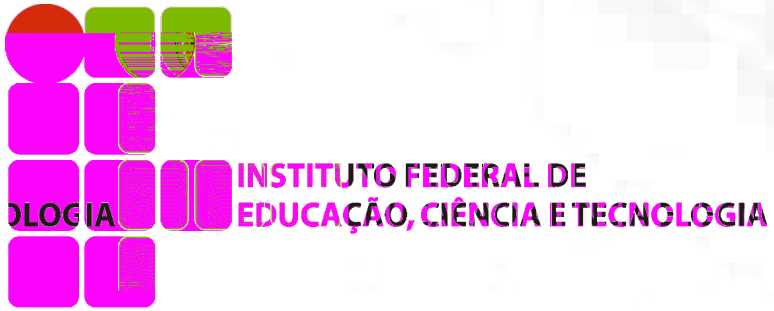
INF011 Padrões de Projeto

Memento

Sandro Santos Andrade
sandro@ufba.br

Instituto Federal de Educação, Ciência e Tecnologia da Bahia
Departamento de Tecnologia Eletro-Eletrônica
Graduação Tecnológica em Análise e Desenvolvimento de Sistemas

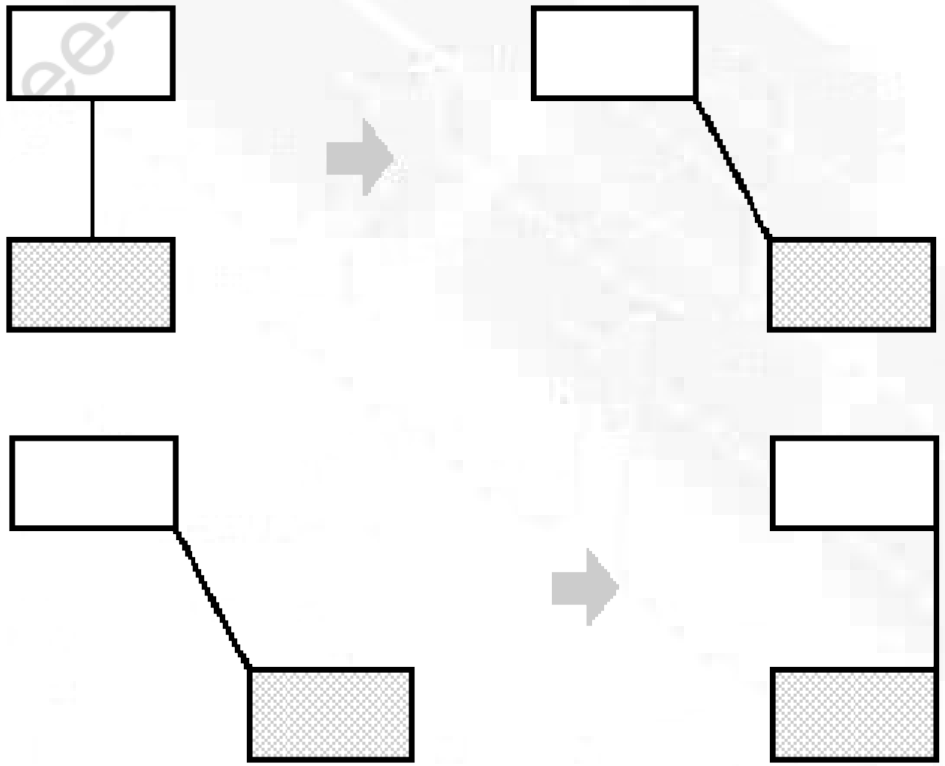


Memento

- [illegible]
  - [illegible]
- [illegible] *Token*
- [illegible]
  - [illegible] *checkpoints* [illegible] *undo*

Memento

- j f2 d j C
  - rf j2 ~e ~e fb j j j @ os o s @ sj. t s j2 o *ConstraintSolver* C ~e s o j b d j r s*undo*



