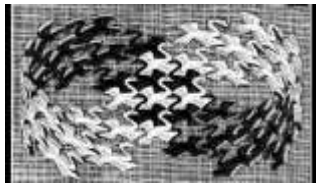
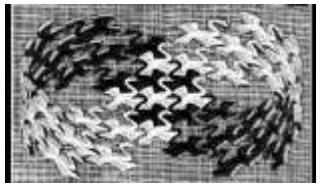
Flyweight



Mucha ľahšia ako mravec?

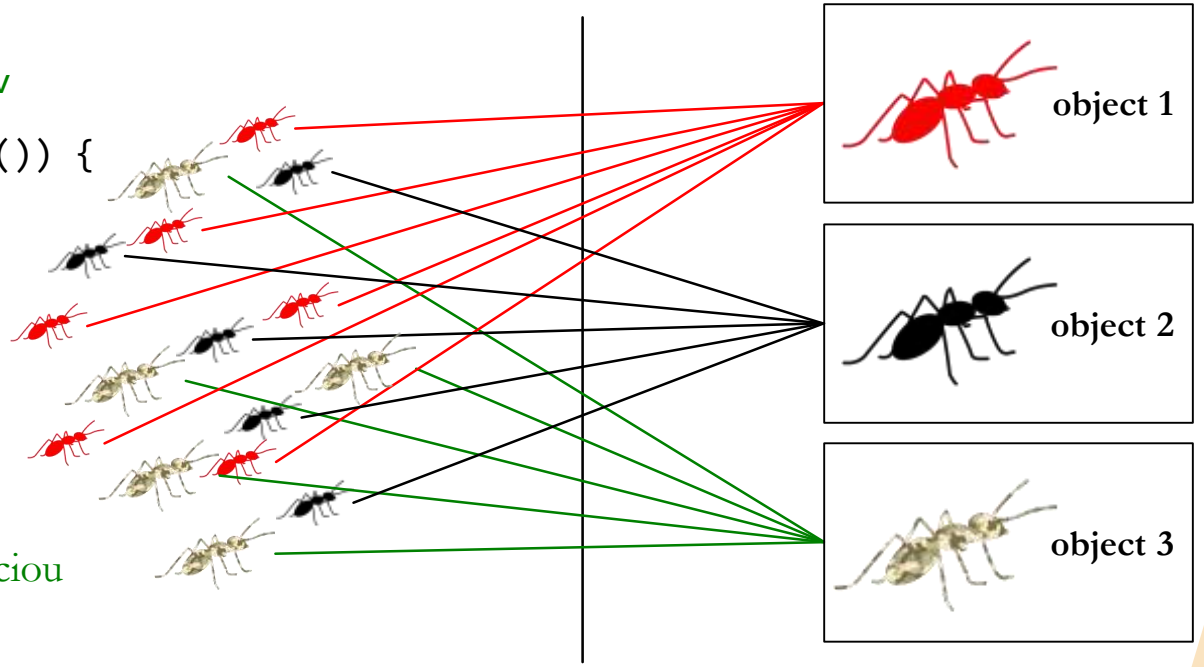
```
// chceme vykresliť všetkých mravcov
foreach (Ant ant in antHill.GetAnts()) {
     Render(ant);
}
```

☹ každý mravec – objekt zaberá miesto v pamäti



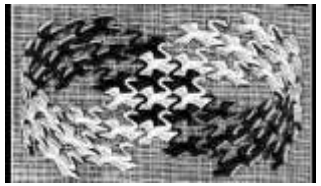
Mucha ľahšia ako mravec?

```
// chceme vykresliť všetkých mravcov
foreach (Ant ant in antHill.GetAnts()) {
     Render(ant);
}
```



☺ rôznych „typov“ mravcov je málo (robotníci, vojaci, čierne, červené)

