

# Dátové štruktúry a algoritmy

## **Zadanie č.4 (10 bodov)**

### **Úloha: Binárny strom**

Pomocou štruktúry binárny strom navrhnete počítačovú hru, ktorá formou konverzácie s počítačom umožní určiť hľadané zviera, ktoré si hráč práve myslí. Počítač kladie otázky hráčovi, ktorý odpovedá áno alebo nie.

### **Vstup programu**

Príklad:

*Tu sa zobrazí cele menu*

...

Vitaj v hre Animal.

Mysli na zviera a pokúsím sa ho uhádnuť ...

Žije to na súši (y/n)? y

Má krídla (y/n)? n

Je to slon (y/n)? y

Uhádol som (povedal program...)

Chcete hrať znova (y/n)? y

Mysli na zviera a pokúsím sa ho uhádnuť ...

Žije to na súši (y/n)? n

Je to veľryba (y/n)? n

Na aké zviera myslíš? žralok

Zadajte otázku tak, aby odpoveď bola

áno - pre žraloka

nie - pre veľrybu

→ Je chladnokrvný

Mali ste šťastie (povedal program...)

Mysli na zviera a pokúsím sa ho uhádnuť ...

Žije to na súši (y/n)? n

Je chladnokrvný (y/n)? y

Je to žralok (y/n)? n

Na aké zviera myslíš? elektrický úhor

Zadajte otázku tak, aby odpoveď bola

áno - pre žraloka

nie - pre veľrybu

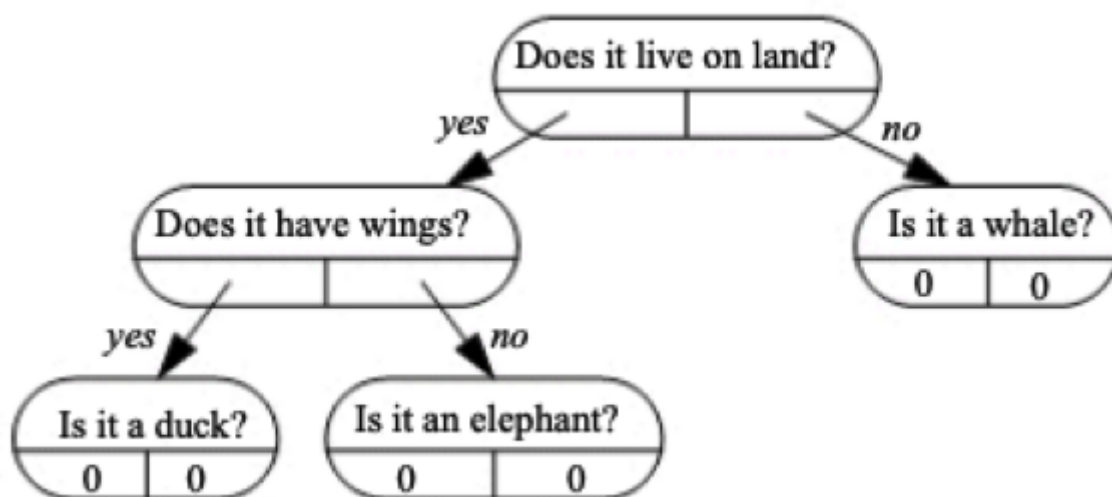
→ Spíňa to šokujúci zážitok

Mali ste šťastie ...

