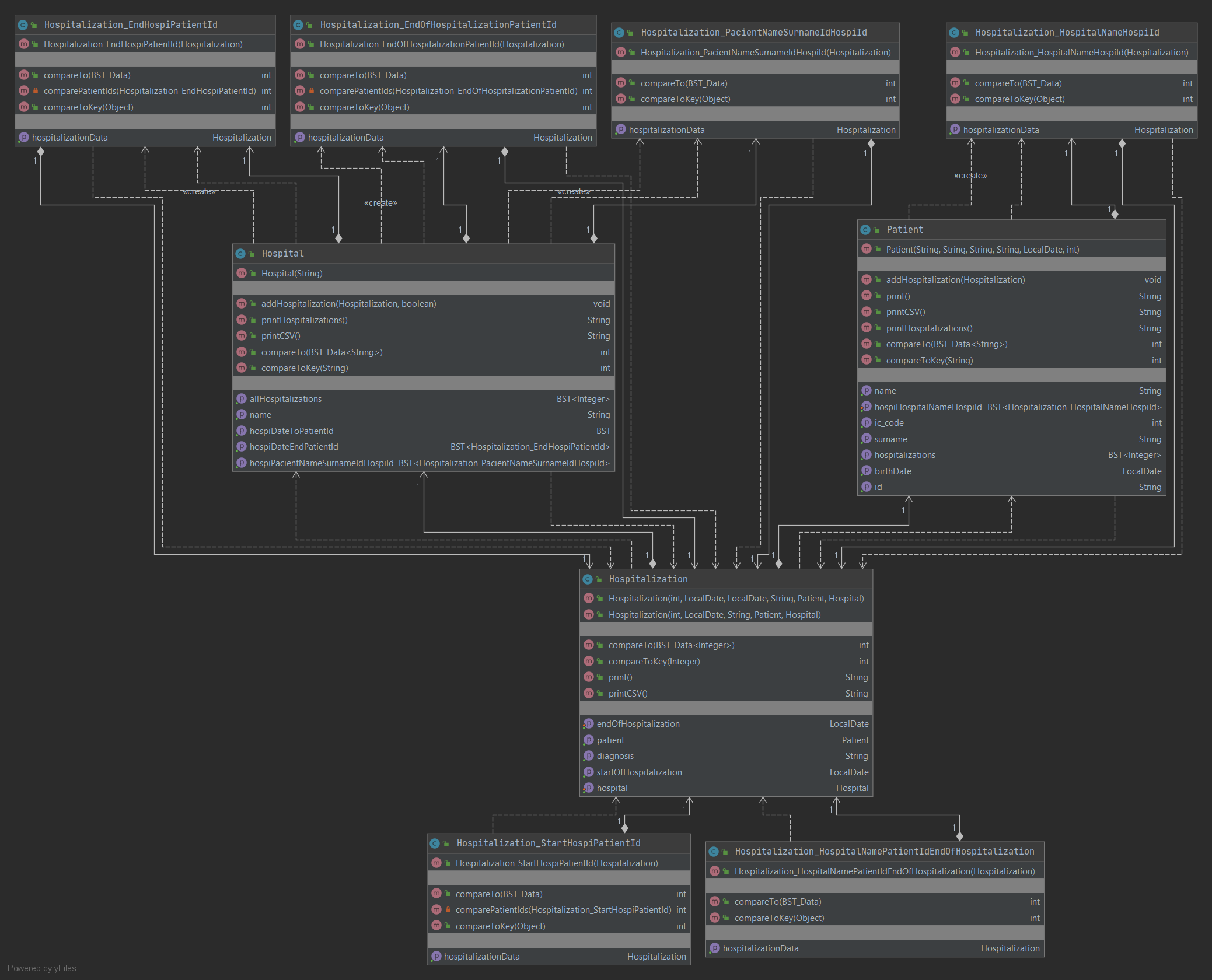