Memento

- j f2 d j C
  - *Memento* p j. t s2 ps dẽ d s @p *snapshot* r j so2dr j f @ẽ j@r sj p4ẽ j. t s2 j **originador** r j *Memento*
  - ẽ soqj j ẽẽs2 r s*undo* j pqj r j *Memento* C
    - srf2 ẽ j j p sls2 j d2sẽd r dj ˜: sẽd j *move()* ẽs fpfb2dp *Memento* r j *ConstraintSolver*
    - *ConstraintSolver* fed sẽs2 ẽ @j *Memento* f @2 f@dr s *SolverState* j @ r@ so2ep2pẽdor sr dr j o psr so ẽs s j so2dr j d2pdr do s pd osos dfẽ s d @ẽ @dor j *ConstraintSolver*
    - fdo2dẽ s pdr@ j popfẽj r sdfb sẽdj ˜: sẽd j j srf2 ẽ s f@dr s j 2dj *SolverState* ˜: dẽdj *ConstraintSolver*
    - dosdr j @ @ẽ d j r j *SolverState* j *ConstraintSolver* j rffb d opdo so2ep2pẽdd @ẽ @do ˜: dẽdj so2dr j d @fẽj ẽ

Memento

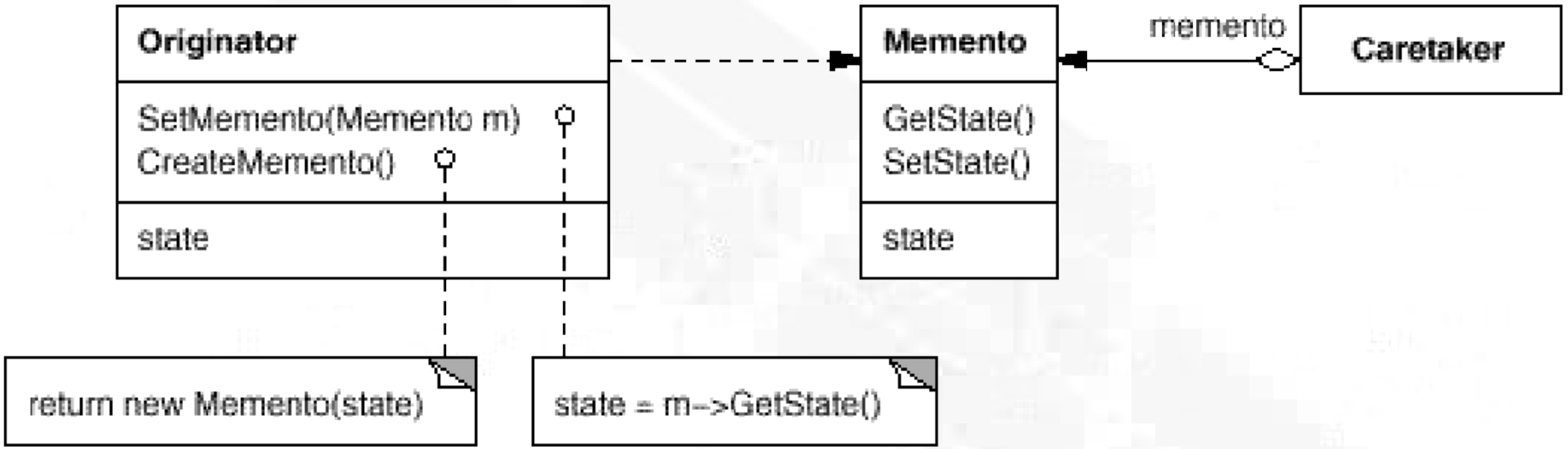
- ~:f dffr dr s C
- pdr@r sos ds os od dep *snapshot* rj so2drj j p~:dez2s rj so2drj r s p j.t s}2 r s j rj ps s s~:j ood os~e ~es o2dp~edrj ~:d~ed so2s so2drj s p j s j@ hp2p~e

E

- f2 dep df @~edo d s rf~es2d~:d~edj. 2s~ej so2drj ffedsl ~:j ~e r s2d s or sf ~j s s @ j s fj dej s @d:op d s j@rj j.t s}2