☺ vykresľované mravce sa líšia najmä pozíciou

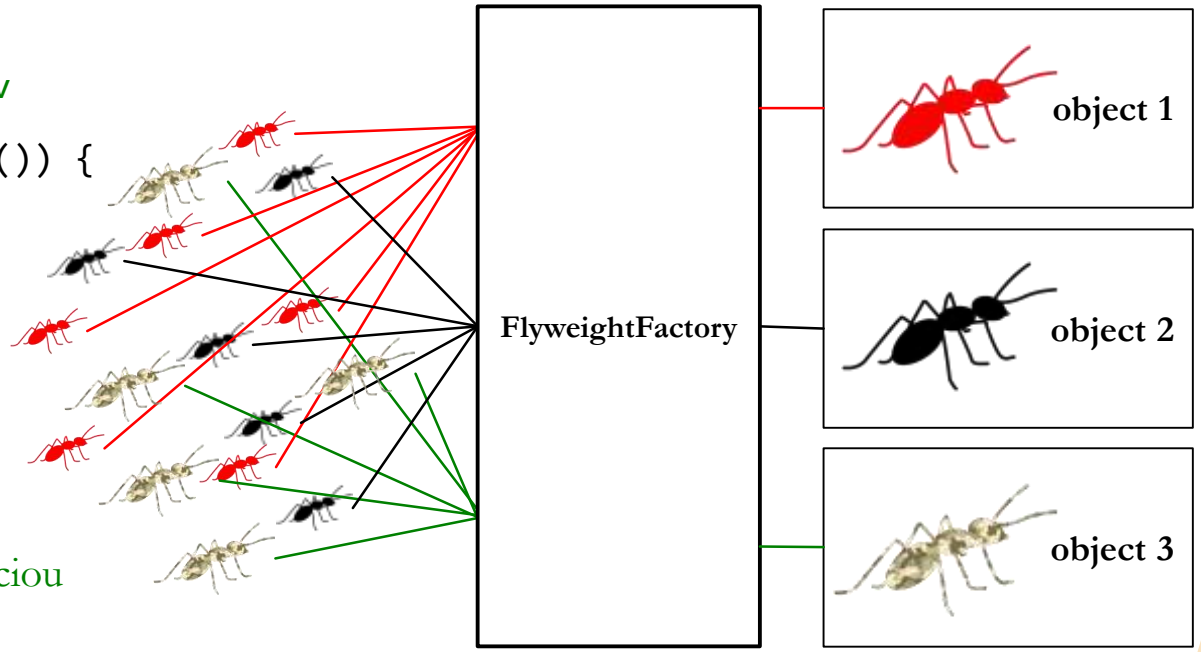


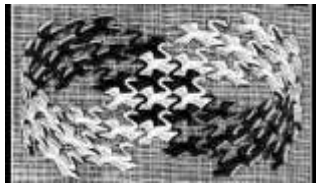
Mucha ľahšia ako mravec?

```
// chceme vykresliť všetkých mravcov
foreach (Ant ant in antHill.GetAnts()) {
     Render(ant);
}
```

☺ rôznych „typov“ mravcov je málo
(robotníci, vojaci, čierne, červené)

☺ vykresľované mravce sa líšia najmä pozíciou

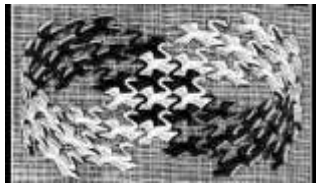




Návrhový vzor Flyweight

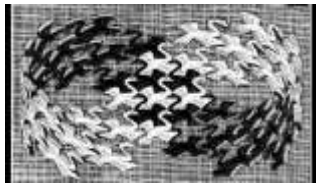
- štrukturálny návrhový vzor
- efektívnejšie využívanie pamäti
- **flyweight** objekt
  - zdieľaný objekt, ktorý môže byť súčasne použitý vo
  - v každom použitom kontexte sa tvári ako nezávislý
    - nedá sa odlíšiť od nezdieľaného objektu
  - reprezentuje mnohopočetné entity
  - rozoznáva dva stavy

