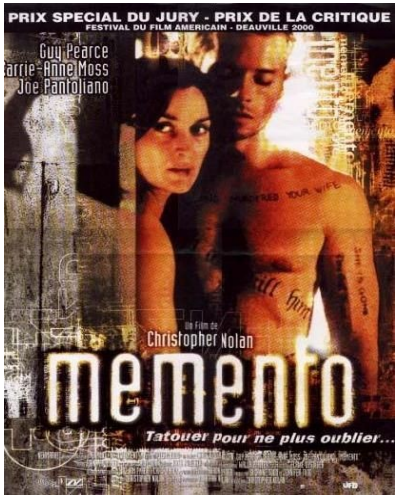
Memento

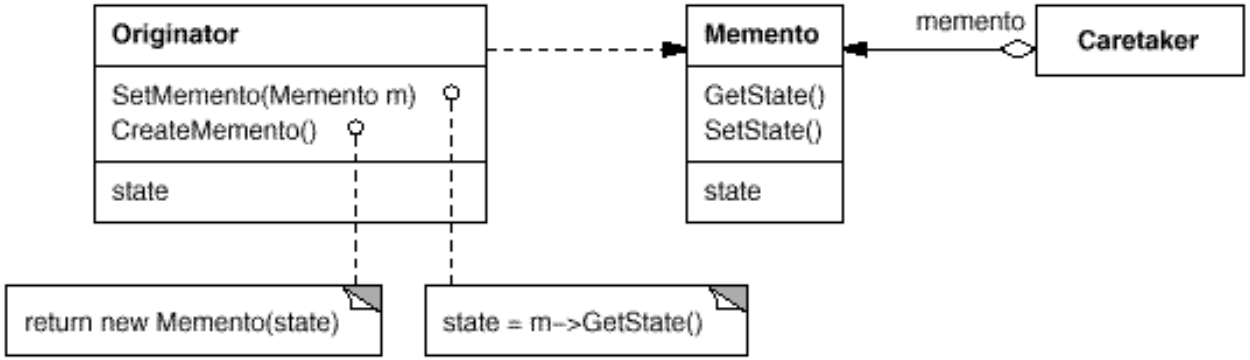

PRIX SPECIAL DU JURY - PRIX DE LA CRITIQUE
FESTIVAL DU FILM AMERICAIN - DEAUVILLE 2000
Guy Pearce
Carrie-Anne Moss
Joe Pantoliano
Un Film de
Christopher Nolan
memento
Tatouer pour ne plus oublier...

Memento

- **Účel**
  - kategorie: behavioral patterns
  - uchovávání vnitřního stavu objektu
- **Motivace**
  - undo/rollback
  - checkpointy
  - pro dočasné operace nebo zotavení z chyby
- **First Thoughs**
  - metoda `void save()` a `void load()`
  - metoda `smt save()` a `void load(smt)`
  - spoustu stavů `smt` potřebujeme správce stavů
  - objekt `smt` je **`Memento`**