Memento

- o2ep2ped C



Memento

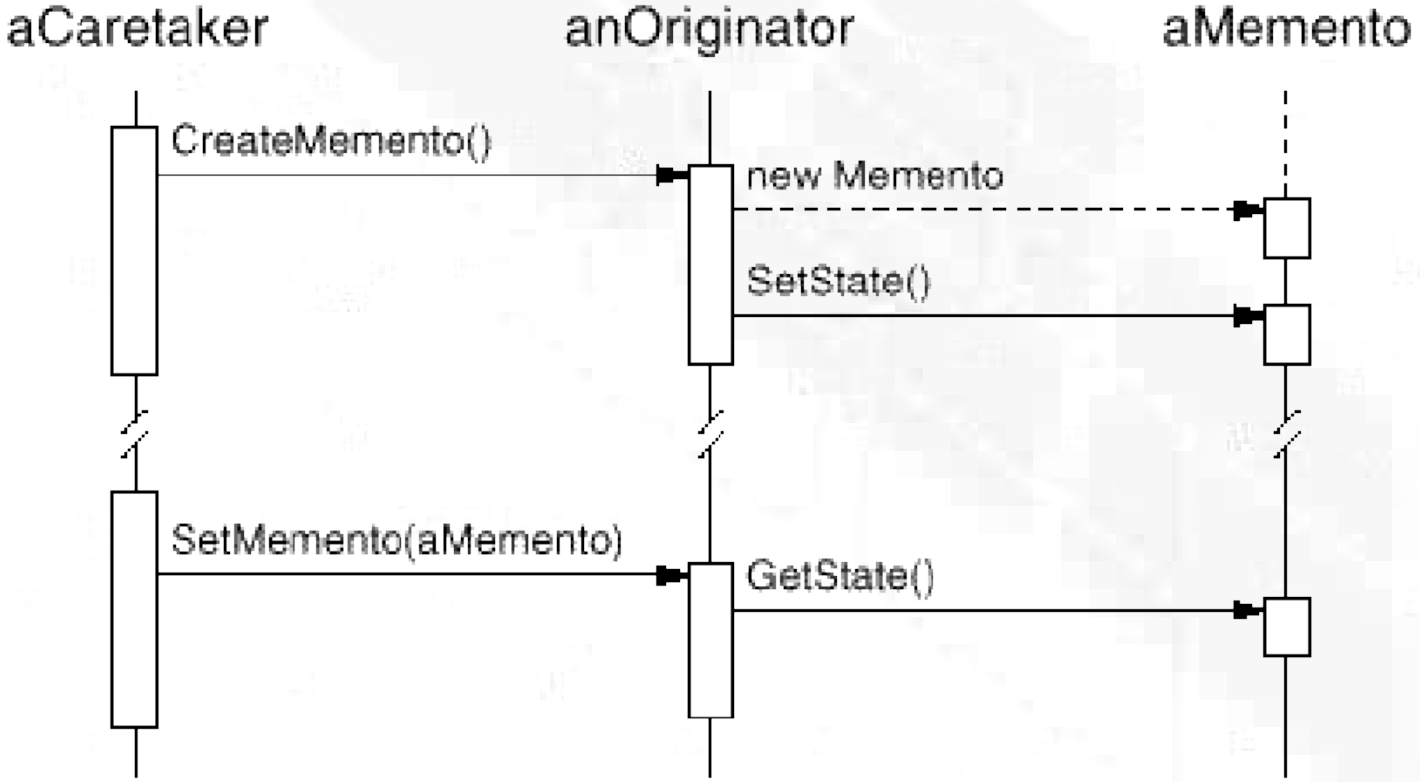
- dẽf2f ˜: d @o C
  - *Memento* j sẽ 2d2s C
    - ˜e d s @j so2drj f @ẽj@rj j. t s2 j fef @j ˜eu j r s dẽ d s @ej s @j so2drj f@j @soo fej ˜: dẽd d rf o ˜esf2 d j rj j. t s2
    - N˜: sr sj d soj r sj p2e j. t s2 o ps @ os d j j fef @j ˜e
    - *Mementos* ˜:j oops r pddf @ẽd so C
      - d˜soẽ2d *narrow* pf2 d d˜: sj *Caretaker* j s @˜: dẽd ˜: doodẽj *Memento* ˜: dẽdj p2e oj. t s2 o
      - d d ˜: d *wide* pf2 d d˜: sj j fef @j ˜e ˜: dẽd d soodẽ 2rj oj or drj o @soo fej o ˜eso2dpẽd j rj so2drj ˜:˜e fj