trochu zvětšit písmo - špatně čitelné



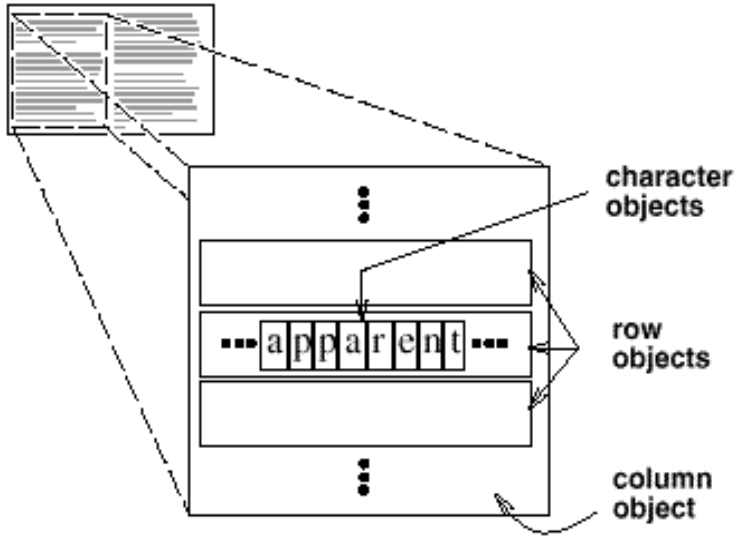
Kľúčová ! vlastnosť flyweight objektov

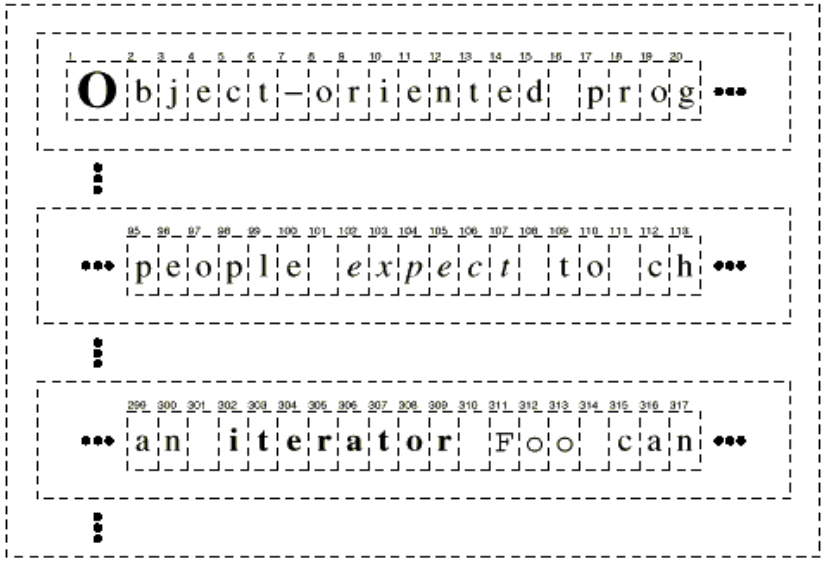
- rozoznáva dva stavy
- **Intrinsic (vnútorný) stav**
  - dáta uložené priamo vo flyweight objekte
  - nezávislé na použitom kontexte
  - nebráni zdieľaniu – môže byť zdieľaný
- **Extrinsic (vonkajší) stav**
  - flyweight objektu dodávaný z vonkajšieho prostredia
  - stav / kontext počíta (ukladá) klient
  - závisí na kontexte – nie je možné ho zdieľať
- **Volanie metód klientom**
  - flyweight pozná svoj stav
  - volajúci (klient) dodá vonkajší stav / kontext



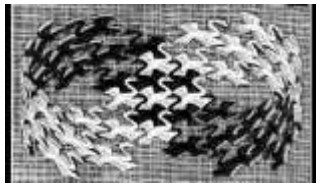
Objektovo orientovaný dokumentový editor

- **Umožňuje**
  - podpora pre nové znakové sady
  - schopnosť editovať / formátovať text, príp. iné elementy
  - rovnaké zaobchádzanie s textom resp. inými elementmi
  - reprezentácia fyzickej štruktúry dokumentu
- **Naivné riešenie**
  - triedy – strana, riadok, znak
  - objekty – výskyty v dokumente
  - každý objekt obsahuje svoj kontext
- **Problém**
  - príliš veľa **podobných** objektov
  - objekty obsahujú **veľa dát**
  - nadmerná spotreba pamäti



