Предмет: Информатика

Одделение: 7

Наставна тема: Програмирање со стандарден структурен јазик

Наставна содржина: Структура за избор од две можности

Час: 27

Наставник: Слаѓана Елезовиќ

Цел на часот: совладување на структурата за одлучување во програмскиот јазик Python

4. Структура за одлучување

Во секојдневниот живот често донесуваме различни одлуки, на пример: „ако врне дожд ќе понесам чадор, ако не врне тогаш нема да го земам“ или „ако имам грешка во задачата истата ќе ја поправам, ако немам тогаш ќе го предадам тестот“.

Во алгоритамот постои место каде се одлучува која наредба ќе се изврши акоја нема. Во БЛОК ДИЈАГРАМОТ постои геометриска форма со која се дефинира блокот за одлука и тој изгледа како ромб.

ДА

Блок за одлука

Во ромбот се пишува условот.

НЕ

Алгоритмот( или текот на програмата кога ќе го пишуваме кодот) по условот продолжува да се движи во еден од двата можни правци. Ако е исполнет условот поставен во ромбот, тогаш текот на програмата продолжува со одредени наредбите што следат во правец на стрелката „ДА“, додека ако условот во ромбот не е исполнет тогаш следуват други наредби во правец на трелката „НЕ“.

Таквиот алгоритам го нарекуваме **алгоритам со структура за одлучување**

**Пример:** определи дали еден број внесен преку тастатура е позитивен или негативен

**Блок дијаграмот**

**Алгоритамот**

**Програмскиот код**

Бројот **а** е негативен

влез на податокот  
 а

a>0

Бројот а е позитивен

Input (a)

If a>0:

Print(‘Бројот а е позитивен’)

else:

print(‘бројот а е негативен’)

не

да

Влез на бројот (a)

Ако бројот а>0, тогаш бројот е позитивен

Ако бројот а<0, тогаш бројот е негативен

Од покажаниот пример можеме да забележиме дека синтаксата или инструкциите кои ни се неопходни за структурата за одлучување се:

Значи за да ја употребиме структурата за одлучување треба да запазиме правило кое глеси : по наредбата if доаѓа условот па доаѓа знакот „ : “. Истото важи и за вториот дел те.наредбата else, по која следува знакот „:“

If услов :

наредба 1

else:

наредба2

Во овој дел веќе се наметнува потребата од прикажување на таблицата со математичките операции и нивните соодветни знаци те. операции во програмскиот јазик Python

Ако се прашувате зошто математичкиот знак за еднаквост во програмскиот јазик Python го пишуваме со == а не едноставно со =, тоа е затоа што во програмскиот јазик знакот = се употребува за ДОДЕЛУВАЊЕ(придружување) некоја вредност на променливата. Значи ако сакаме да внесеме, пример, вредност 5 во променливата а пишуваме а=5, но ако сакаме во некој услов да провериме дали променливата а има вредност 5, тогаш пишуваме   
**if a==5:**

|  |  |
| --- | --- |
| Математика | Python |
| < (помало) | < |
| > (поголемо) | > |
| = (еднакво) | == |
| ≤ (помало и еднакво) | <= |
| (поголемо и еднакво) | >= |
| ≠ (различно од) | != |

Наредба на одлука...**дали** променливата а има вредност 5

да

не

а==50

Вежба: Да се пресмета периметар на квадрат со страна а. Ако а<=0 тогаш програмата треба да врати одговор на екран „страната на квадратот не може да биде негативна“

Од математика знаеме дека периметарот на квадрат се определува со формулата:

L=а+а+а+а=4\*a каде а е страната на квадратот а \* е знак за множење

а

а

а

а

**Блок дијаграмот**

**Алгоритамот**

Внеси ја страната на квадратот (a)

Ако бројот а>0, тогаш L=4\*а

Ако бројот а<=0, тогаш испечати „Страната на квадратот не може да биде негативна“

Периметарот на квадратот е : L

Страната на квадратот не може да биде негативна

не

да

L=4\*а

a>0

влез на страната  
 а

е блок за пресметување

Да се потсетиме блокот

Со првата наредба print(‘Внеси ....., го испишуваме на екран текстот “Внеси ја страната на квадратот“

Со втората наредба a=input() внесуваме вредност во променливата а.Таа вредност ја внесува корисникот на нашиот програм.