Memento

- dõf2f ˜: d @&o C
  - *Originator* xj @dõfd @j s˜e C
    - xfedj *Memento* j @ r@j *snapshot* rj so2drj f @&˜ej@ d2pd
    - odj *Memento* ˜: dõd˜es o2dpõdõe osp so2drj f @&˜ej@
  - *Caretaker* s df@ j r s*undo* C
    - õsõj @ s ˜: s d po2rf drj *Memento*
    - Fp @dj ˜: sõdj *Memento* j p s d f @dj osp j @& rj

Memento

- xj d j ẽd osoC
  - *Caretaker* qj f f 2dj *Memento* qj j f̃ef @dj ẽj d @
    :j ẽp 2s ẽj s s @j j s f@dr s j 2d qj j f̃ef @dj ẽ



Memento

- xj @s p f@doC
  - ~s os~o dj of f 2s or s s @d: op d s j@C
    - *Memento* sf 2ds ~:j ~e j@e d oso ps oj s @j jfef @dj ~er s sfed df@p de do~:esf od os~e de d s @d oj p~edr j jfef @dj ~e
    - Nj dj p2j oj t sP or sr s2d sd f @~ej@o~j 2s f@d s @j ~: dj o~~s os~o dr@j of f 2s orj s @d: op d s j@
  - f ~:ffb dj jfef @dj ~eC
    - j p2edo oj p os oj jfef @dj ~e d @fed s~eos or s so2drj d f @~ej@p~s pfb2drj o~:sj o f s @o
    - Nj P 2dfed jPr d d~es o~j @dffr dr sr s s~es f@d s j@ r so2s de d s @ s j@~:de dj jfef @dj ~e
    - j r sf rf ~e psj o f s @o s~es f@de j j so2drj d @ os j jfef @dj ~efb ~: so

Memento

- xj @os p f@do C
  - pọ rj *Memento* ˜:j r s s˜ede˜ed j2 o poj2 o C
    - j2 *overhead* ˜:j r s os˜e s˜edrj osj j fe f @j ˜ef2 s˜er s j ˜f de ˜ed r@so pd f@ dr s or sf p˜e d j ˜: dedj *Memento* j p osj o f s @o fed s˜esj2 ˜e @ *Mementos* ˜: dedj j fe f @j ˜e p j2 bes ps @ s @
    - ˜: dr ˜ej ó d:˜ ˜:fodj pd r@j s @: op d s j2@ s ˜es o2dp˜ed j rj s o2drj rj j fe f @j ˜e 2 poj2. d j
  - sfbf @@ dof @˜ebd s *narrow* s*wide* C
    - j r s os˜erf bf s d p dof @pd s @ d˜ed f2e ps ọ s @j j fe f @j ˜e ˜e d soodej s o2dr j rj *Memento*
  - x poj2 o df f j @or s po2rf dr s*Mementos* C
    - *Caretaker* ˜es o˜:j @ s ˜: s dr sodj d j r s *Mementos* ˜:j ˜e s s @ od sj 2d d @ rj *Memento*