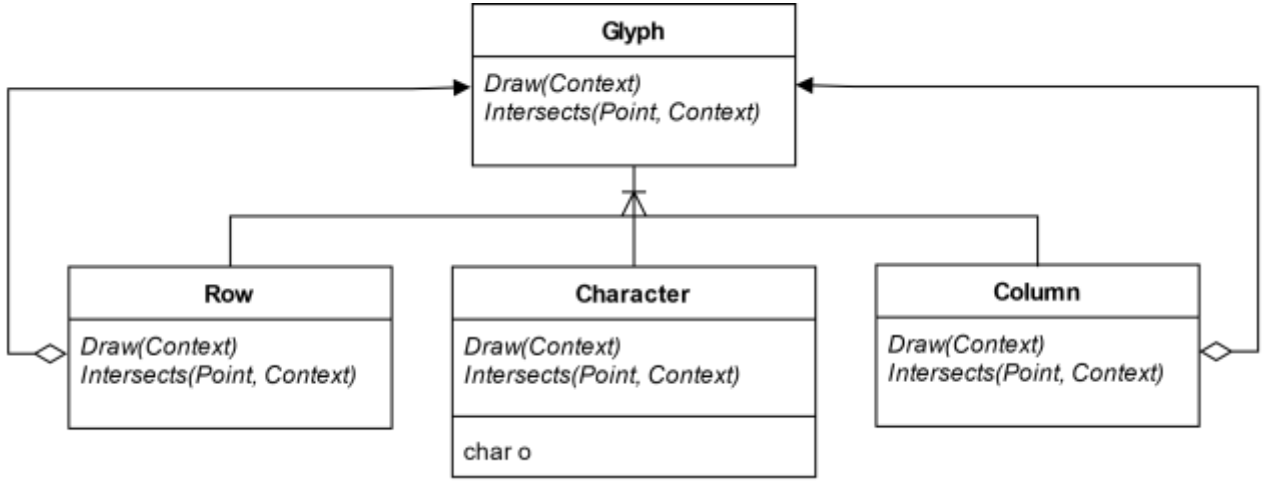


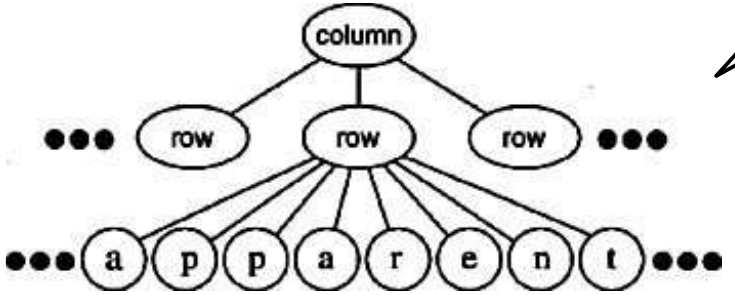
Character
+ characterCode
+ font
+ color
+ position
+ style



Objektovo orientovaný dokumentový editor pomocou Flyweight

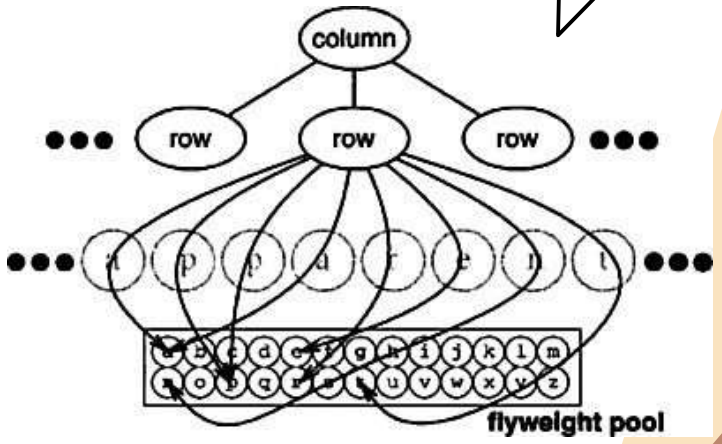
- Abstraktná trieda **Glyph** na grafické objekty
- **Character**
  - flyweight objekty pre rôzne znaky
  - intrinsic – kód znaku
  - extrinsic – font, pozícia, ...
- **Row, Column**
  - flyweight objekty bez zdieľaného intrinsic stavu