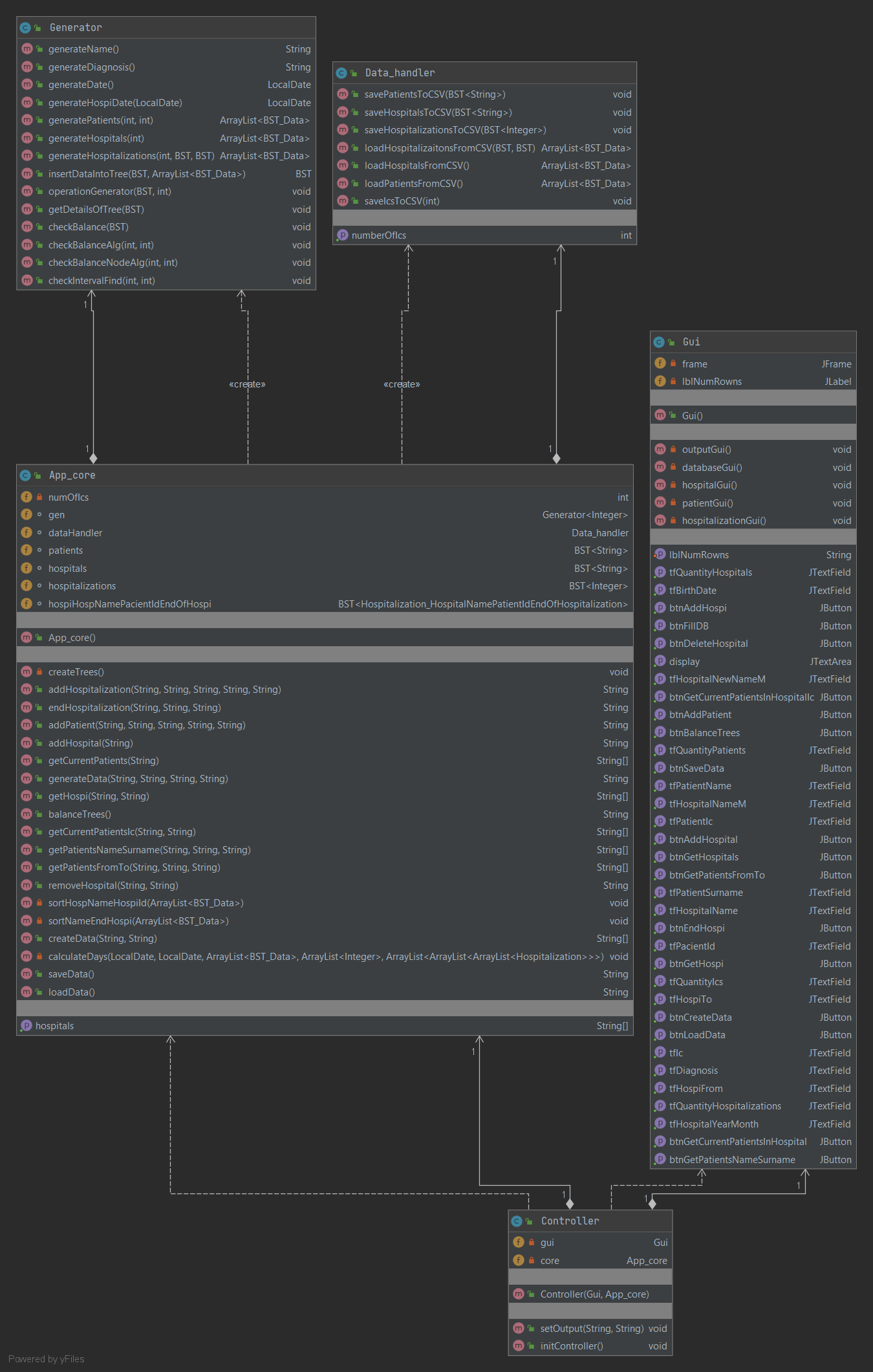
UML diagramy tried