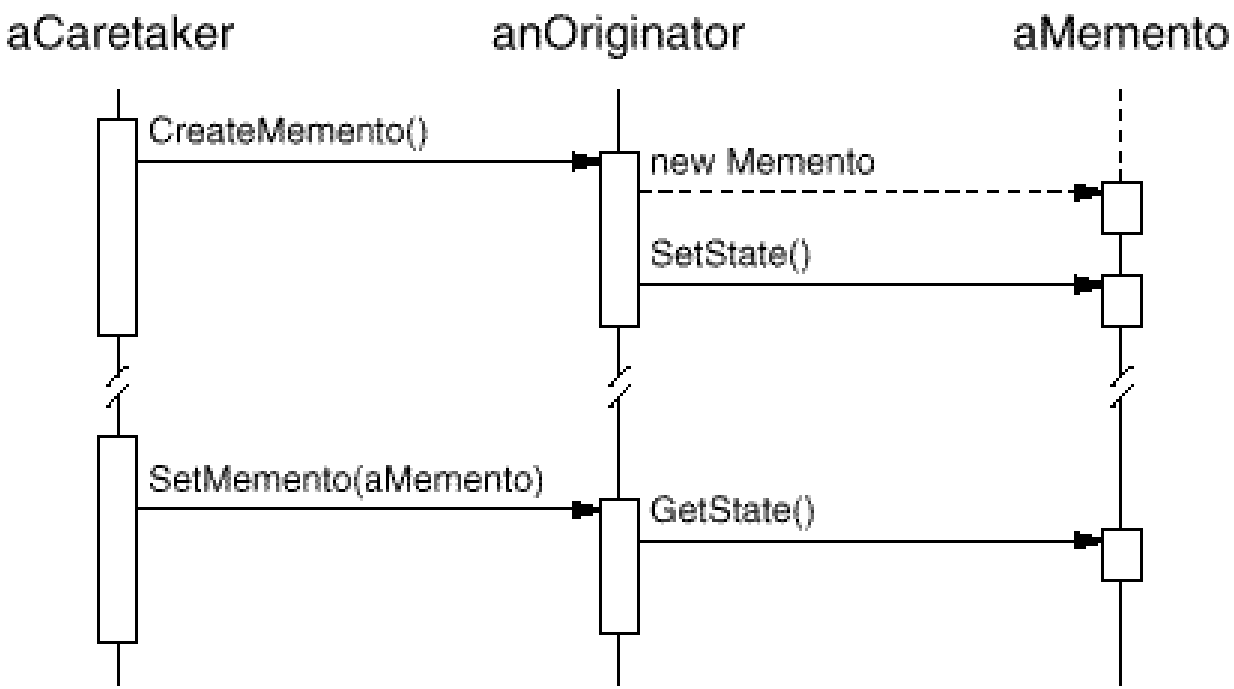
Memento – struktura

- **Struktura**



- **Účastníci**
  - **`Originator`**
    - objekt, jehož vnitřní stav chceme uložit
  - **`Memento`**
    - obsahuje vnitřní stav `Originatoru`
  - **`Caretaker`**
    - řídí ukládání/načítání `Originátoru` do/z `Mementa`
    - nezná vnitřní strukturu objektu `Memento`
    - nezná vnitřní strukturu objektu `Originator`