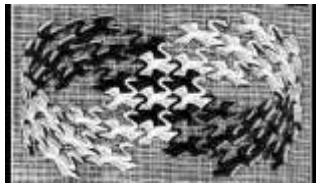




logická reprezentácia

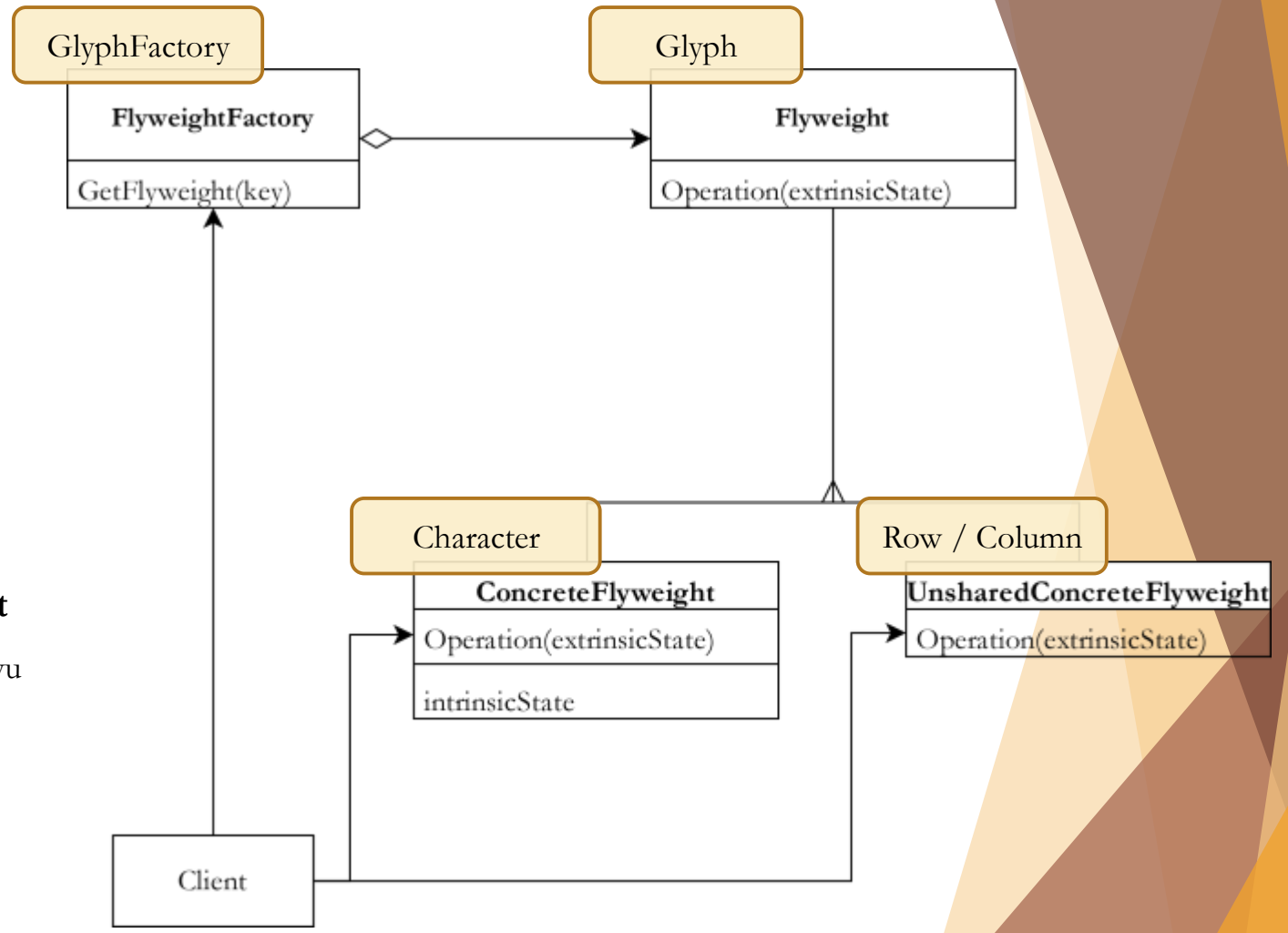
fyzická reprezentácia

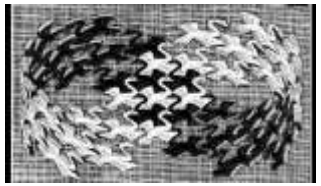




Komponenty flyweight návrhového vzoru

- **Flyweight**
  - deklaruje interface pre flyweight objekty
- **FlyweightFactory**
  - vytváranie a spravovanie flyweightov
  - zabezpečuje zdieľanie a prístup
- **ConcreteFlyweight : IFlyweight**
  - poskytuje prístup k inštancii
  - dátové položky – vnútorný stav
- **UnsharedConcreteFlyweight : IFlyweight**
  - Flyweight **nevynucuje** zdieľanie intrinsic stavu
  - obvykle obsahuje **ConcreteFlyweight** ako svojich potomkov
- **Client**
  - používa /udržiava referencie na flyweighty
  - poskytuje externý stav / kontext