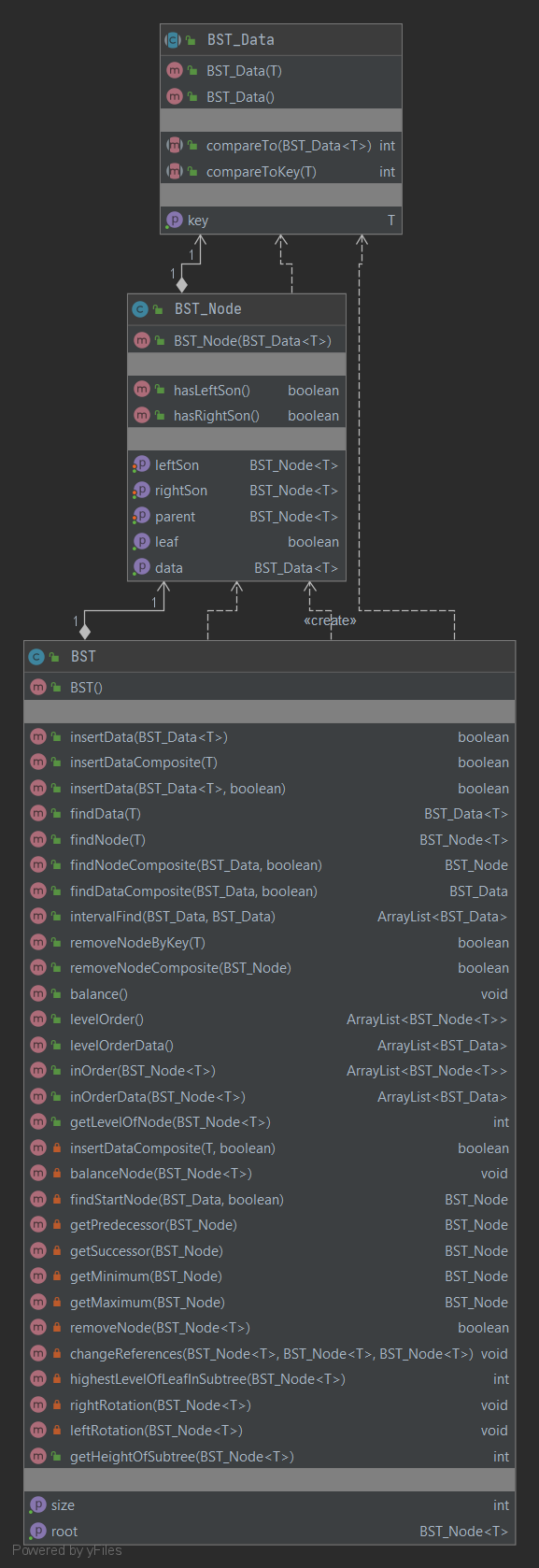
Na nasledujúcom obrázku je zobrazený diagram tried, ktoré uchovávajú dáta o jednotlivých prvkoch systému (Hospital, Patient, Hospitalization) a triedy, ktoré pomáhajú utriediť dodatočné štruktúry podľa rôznych atribútov.



