NTI/PJPA - Programovací jazyk Python (2023)

My courses
ITE/CITE - Číslicová technika (2022)
ITE/EDK - Elektronická dokumentace (2022)
ITE/MTLB - Výpočty, simulace a vizualizace Matlab (2022)
ITE/SGI - Signály a informace (2023)
ITE/ZKO - Základy konstruování (2023)
KAP/ULA - Úvod do lin. algebry a diskrétní mat. (2022)
MTI/ALG1 - Algoritmizace a programování 1 (2022)
Samostatné úlohy z předmětu Algoritmizace a programování 1
MTI/ALG2 - Algoritmizace a programování 2 (2022)
MTI/CIP - Číslicové počítače (2023)
MTI/DBS - Databázové systémy (2023)
MTI/PRJC - Programování v jazyce C/C++ (2023)
MTI/STIN - Softwarové inženýrství (2023)
MTI/UDI - Úvod do inženýrství (2022)
MTI/VAPW - Vývoj aplikací pro Windows (2023)
NTI/ALD - Algoritmizace a datové struktury (2023)
NTI/OPS - Operační systémy (2023)
NTI/PJPA - Programovací jazyk Python (2023)
Počítačové sítě
NTI/PST - Počítačové sítě (2022)
NTI/SH - Úvod do Shellu (2022)
NTI/TWS - Tvorba WWW stránek (2023)
NTI/USA - Úvod do statistické analýzy (2023)

Navigation
Dashboard
Site home
Site pages
Courses enrollment (STAG)
Courses unenrolment
Propojení se STAGem
My courses
ITE/CITE - Číslicová technika (2022)
ITE/EDK - Elektronická dokumentace (2022)
ITE/MTLB - Výpočty, simulace a vizualizace Matlab ...
ITE/SGI - Signály a informace (2023)
ITE/ZKO - Základy konstruování (2023)
KAP/ULA - Úvod do lin. algebry a diskrétní mat. (2...
MTI/ALG1 - Algoritmizace a programování 1 (2022)
Samostatné úlohy z předmětu Algoritmizace a progra...
MTI/ALG2 - Algoritmizace a programování 2 (2022)
MTI/CIP - Číslicové počítače (2023)
More...
Courses
FM
DFM
ITE
MTI
NTI
2023/24
NTI/ADA - Algoritmy a datové struktury (2023)
NTI/AFI-P - Automaty a formální jazyky (2023)
NTI/ALD - Algoritmizace a datové struktury (2023)
NTI/AMP - Alternativní metody programování (2023)
NTI/APMO - Aplikace počítačových modelů (2023)
NTI/ARP - Architektura počítačů (2023)
NTI/CFD - Výpočetní mechanika tekutin (2023)
NTI/DPG - Distribuované programování (2023)
NTI/EMM - Experimentální metody v mechanice (2023)
NTI/JPD - Jazyky pro popis dat (2023)
NTI/KAS - Kybernetická bezpečnost a šifrování (2023)
NTI/PJPA - Programovací jazyk Python (2023)
Participants
Competencies
Grades
Programovací jazyk Python - PJPA LS 2024
Úkoly a cvičení
úkol 0 - přihlaste se na gitlab.tul.cz
úkol 1. - první program
úkol 2. - konvexní čtyřúhelník
úkol 3. - transformace dat
úkol 4. - Caesarova šifra
úkol 5. - algoritmizace problému
úkol 6. - regulární výrazy
úkol 7. - zpracování JSON a HTML dat
úkol 8. - Poker (starší zkoušková otázka)
úkol 9. - Binární Vyhledávací Strom
úkol 10. - Cenzor (starší zkoušková otázka)
účast na přednáškách
1. Ponořme se do Pythonu - úvod do předmětu
2. proměnné a konstanty
3. struktuované datové typy - kolekce a sekvence
4. další vlastnosti jazyka
5. testování kódu
6. standardní textové formáty a jejich zpracování
7. funkce a jejich pokročilé využití
8. tvorba vlastních typů, principy OOP
9. tvorba aplikací s CLI (command line interface)
10. další moduly standardní knihovny jazyka Python
11. Výkonnost Python programů
Topic 14
2022/23
2021/22
2020/21
2019/20
2018/19
Aplikace GIS
Diplomové a bakalářské práce 2021/22
Geografické informační systémy
Kopetschke DP, BP, PRO, PRJ 2016/17
Počítačové sítě
RSS
Kurzy mimo STAG
Bezpečnost práce na elektrickém zařízení v laborat...
Admission Test Mechatronics 2024
Samostatný elektrotechnik pro elektromagnetickou k...
Fraktální geometrie
Hodnocení kvality výuky (BS-IT 2021/22)
Virtuální setkání akademické obce FM
Příprava na přijímačky z informatiky
Příprava na přijímačky z matematiky
Studentská konference Fakulty mechatroniky
MTI/CSHARP - T1
Podnikový informační systém SAP
NÁVODY, MANUÁLY
Další podpůrné materiály
Kurzy pro zaměstnance TUL
Externí kurzy
FA
FE
FP

