

- **Chamada ao método set()**

```
Contato c = new Contato("Ana", "123");
String x = "Beatriz";
c.setNome(x);
```

Encapsulamento

- Métodos get/set
  - **Em geral, quando encapsulamos escrevemos todos os métodos get/set.**
  - **Isso é regra geral, mas há exceções: outros métodos que façam o papel de get/set.**

Encapsulamento

- Implementar get/set em:
  - Agenda
  - Contato