Na ďalšom obrázku sú zobrazené triedy zodpovedné za beh samostatnej aplikácie.

* Trieda **Gui** - slúži na zobrazenie používateľského rozhrania.
* Trieda **App\_core** - zabezpečuje tvorbu údajových štruktúr a spracovávanie požadovaných operácií informačného systému.
* Trieda **Controller** - zabezpečuje komunikáciu medzi používateľským rozhraním a hlavnou časťou aplikácie.
* Trieda **Generator** - generuje dáta do údajových štruktúr, umožňuje naplniť prázdnu štruktúru z preddefinovaného zoznamu prvkov a obsahuje metódy na testovanie údajovej štruktúry.
* Trieda **Data\_handler** - ukladá/ načítava dáta údajových štruktúr do/zo súborov CSV formátu.



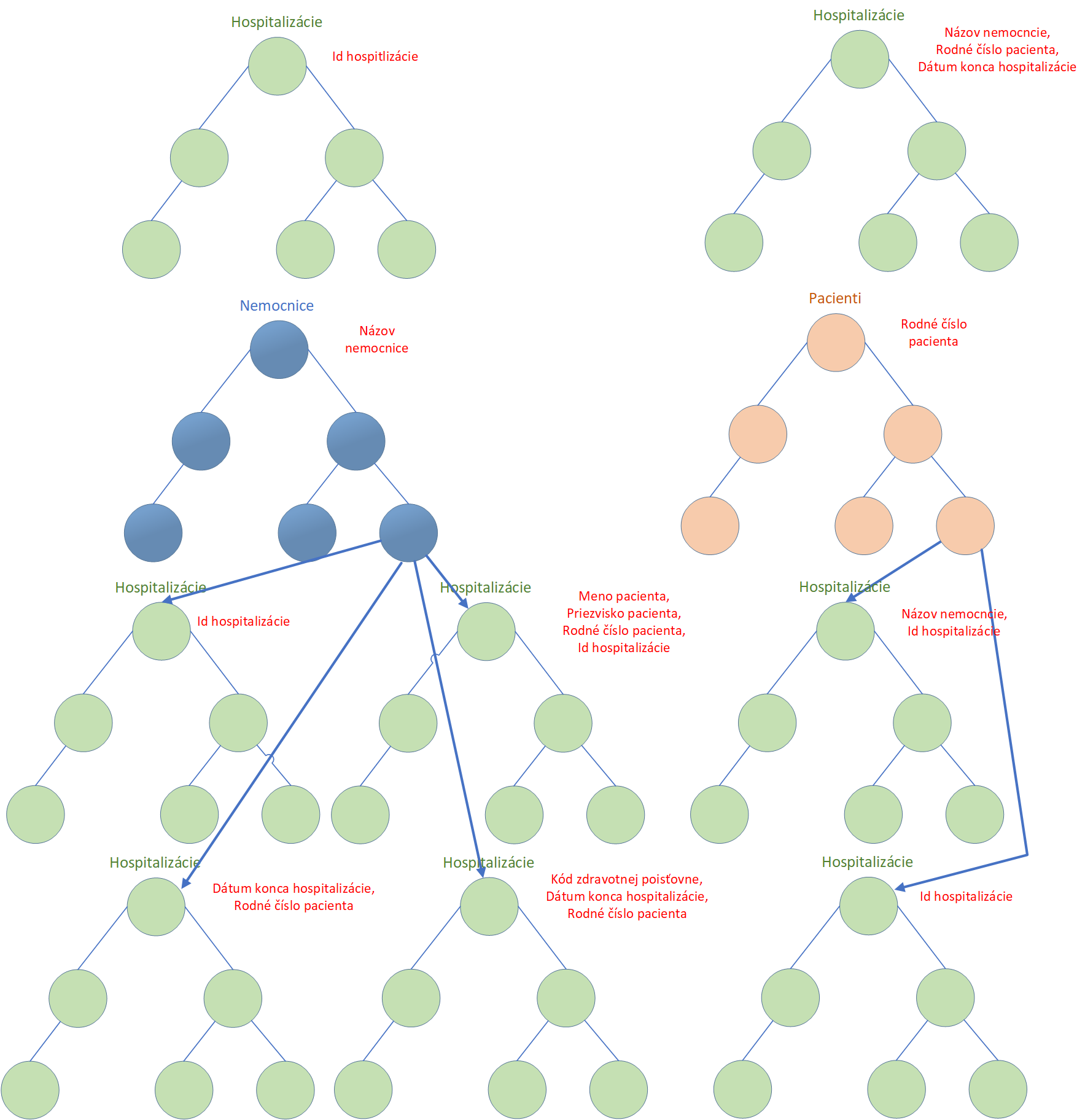
Na nasledujúcom obrázku sú zobrazené triedy, ktoré tvoria údajovú štruktúru vychádzajúcu z binárneho vyhľadávacieho stromu (BVS).