úkol 9. - Binární Vyhledávací Strom

Opened: Monday, 15 April 2024, 12:00 AM
Due: Tuesday, 30 April 2024, 11:59 PM

V tomto úkolu budete implemenetovat Binární vyhledávací strom (dále jen BVS). Tato datová struktura se anglicky označuje jako Binary Search Tree - BST.

Vaším úkolem je naprogramovat BVS s využitím vlastních datových typů = tříd. Budete potřebovat minimálně jednu třídu, která bude reprezentovat samotný strom a druhou třídu, která bude reprezentovat uzel/node ve stromu.

Pro zjednodušení nebudeme v tomto cvičení řešit optimální vyváženost stromu. Stačí implementovat obyčejný BVS. Jelikož se jedná o vyhledávací strom, je nutné určit podle kterého pravidla se budou prvky řadit.

Pro tuto úlohu tedy předpokládejte, že <= (menší nebo rovno) prvky se budou vkládat na levo od uziu s kterým je porovnáván.

Aby bylo porovnávání možné implementovat, bude strom pracovat vždy se stejným datovým typem. Datový typ stromu se vytvoří s prvním vloženým uziem. V případě, kdy do stromu je vložen objekt typu integer, strom pracuje dále pouze s objekty typu integer. V případě, že půjde o řetězec, bude dále možné vkládat do stromu pouze řetězce a tak dále.

Při vkládání druhého a dalších prvků, je potřeba hlídat typ vkládaného objektu a případně vyvolat TypeError.

Prohledávání stromu je možné řešit jak do šířky, tak do hloubky. Prohledávání do šířky budete potřebovat pro implementaci metody __str__ - viz bod 5. níže. Prohledávání do hloubky pak můžete použít pro hledání prvku při operaci in (bod 3.)

Hotový BVS musí umět následující operace:

1. Přidat nový object (add)
2. Kontrolovat zda je přidávaný objekt správného typu
3. Zjistit zda-li se hodnota vyskytuje v BVS (contains, operátor in).
4. Získání počtu objektů ve stromu (len(bvs))
5. Textová reprezentace stromu - metoda vrátí string s hodnoty uzlů stromu vypsaných metodou do šířky. Implenentace metody __str__, použití str(bvs), print(bvs).
6. Možnost iterativního procházení stromu. Implenetace metody __iter__ a __next__, použití cyklus for, funkce next().

Výsledené řešení splňovat také následující podmínky:

- musí jít o testovatelný kód - vše tedy musí být buť ve funkcích, nebo uzavřeno pod __name__ == '__main__'
- modul musí být v adresáři cv09 a musí se jmenovat bst.py. Hodnocené třída implementující požadovné funkce se musí jmenovat BTS. Opět můžete využít šablonu ze vzorového repozitáře.
- pro řešení vám musí stačit standardní instalace pythonu (bez balíčků třetích stran)
- výsledný kód musí při testu programem PyLint se standardním nastavením získat alespoň 8 bodů. Pokud dosáhnete horšího hodnocení, dostanete za každý bod pod 8 mínus jeden bod z celkového hodnocení úkolu.

Add submission

Submission status

Submission status	No submissions have been made yet
Grading status	Graded
Time remaining	Assignment is overdue by: 20 days 13 hours
Last modified	-
Submission comments	Comments (0)

Feedback

Grade	15.00 / 15.00
Graded on	Tuesday, 21 May 2024, 10:11 AM
Graded by	LM Lukáš Mázl
Feedback comments	V pořádku

úkol 8. - Poker (starší zkoušková otázka) Jump to... úkol 10. - Cenzor (starší zkoušková otázka)