Memento - interakce



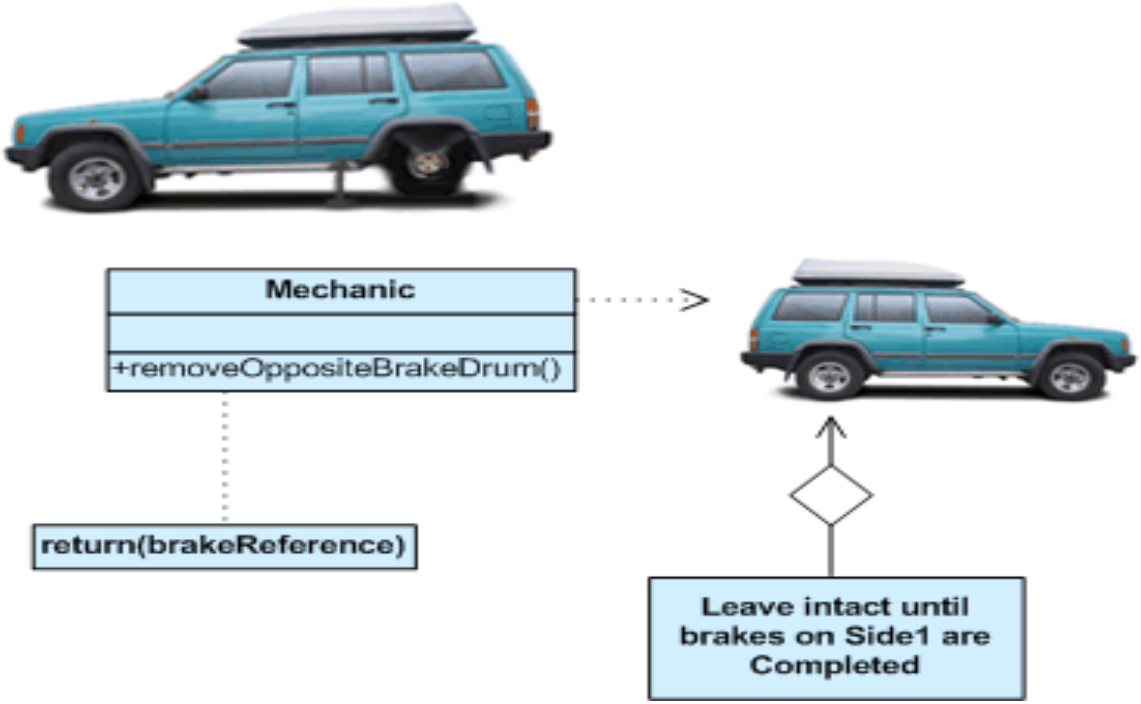
Memento – souvislosti

- **Vlastnosti**
  - uchovává vnitřní stav objektu a umožňuje jeho pozdější obnovení
  - uchovává stav mimo vlastní objekt
  - `Memento` může být paměťově náročné
  - `Memento` je úzce svázané s typem objektu jehož obsah si pamatuje
  - `Caretaker` nezná vnitřnosti `Mementa` nebo `Originátoru`
  - `Caretaker` může aplikovat různé strategie pro uchovávání/zahazování

- **Využití**
  - ovládání rollbacku u databází
  - undo v editorech
  - ...

Memento – příklady

- **Kutil opravující brzdy u auta**
  - Auto má od každého kola dva exempláře
  - Rozebere jen jedno kolo
  - Druhé hraje roli Mementa
    - Podle něj se rozbrané kolo zase složí dohromady



Memento – příklady

- **Píšeme hazardní hru - kostky/poker/mariáš/solitaire**
  - Hráč chce mít možnost si hru uložit
  - Hráč chce mít možnost zkusit si zachovat se za stejných podmínek jinak
  - Použijeme `memento` pro uložení `seed` generátoru náhodných čísel

