

## SEMESTRÁLNÍ PRÁCE B

Maximální možný bodový zisk: **4 body**

## A) Motivační příklad:

Modifikuje příklad autopůjčovny ze semestrální práce A tak, aby jednotlivé automobily bylo možné efektivně vyhledávat dle specifikovaného klíče. Vyhledávání bude realizováno podle klíče *SPZ*.

## B) Použité datové struktury:

Abstraktní datová struktura umožňující vyhledávání dle klíče je realizována jako **binární vyhledávací strom** (BVS) v dynamické paměti (tabulka na binárním stromu). Tato abstraktní datová struktura tabulka nahradí původní lineární strukturu ADL v pobočce a umožní tak vyhledávání vozidel dle klíče.

Komponenta	ABSTRTABLE	pracuje s generickým prvkem
K extends Comparable<K>, V	(K-key, V-value)	a implementuje rozhraní:
void zrus()	zrušení celé tabulky	
boolean jePrazdny()	test prázdnosti tabulky	
V najdi(K key)	vyhledá prvek dle klíče	
void vloz(K key, V value)	vloží prvek do tabulky	
V odeber(K key)	odebere prvek dle klíče z tabulky	
Iterator vytvorIterator	(eTypProhl typ) vytvoří iterátor, který umožňuje procházení stromu do šířky/hloubky ( <b>in-order</b> )	

Iterátor využívá ADS **zásobník/fronta** (ABSTRLIFO/ ABSTRFIFO) postavenou nad ADS ze semestrální práce A (jako nová samostatná třída)

void zrus()	zrušení celé fronty/zásobníku
boolean jePrazdny()	test prázdnosti
void vloz(T data)	vloží prvek do zásobníku/fronty
T odeber()	odebere prvek ze zásobníku/fronty
Iterator vytvorIterator	vrací iterátor zásobníku/fronty

Poz:

- Iterátor BVS využívá ADS zásobník/fronta postavenou nad ADS ze semestrální práce A

C) Pro ověření funkčnosti implementovaných upravte rozhraní **Autopujcovna** a **GUI** umožňující práci nad novými strukturami:

- Najdi automobil ve všech pobočkách dle klíče
- Najdi automobil v aktuální pobočce dle klíče
- Vlož automobil do aktuální pobočky
- Odeber automobil dle klíče

**ProgAutopujcovna** necht' dále umožňuje zadávání vstupních dat z klávesnice, ze souboru a z generátoru, výstupy z programu necht' je možné zobrazit na obrazovce a uložit do souboru.