3

Objektové vlastnosti

Enumerátory, iterátory, delegáti, udalosti

KST Katedra softvérových technológií
Fakulta riadenia a informatiky, Žilinská univerzita v Žiline

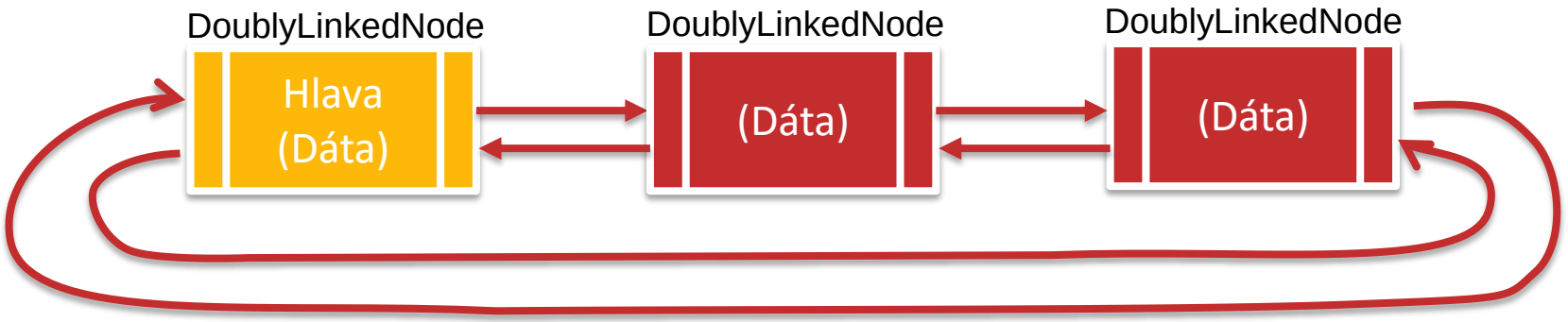
Obsah

- Vytvoríme jeden solution a v ňom dva projekty
  - Projekt typu Class Library (DLL)
  - Projekt typu Console Application (EXE)

- Naučíme sa používať:
  - Class diagram, Object Browser
  - Dokumentačné komentáre (///)
  - Rozhrania IEnumerable<T>, IEnumerator<T>, iterátor (yield)
  - Indexer v triede
  - Delegátov, udalosti

Projekt 1

- Vytvorte **projekt typu knižnica (Class Library)**
- Stiahnite si z moodle archív, rozbaľte a **pridajte do projektu triedy DoublyLinkedList** a **DoublyLinkedNode**
  - DoublyLinkedList je generická údajová štruktúra obojsmerne zreťazeného zoznamu implementovaná s hlavou
  - DoublyLinkedNode predstavuje vrchol, pomocou ktorého sú uchovávané dáta typu T, ktoré požadujeme v zozname uchovávať



Projekt 1 – úlohy

- V triede DoublyLinkedList **vytvorte** indexer, ktorý bude sprístupňovať jednotlivé prvky zoznamu (typu T) podľa indexu
- **Implementujte rozhranie** IEnumerable<T> (prípadne ICollection<T> alebo IList<T>) tak, aby bolo možné prechádzať zoznam (implementácia rozhrania nám umožní používať foreach cyklus na prejdenie všetkých prvkov zoznamu)
- **Vytvorte** novú triedu **DoublyLinkedEnumerator** implementujúcu rozhranie IEnumerator<T> a použite ju v metóde GetEnumerator rozhrania IEnumerable<T>

Projekt 2

- **Vytvorte konzolovú aplikáciu**, v ktorej otestujte objekt typu DoublyLinkedList (aby ste ho mohli použiť, pridajte si referenciu na prvý projekt obsahujúci DoublyLinkedList)

```
var numbers = new DoublyLinkedList<int>();
for (int i = 1; i < 6; i++)
    numbers.Add(i);
```

  - **vyskúšajte** si for() s **indexerom** (neoptimálne iterovanie)

```
for (int i = 0; i < numbers.Length; i++)
  Console.WriteLine(numbers[i]);
```

  - **vyskúšajte** si foreach() s **enumerátorom** (optimálne iterovanie), debugovaním si pozrite správanie enumerátora

```
foreach (var number in numbers)
    Console.WriteLine(number);
```

Projekt 1 - modifikácia

- Upravte triedy **DoublyLinkedList** takto:
  - Pridajte udalosť **Changed**, ktorá sa vyvolá pri každej zmene obsahu (t. j. po pridaní alebo odstránení prvku)
  - Nahraďte enumerátor v metóde GetEnumerator() za iterátor s použitím kľúčového slova **yield** (dopredný iterátor)
  - Pridajte do triedy **DoublyLinkedList** metódu, ktorá bude poskytovať **spätnú iteráciu** (spätný iterátor)

Projekt 2 - modifikácia

- Vráťte sa do **konzolovej aplikácie**, **registrujte sa na udalosť Changed** a otestujte ju

- **Otestujte funkčnosť** dopredného a spätného **iterátora**

KST Katedra softvérových technológií
Fakulta riadenia a informatiky, Žilinská univerzita v Žiline