Trieda **BST\_Data –** predstavuje predka všetkých typov dátových tried, ktoré sa môžu nachádzať v BVS (Pacient, Nemocnica, Hospitalizácia, prípadne ďalšie triedy potrebné pre usporiadanie dát v určitom poradí).

Trieda **BST\_Node –** udržiava si referencie na svojich potomkov a predka. Taktiež sú v nej uložené dáta.

Trieda **BST** – zabezpečuje funkcionalitu BVS a obsahuje ďalšie pomocné metódy, napr. pre prejdenie celej štruktúry. Ukladá si referenciu na koreň stromu a aktuálny počet prvkov stromu.

# Návrh údajových štruktúr



V aplikácií sú štyri hlavné stromy. Prvá stromová štruktúra v sebe zaznamenáva všetky hospitalizácie. Tieto hospitalizácie sú vkladané podľa kľúča **id hospitalizácie** – automaticky inkrementovaná hodnota. Ďalšia štruktúra zaznamenáva referencie na všetky hospitalizácie zoradené podľa **názvu nemocnice**, **rodného čísla pacienta** a **dátumu konca hospitalizácie**. Strom nemocníc má ako kľúč **názov nemocnice** a obsahuje všetky nemocnice. Každá nemocnica si udržiava štyri ďalšie stromy hospitalizácií zotriedených podľa určitého kľúča. Buď podľa **id hospitalizácie**, alebo podľa **mena, priezviska, rodného čísla pacienta a id hospitalizácie**, alebo podľa **dátumu konca hospitalizácie** a **rodného čísla pacienta**, alebo **kódu zdravotnej poisťovne**, **dátumu konca hospitalizácie** a **rodného čísla pacienta**. Posledným hlavným strom je strom pacientov zoraďovaný podľa **rodných čísiel pacientov**. Tento strom si udržiava dva pomocné stromy hospitalizácií. Jeden je zotriedený podľa **názvu nemocnice** a **id hospitalizácie**, druhý je strom všetkých hospitalizácií daného pacienta zoradený len podľa **id hospitalizácie**.

Zoznam najčastejšie používaných značiek početnosti:

* **a** – počet hospitalizácií,
* **b** – počet nemocníc,
* **c** - počet pacientov,
* **d** – počet hospitalizácií v zadanej nemocnici,
* **e** – počet hospitalizácií v zadanej nemocnici, kde dátum konca hospitalizácie je neevidovaný,
* **f** – počet hospitalizácií pacienta

## Operácia – Vyhľadanie záznamov pacienta v nemocnici

Vyhľadám v strome pacientov požadovaného pacienta. Následne nájdem všetky hospitalizácie, kde sa zhoduje názov nemocnice so zadaným názvom nemocnice.

**c** - počet pacientov, **f** – počet hospitalizácií pacienta

Zložitosť: O(log(c) + log(f))

## Operácia - Vyhľadanie záznamov pacientov v nemocnici podľa mena a priezviska

Vyhľadám si v strome nemocníc zadanú nemocnicu. Následne pomocou intervalového vyhľadávania nájdem všetky záznamy, ktoré sa zhodujú so zadaným meno a priezviskom pacienta.

**b** – počet nemocníc, **d** – počet hospitalizácií v zadanej nemocnici

Zložitosť: O(log(b) + log(d))

## Operácia – Pridanie novej hospitalizácie

Novú hospitalizáciu vkladám do hlavného stromu hospitalizácií. Následne ukladám referenciu na novú hospitalizáciu do každého stromu. Jedinou výnimkou je hospitalizácia, ktorej dátum konca nie je zadaný, referenciu na takúto hospitalizáciu vkladám ešte aj do jedného ďalšieho stromu (zoradeného podľa dátumu konca hospitalizácie a rodného čísla pacienta).