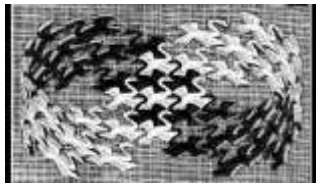


Implementácia dokumentového editoru

```
abstract class Glyph {
   // parametere poskytujúce extrinsic kontext
   public abstract void Draw(Window w, GlyphContext glContext);
}
```

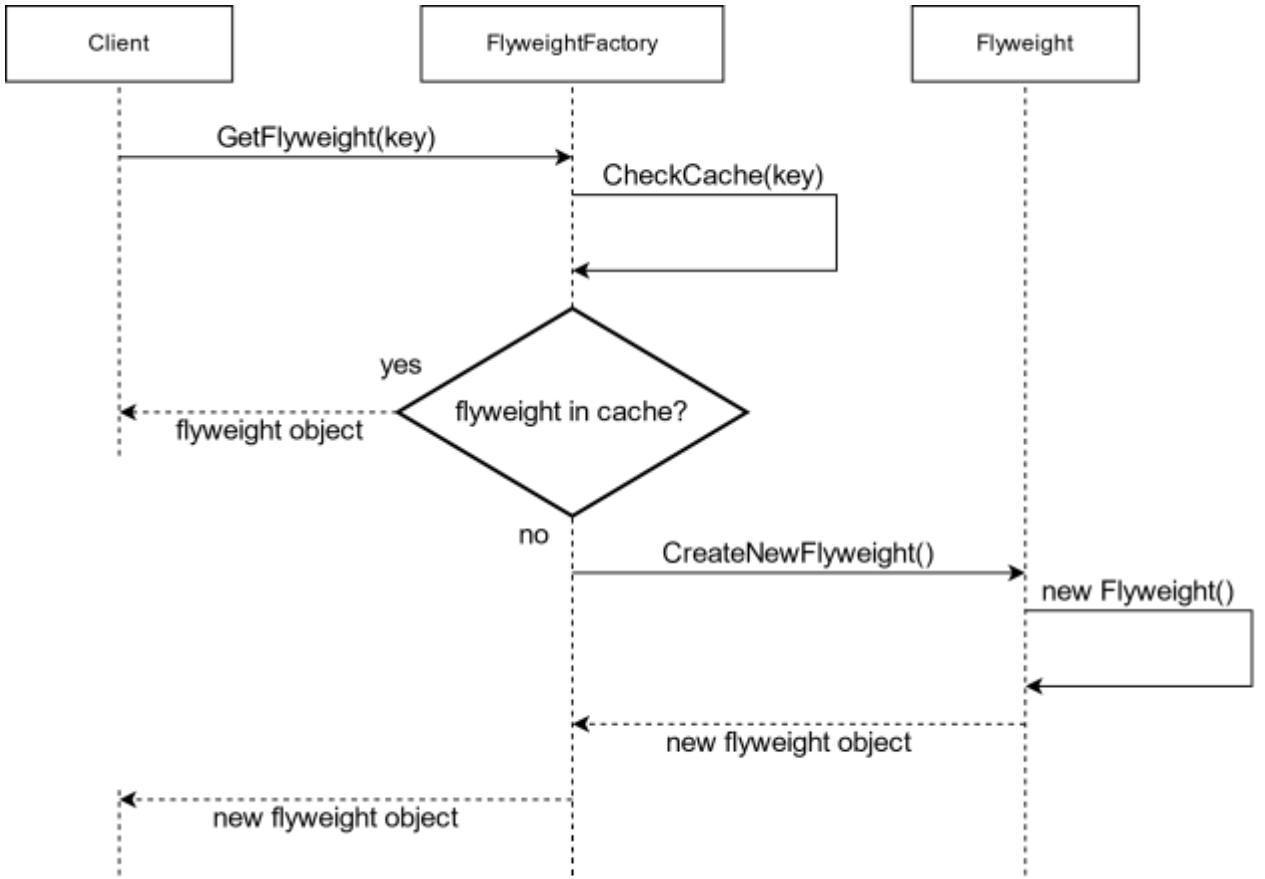
```
class Character : Glyph {
   private char charCode;  // intrinsic stav
   public Character(char c) {
      charCode = c;
   }
   public override void Draw(Window w, GlyphContext glContext) {
      // inštancia glContext ako súčasť extrinsic stavu
      // vykreslí znak "charCode" na okno "w" s použitím "glContext"
   }
}
```