Memento – příklady

- **Kostka - Originator**

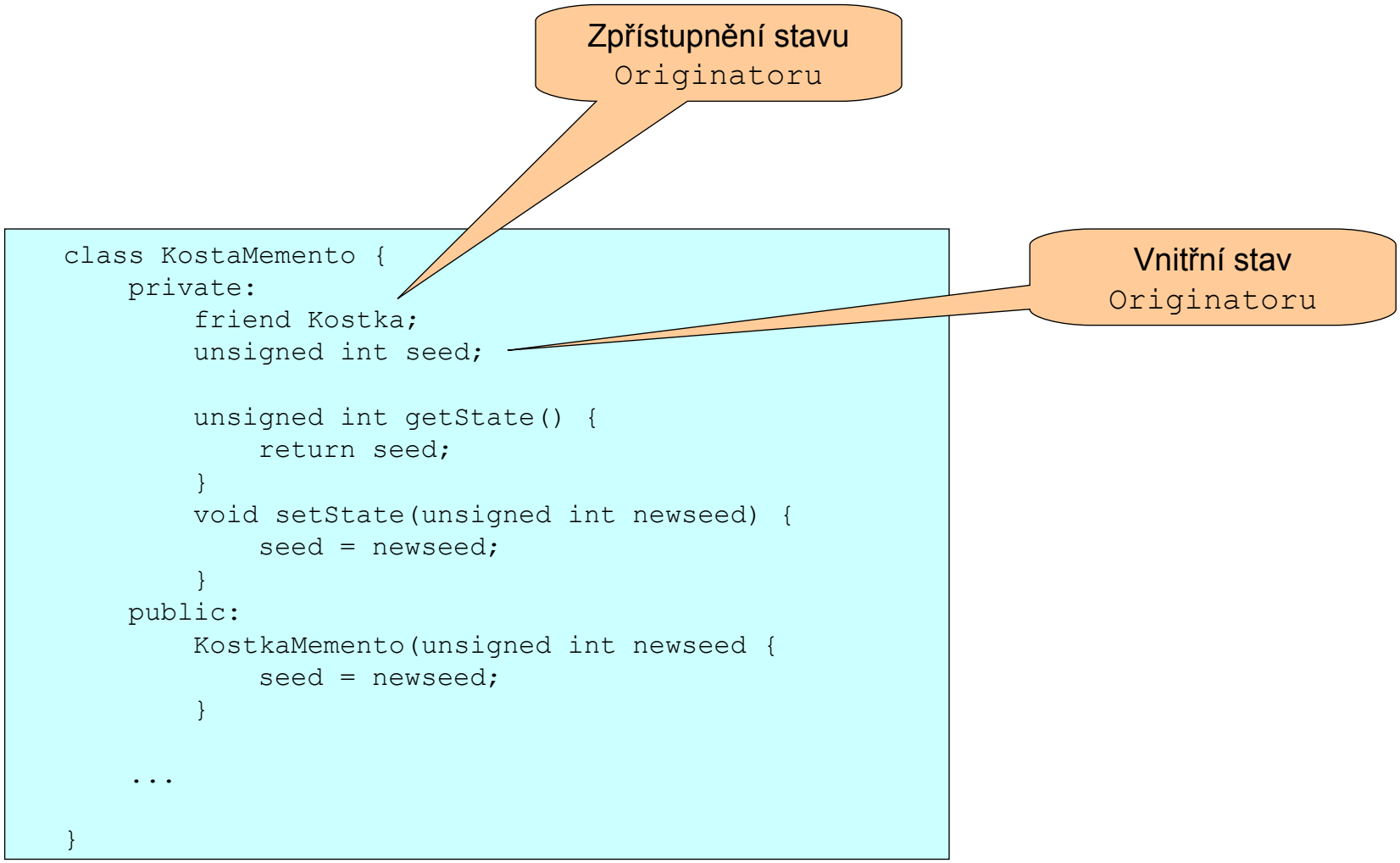
```
public class Kostka {
   private:
       unsigned int seed;

   public:
       unsigned int hod() {
           seed = random();
           srandom(seed);
           return seed%6+1;
       }

       void setKostkaMemento(KostkaMemento* In) {
           srandom(In.getState());
       }

       KostkaMemento* createKostkaMemento() {
           return new KostkaMemento(seed);
       }

   ...
```

Obnoví vnitřní stav `Originatoru` z `Mementa`

Vytvoří `memento` s aktuálním stavem

Memento – příklady

- **KostkaMemento**




```
class KostaMemento {
    private:
        friend Kostka;
        unsigned int seed;

        unsigned int getState() {
            return seed;
        }
        void setState(unsigned int newseed) {
            seed = newseed;
        }
    public:
        KostkaMemento(unsigned int newseed {
            seed = newseed;
        }

    ...

}
```

Memento – příklady

- **Klient - Caretaker**

```
public class GameSaver {
   private:
       zasobnik redo;
       zasobnik undo;
   ...

   public:
       void save() {
           redo.clear();
           undo.push(kostka.createKostkaMemento());
       }

       void undo() {
           KostkaMemento* tmp;
           tmp = undo.pop();
           if(tmp) {
               kostka.setKostkaMemento(tmp);
               redo.push(tmp);
           }
       }

       ...

   }

}
```

Uložení aktuálního stavu do mementa

Obnova dat z mementa

Memento – implementace - Java, C++, C#

- **Java**
  - `Memento` je vnitřní třída třídy `Originator`
    - Bez problémů přistoupí k privátním položkám třídy `Originator`
- **C++**
  - `Memento` – privátní metody
    - Zpřístupněné `Originátoru` pomocí klíčového slova `friend`

```
class Originator {
public:
   Memento* CreateMemento();
   void SetMemento(const Memento*);

private:
   State* _state;
};
```

```
class Memento {
public:
   virtual ~Memento();
private:
   friend class Originator;
   Memento();
   void SetState(State*);
   State* GetState();
private:
   State* _state;
};
```

Metody zpřístupněné Originatoru