Memento

- N~: s s @d j C
  - p~j ~e2sr df @pd s C
    - d~ef ~: s s @d~edf @~ed s o*narrow* s*wide* d
      f @pd s r s~:~ ~d d j r s sf~edfr sd s @
      op~:j ~e2d~erjf o @s or s~:~e 2s j so2f2 dC

```
class State;

class Originator {
public:
    Memento* CreateMemento();
    void SetMemento(const Memento*);
    //// ....
private:
    State* _state;          //// internal data structures
    //// ....
};
```

Memento

- N˜: s s @d j C
  - p˜j ˜e2sr df @pd s C
    - d˜ed f ˜: s s @d˜edd f @˜ed sd *narrow* s*wide* d
      f @pd s r s˜:˜p ˜ed d j r s sf˜edf r sd s @
      op˜:j ˜e2d˜erjf o @s or s˜:˜e 2s j so2f2 dC

```
class Memento {
public:
    // narrow public interface
    virtual ~Memento();
private:
    // private members accessible only to Originator
    friend class Originator;
    Memento();

    void SetState(State*);
    State* GetState();
    // ...
private:
    State* _state;
    // ...
};
```

Memento

- N˜: s s @d j C
  - ˜e d s @r@ pr d @df @es s @d oC
    - pdr@j *Mementos* oj fedrj osr s j f r j o˜: dedj jfef @j ˜e s p d os p f@d˜: ˜esf o2dj *Memento* ˜:j r s de d s @e j s @d pr d @f @es s @d r j so2drj r j jfef @j ˜e
    - l Cj dr@ *undoable* ˜j r s pf2 de *Mementos* ˜: ded ded f@e ps j dr@ o os d ˜es o2dpedrj o˜: dedj osp so2drj d @fej ˜e pdr@ p˜es r sohs p2 o

Memento

- x´rf j s s ˜:j C

```
class Graphic;
    // base class for graphical objects in the graphical editor
class MoveCommand {
public:
    MoveCommand(Graphic* target, const Point& delta);
    void Execute();
    void Unexecute();
private:
    ConstraintSolverMemento* _state;
    Point _delta;
    Graphic* _target;
};
```

Memento

- x´rf j s s ˜:j C

```
class ConstraintSolver {
public:
    static ConstraintSolver* Instance();

    void Solve();
    void AddConstraint(
        Graphic* startConnection, Graphic* endConnection
    );
    void RemoveConstraint(
        Graphic* startConnection, Graphic* endConnection
    );

    ConstraintSolverMemento* CreateMemento();
    void SetMemento(ConstraintSolverMemento*);
private:
    // nontrivial state and operations for enforcing
    // connectivity semantics
};
```

Memento

- x´rf j s s ˜:j C

```
class ConstraintSolverMemento {
public:
    virtual ~ConstraintSolverMemento();
private:
    friend class ConstraintSolver;
    ConstraintSolverMemento();

    // private constraint solver state
};
```

```
void MoveCommand::Execute () {
    ConstraintSolver* solver = ConstraintSolver::Instance();
    _state = solver->CreateMemento(); // create a memento
    _target->Move(_delta);
    solver->Solve();
}
```

```
void MoveCommand::Unexecute () {
    ConstraintSolver* solver = ConstraintSolver::Instance();
    target->Move(-delta);
state                solver->SetMemento(_state); // restore solver
                     solver->Solve();
                }
```

Memento

- ǫ o j @sfrj oC
  - *UniDraw*
  - d @
  - x

Memento

- d r ẽ o s o ẽ s d f j @ d j o C
  - *Commands* ̃ j r s p f 1 d ẽ *Mementos* ̃ : d ẽ d d @ ẽ j s o 2 d r j r s j ̃ : s ẽ d o s o ẽ s s ẽ o $ s o
  - *Mementos* ̃ j r s o s ẽ p f 1 d r j o ̃ : d ẽ d f 2 s ẽ d o s o

INF011 Padrões de Projeto

Memento

Sandro Santos Andrade
sandro@ifba.edu.br

Instituto Federal de Educação, Ciência e Tecnologia da Bahia
Departamento de Tecnologia Eletro-Eletrônica
Graduação Tecnológica em Análise e Desenvolvimento de Sistemas