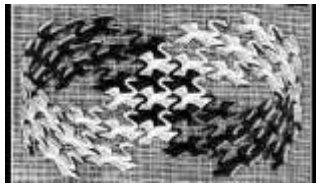
```
class GlyphContext {
   private Tree fonts;
   private int index; // context position
   public GlyphContext() {
      index = 0;
   }
   public virtual void Next(int step) {
      index += step;
   }
   public virtual void Insert(int quantity) { ... }
   public Font GetFont() {
      return fonts[index];
   }
   public void SetFont(Font font, int span) { ... }
}
```



Získanie inštancie flyweight objektu

```
static class GlyphFactory {
   private static Dictionary<char, Character> characters = new ...
   public static Row GetRow() {
      return new Row();
   }
   public static Column GetColumn() {
      return new Column();
   }
   public static Character GetCharacter(char c) {
      if (! characters.ContainsKey(c)) {
         characters.Add(c, new Character(c));
      }
      return characters[c];
   }
}
```

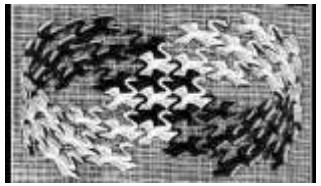




Kritéria (ne)použiteľnosti flyweight

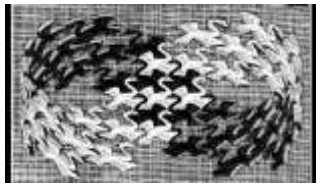
Využitie návrhového vzoru flyweight predpokladá :

1. aplikácia / využíva **veľké množstvo** objektov
2. uloženie samostatných objektov spôsobuje **veľké pamäťové nároky**
   - relatívne voči počtu objektov
3. stav objektov môže byť reprezentovaný ich **externým kontextom**
4. po odstránení **externého stavu** ostane iba **malá** množina rôznych objektov
5. aplikácia **nezávisí** na identite objektov
   - metóda `Equals()` na flyweight objektoch vráti `true` aj napriek rôznym konceptuálnym rozdielom / externým kontextom



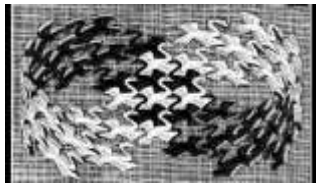
Nástrahy používania flyweight

- time-space tradeoff
  - \+ vytvorený s cieľom šetriť pamäť
  - \+ menšie množstvo potrebných alokácií objektov
  - – nutnosť manipulovať s externým kontextom
- odstránenie vonkajšieho (extrinsic) stavu
- správa zdieľaných objektov
  - klient inštancie nevytvára priamo
  - z pohľadu klienta by mali byť flyweight objekty nemenné
- garbage collection
  - počítanie referencií
  - klient by mal oznámiť **FlyweightFactory**, ak prestane flyweight používať



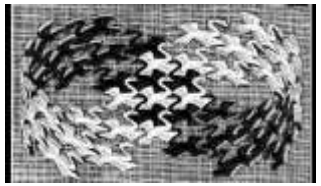
Vzťahy s inými návrhovými vzormi

- **Flyweight** sa často kombinuje s návrhovým vzorom **Composite**
- **State** a **Strategy** objekty sa odporúča implementovať ako flyweight objekty



Použitie v praxi

- **Interviews 3.0**
  - architektúra výkonného editoru dokumentov „Doc“
  - proof of concept
  - každý znak – glyph objekt
    - jedna glyph inštancia na každý rôzny znak
  - intrinsic stav
    - kód znaku
    - index do tabuľky štýlov
  - extrinsic stav
    - pozícia znaku v dokumente



Použité zdroje

Gamma, E., Helm, R., Johnson, R.,, Vlissides, J. M. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional. ISBN: 0201633612

https://en.wikipedia.org/wiki/Flyweight_pattern

https://refactoring.guru/design-patterns/flyweight