**a** – počet hospitalizácií, **b** – počet nemocníc, **c** - počet pacientov, **d** – počet hospitalizácií v zadanej nemocnici, **e** – počet hospitalizácií v zadanej nemocnici, kde dátum konca hospitalizácie je neevidovaný, **f** – počet hospitalizácií pacienta

Zložitosť: O(2\*log(a) + log(b) + 3\*log(d) + log(e) + log(c) + 2\*log(f))

## Operácia – Ukončenie aktívnej hospitalizácie

Nájdem si v strome všetkých hospitalizácií zoradených podľa názvu nemocnice, rodného čísla pacienta a dátumu konca hospitalizácie hospitalizáciu, ktorú idem upraviť. Následne nájdem a odstránim v nemocnici zo stromu zoradeného podľa dátumu konca hospitalizácie a rodného čísla pacienta hospitalizáciu, ktorú ukončujem. Nakoniec zmením dátum konca hospitalizácie.

**a** – počet hospitalizácií, **e** – počet hospitalizácií v zadanej nemocnici, kde dátum konca hospitalizácie je neevidovaný

Zložitosť: O(log(a) + 2\*log(e))

## Operácia – Výpis pacientov z intervalu hospitalizácií

Nájdem si zvolenú nemocnicu. Uložím si do stromu všetky hospitalizácie, ktoré majú dátum konca väčší ako je zadané minimum (dátum od). Zo získaného stromu si následne do zoznamu uložím všetky hospitalizácie, ktoré majú dátum začiatku menší ako je maximum (dátum do). Následne si do finálneho stromu uložím všetkých pacientov, ktorí sa nachádzajú v získanom zozname hospitalizácií.

**b** – počet nemocníc, **d** – počet hospitalizácií v zadanej nemocnici, g – počet získaných prvkov po prvom intervalovom vyhľadávaní, h – počet získaných prvkov po druhom intervalovom vyhľadávaní

Zložitosť: O(log(b) + log(d) + log(g) + log(h))

## Operácia – Pridanie pacienta

Pridám pacienta do stromu pacientov.

**c** - počet pacientov

Zložitosť: O(log(c))

## Operácia – Tvorba podkladov pre účtovné oddelenie

Nájdem si zvolenú nemocnicu. Vykonám intervalové vyhľadávanie. Vytvorím si trojrozmerný zoznam, ktorý obsahuje všetky poisťovne, tie obsahujú zoznam všetkých dní v mesiaci a ten obsahuje všetky hospitalizácie vykonané v daný deň (i\*j). Vypočítam si pre každú poisťovňu počet dní hospitalizácií (h\*i). Následne už len vypíšem všetky údaje.

**b** – počet nemocníc, **d** – počet hospitalizácií v zadanej nemocnici, g – počet získaných prvkov po prvom intervalovom vyhľadávaní, h – počet získaných prvkov po druhom intervalovom vyhľadávaní, i – počet zdravotných poisťovní, j – počet dní v mesiaci

Zložitosť: O(log(b) + log(d) + log(g) + log(h) + i \* j + h \* i)

## Operácia – Výpis aktuálne hospitalizovaných pacientov

Nájdem si zvolenú nemocnicu a pomocou prehliadky InOrder vypíšem všetky dáta v strome hospitalizácií zoradených podľa dátumu konca hospitalizácie a rodného čísla pacienta.

**b** – počet nemocníc

Zložitosť: O(log(b))

## Operácia - Výpis aktuálne hospitalizovaných pacientov určitej poisťovne

Nájdem si zvolenú nemocnicu. Nájdem si pomocou intervalového vyhľadávania všetky aktuálne hospitalizácie zo zadanej poisťovne. Následne vypíšem všetkých pacientov.