Chcete hrať znova (y/n)? n

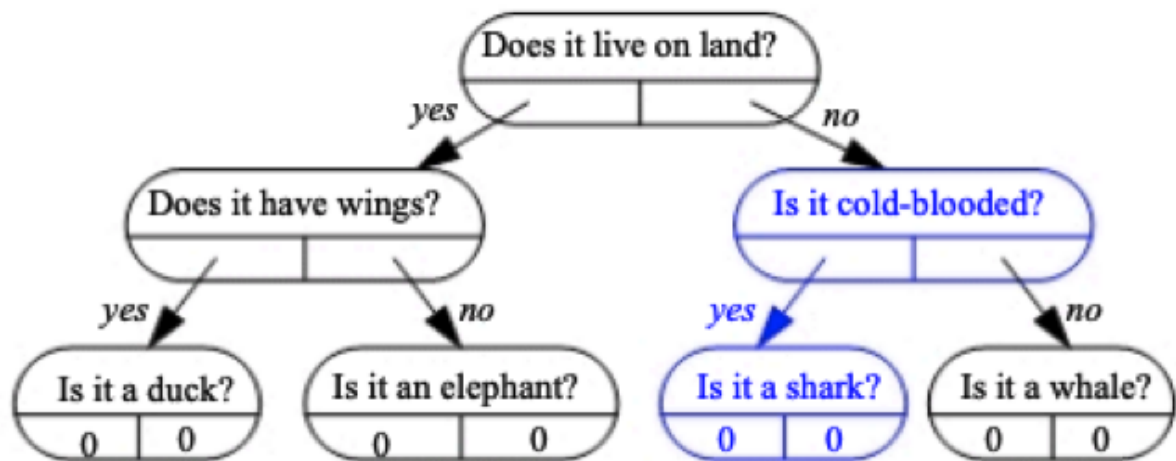
## Zadanie

Program hrá hru vytvorením špeciálneho stromu, ktorý bude slúžiť ako jeho vedomostná základňa, obsahujúce otázky a zvieratá, ktoré „pozná“. Napríklad, keď bol program prvý krát spustený tak „vedel“ iba tri zvieratá: kačica, slon a veľryba. Jeho vedomostná základňa bola pôvodne štruktúrovaná takto

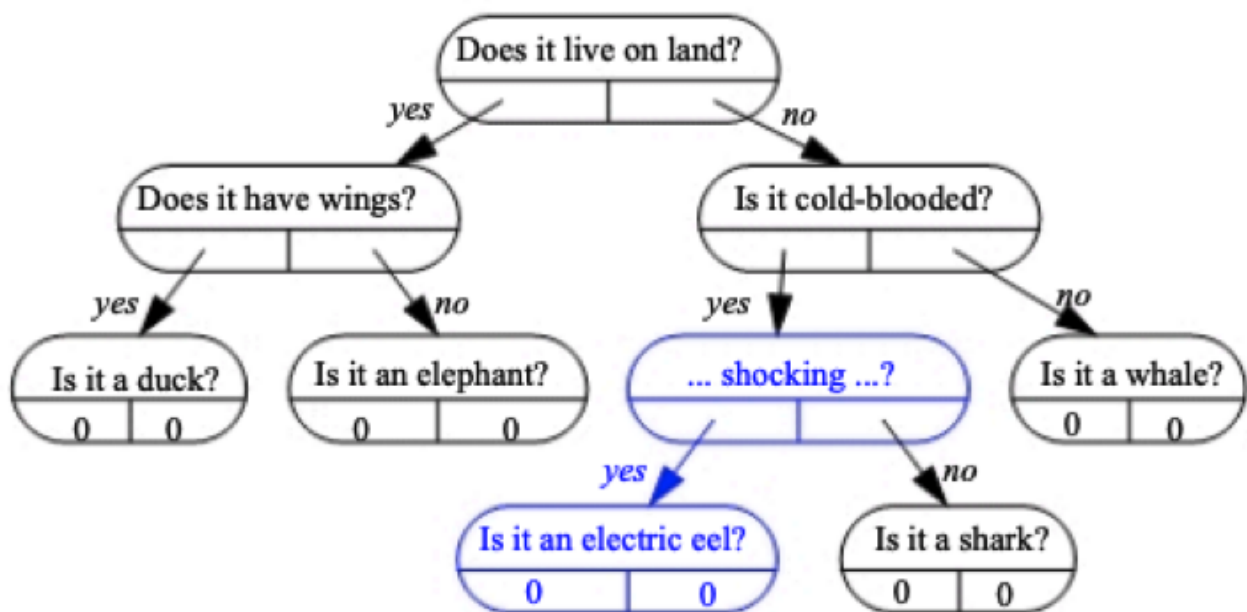


Takáto štruktúra sa nazýva rozhodovací strom - každý uzol v strome obsahuje informácie potrebné na rozhodnutie áno-nie medzi jej dvoma pod stromami. Hádanka začína položením najvyššej otázky a na základe odpovede hráča nasleduje vetvu áno alebo nie vetvu na nasledujúcu otázku. Hádateľ pokračuje v tomto procese a zostupuje stromom, až kým nedosiahne uzol listu, v tom prípade položí svoju poslednú otázku a uhádne zviera uložené v tomto uzle. Implementáciou takého stromu ako prepojenej štruktúry možno ľahko vložiť (alebo vymazať) nové uzly. V prvej hre teda program „mal šťastie“, pretože sa stalo, že zviera, na ktoré hráč myslel (slon), bolo jedným z troch, o ktorých vedel. Avšak v druhej hre užívateľ myslel na žraloka - zviera, o ktorom program

nevedel. Na základe informácií poskytnutých hráčom sa program naučil rozlišovať medzi veľrybou a žralokom vytvorením a vložení uzlov, ktoré jej umožnia rozlíšiť medzi týmito dvoma zvieratami:



Podobne sa v nasledujúcej hre program naučil rozlišovať medzi žralokom a úhorom:



Program sa tak „učí“ tým, že rozširuje svoj rozhodovací strom vždy, keď narazí na zviera, ktoré tam nie je. Aby sa predišlo „zabudnutiu“ na to, čo sa „naučil“, **deštruktor BinTree zapíše dáta stromu do súboru**, keď sa program ukončí, a **konštruktor BinTree prečíta z toho istého súboru**, keď sa program znova spustí. Tento jednoduchý mechanizmus umožňuje programu „zapamätať si“, čo sa „naučil“ v predchádzajúcich hrách. V priebehu času sa takýto program môže stať dosť dobrým pri identifikácii zvierat na základe charakteristík, ktoré sa učia.

Formát vstupného súboru si zvolte tak, aby sa vám s nim dobre pracovalo v rámci binárneho stromu.

Ďalej program umožní používateľovi interaktívne používať operácie údajového typu binárny strom, ktorý v príklade používate. Zahrňte nasledujúce možnosti v rámci interakcie s užívateľom:

- a. Vytvorenie prázdneho stromu
- b. Vytvorenie ľavého nasledovníka
- c. Vytvorenie pravého nasledovníka
- d. Nastavenie na root
- e. Nastavenie na ľavé dieťa
- f. Nastavenie na pravé dieťa
- g. Nastavenie aktívneho prvku na rodiča
- h. Vypíše hodnotu aktívneho vrcholu a jeho hodnoty ľavý a pravý (prípadne null ak nemajú dáta)
- i. Spustenie hry
- x. Ukonči program.

Tento program opakovane vypíše svoju ponuku (menu) pre užívateľa na obrazovku pri zadávaní možností a-i,x.

Možnosti a-g slúžia na manuálne pridávanie dát počas behu programu.

Vstupom do tohto programu je jedna z hore uvedených možností a-h, x zadaná užívateľom cez klávesnicu v rámci nekonečnej slučky až kým užívateľ neukončí program cez voľbu 'x'. Program po načítaní danej voľby prevedie príslušnú operáciu a vypíše svoj výstup na monitor.

Vyžaduje sa dôsledné používanie tried v C++ a hlavičkového súboru pre vstup z klávesnice resp. pre výstup do súboru. Hlavnou dátovou štruktúrou programu bude ADT binárny strom (binTree). Môžete ho využiť napr. tak ako bol špecifikovaný na prednáškach z predmetu. Jeho špecifikácia sa bude nachádzať v hlavičkovom súbore: BinTree.h a jeho implementácia v súbore: BinTree.cpp. Navyše dátová (private) časť binárneho stromu bude implementovaná použitím smerníkov a dynamickej pamäte. Musia sa výhradne používať iba operácie binárneho stromu špecifikované v súbore binTree.h pri práci s binárnym stromom. Vyžaduje sa správny štýl programovania a v prípade potreby aj primeraný komentár, resp. príslušné chybové hlásenie. Program bude navrhnutý ako čisto objektovo orientovaný a musí byť funkčný, a zároveň riadne otestovaný v rámci svojho main-programu t.j. testovacieho drajveru main.cpp.

## Odovzdanie

1. Názvy premenných, funkcií, tried, ... píšeme v angličtine (příklad get\_top\_panel() - správne, získaj\_horny\_panel() - nesprávne).
2. Zdrojový kód programu (ošetrite pre rôzne nevalidné vstupy – vstup z terminalu, alebo neexistujúceho súboru), tak aby program nespadol).
3. Spustiteľný kód programu (hodnotí sa len program, ktorý je možné spustiť).

4. Upload potrebných súborov (**.cpp, .txt, ...**) do vyhradeného assignement priečinka v prostredí MS Teams.

**Denní študenti** odovzdávajú podľa pokynov cvičiaceho (miesto odovzdania v MS Teams).

Za každý oneskorený deň odovzdania do MS Teams bude penalizované -1b. Zadanie nie je potrebné prezentovať v deň odovzdania. Je potrebné odovzdať a prezentovať zadanie aj keby bolo za 0b.

**Externí študenti** odovzdávajú podľa pokynov prednášajúceho, prípadne mailom na