**b** – počet nemocníc, **d** – počet hospitalizácií v zadanej nemocnici

Zložitosť: O(log(b) + log(d))

## Operácia - Výpis aktuálne hospitalizovaných pacientov určitej poisťovne zotriedený podľa rodného čísla

Nájdem si zvolenú nemocnicu. V nej si nájdem pomocou intervalového vyhľadávania všetky aktuálne hospitalizácie zo zadanej poisťovne. Následne vypíšem všetkých pacientov.

**b** – počet nemocníc, **d** – počet hospitalizácií v zadanej nemocnici

Zložitosť: O(log(b) + log(d))

## Operácia – Vyváženie všetkých stromových štruktúr

Pre každú stromovú štruktúru vykonám metódu vyvažovania.

**k** – počet stromových štruktúr, **l** – počet prvkov stromu, **m –** počet prvkov podstromu aktuálne vyvažovaného prvku

Zložitosť: O(k \* l \* m)

## Operácia – Pridanie nemocnice

Pridám nemocnicu do stromu nemocníc.

**b** - počet nemocníc

Zložitosť: O(log(b))

## Operácia – Výpis nemocníc zotriedený podľa názvu

Vypíšem všetky nemocnice zo stromu nemocníc pomocou prehliadky InOrder.

**b** - počet nemocníc

Zložitosť: O(b)

## Operácia – Zrušenie nemocnice

Nájdem si v strome nemocníc obe požadované nemocnice. Vložím všetky hospitalizácie do novej nemocnice. Nájdem si intervalovým vyhľadávaním všetky hospitalizácie zo zrušenej nemocnice a zmením im nemocnicu, v ktorej boli vykonané. Následne utriedim všetky stromové štruktúry, ktoré môžu byť vykonanou zmenou zasiahnuté.

**a –** počet hospitalizácií, **b** - počet nemocníc, **n** – počet prvkov rušenej nemocnice, **o** – počet prvkov novej nemocnice

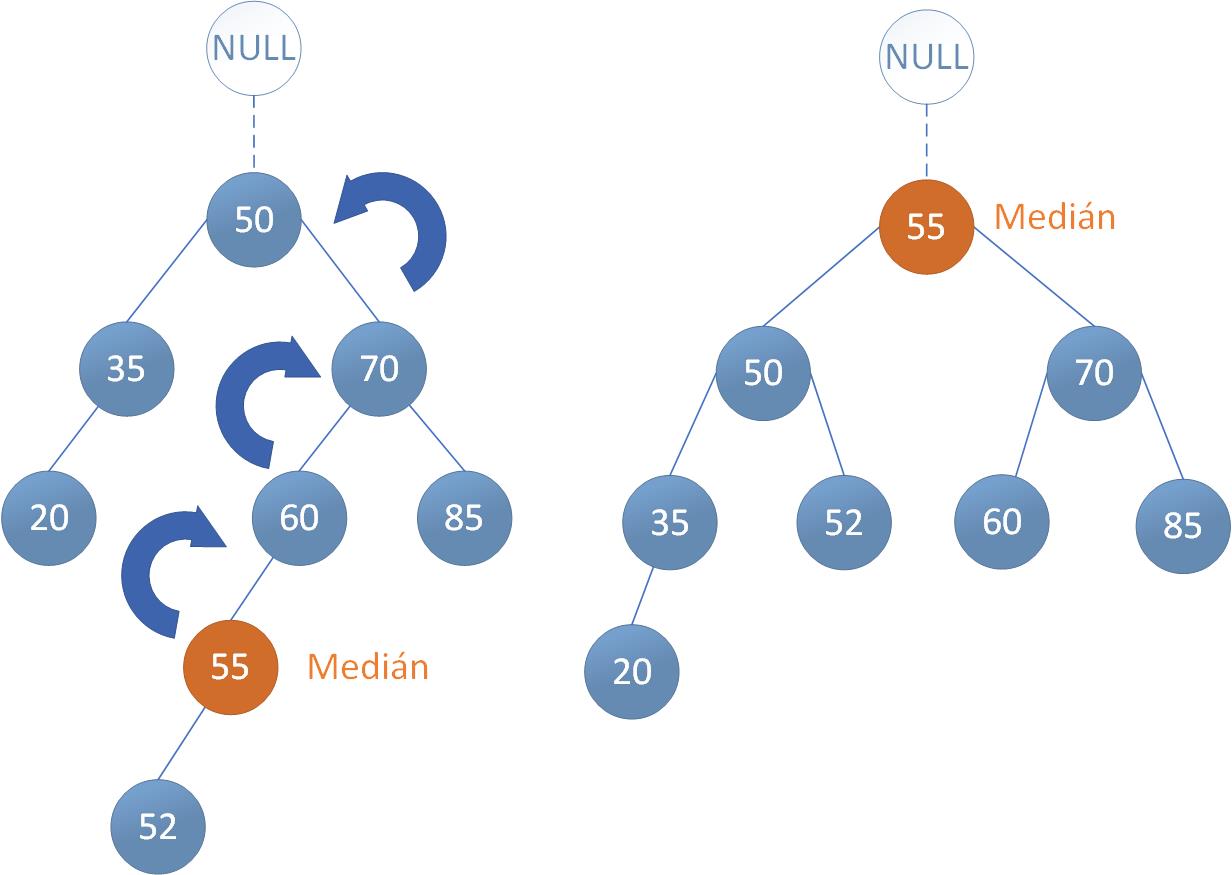
Zložitosť: O(log(b) + log(b) + n + n\*log(o) + log(a) + n)

# Spôsob využitia rotácií

## Vyvažovanie stromu

Postupne prechádzam všetky prvky stromu od koreňa v poradí LevelOrder. Z aktuálneho prvku stromu si získam InOrder zoznam prvkov celého jeho podstromu. Z tohto zoznamu zistím medián ako z indexu veľkosť InOrder zoznamu/2. Zistím si rodiča aktuálneho prvku. Následne vykonávam pravú/ľavú rotáciu nad mediánom pokiaľ sa rodič mediánu nerovná rodiču prvku, nad ktorým sme robili vyváženie.

Nižšie je uvedený postup použitých rotácií na vyváženie stromu podľa koreňa stromu. V tomto prípade rotujem medián dovtedy kým nie je jeho rodič rovný NULL.



## Samovyvažovací mechanizmus

Vložím prvok stromu štandarde ako v binárnom vyhľadávacom strome. Následne si zistím minimálnu možnú výšku celého stromu. Ak je úroveň pridaného prvku stromu menšia alebo rovná minimálnej možnej výške, tak nevykonávam žiadne rotácie. Ak nie, tak si zistím rodiča a starého rodiča pridaného prvku. Najskôr sa pokúsim vykonať pravú rotáciu nad ľavým synom koreňa stromu alebo ľavú rotáciu nad pravým synom koreňa stromu v prípade, že je výška pravého podstromu menšia aspoň o 1 vzhľadom na výšku ľavého podstromu, resp. je výška ľavého podstromu menšia aspoň o 1 vzhľadom na výšku pravého podstromu. Následne si zistím aktuálnu výšku podstromu prvku, ktorým chcem nahradiť vložený prvok. Potom vykonám buď ľavú a pravú / pravú a ľavú/ len ľavú/ len pravú rotáciu vzhľadom na pozíciu (ľavý syn alebo pravý syn) vloženého prvku, prípadne aj rodiča vloženého prvku. Ako posledné skontrolujem či sa zlepšila výška podstromu vzhľadom na predchádzajúcu výšku podstromu. Ak nie, tak vrátim podstrom do pôvodného stavu.

Nižšie je uvedený nákres ako vykonávam len pravú, ľavú a pravú rotáciu. Vykonávanie ostatných dvoch (pravú a ľavú, len ľavú) rotácií funguje obdobne.