Со наредбата a=int(a) се дефинира променливата а да биде од типот integer, те. да може во неа да се чуваат цели броеви

Со наредбата **if a>0 : се дефинира условот кој проверува дали страната а >0. Ако тој услов е исполнет следуваат наредбите за пресметување на периметарот на квадратот**

**Програмскиот код**

print(‘Внеси ја страната на квадратот:’)

a=input()

a=int(a)

if a>0 **:**

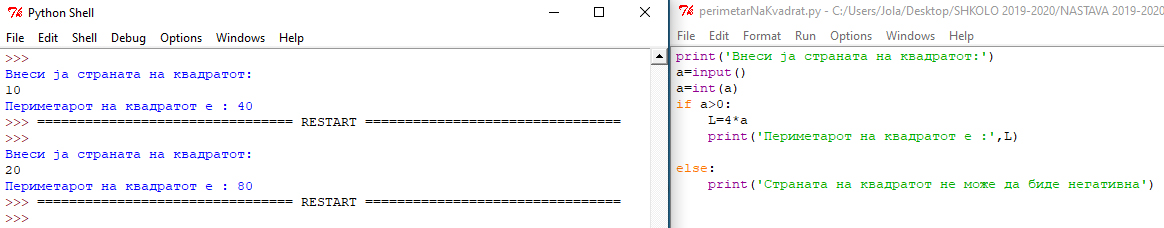
**L=4\*a**

**print(‘Периметарот на квадратот е :’,L)**

**else:**

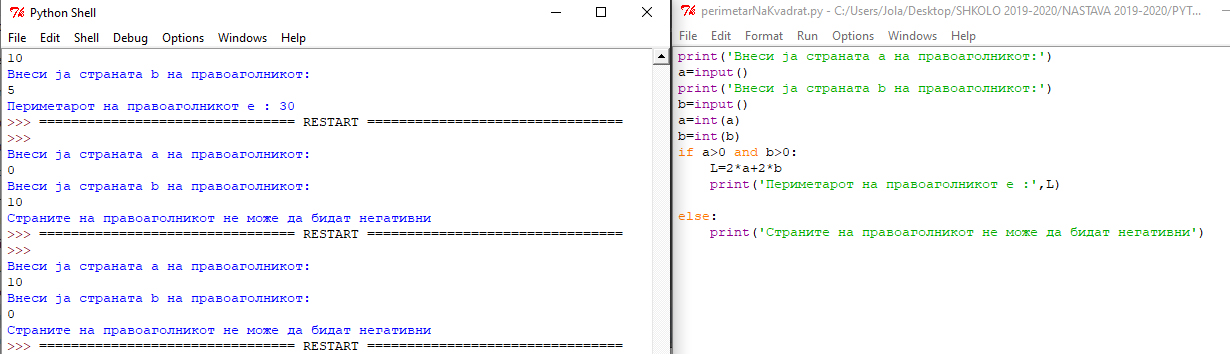
**print(‘Страната на квадратот не може да биде нула или негативна’)**

Со наредбата **else: се дефинира наредбата која следува ако условот “ if ” не е исполнет**



Вежба2:

Најди го периметарот на правоаголник со страни а и b. При што страните а и b треба да видат позитивни цели броеви.

Во овој дел употребивме споредбен израз употребувајќи логички оператор “**И“ кој проверува дали страната а>0 И дали страната b>0. Сега ќе го објасниме операторот И(AND).**

Оператор AND

**ОПЕРАТОР “ AND “**

|  |  |  |
| --- | --- | --- |
| исказ 1 | исказ 2 | резултат |
| вистина | вистина | вистина |
| вистина | невистина | невистина |
| невистина | вистина | невистина |
| невистина | невистина | невистина |

да појасниме: воколку првиот исказ е вистинит и вториот исказ е вистинит, тогаш целиот израз е вистинит. Пример: Ако имам пари(исказ 1 е вистинит) и ако продавницата работи(исказ 2 е вистинит), тогаш ќе купам намирници(резултат). А во нашиот конкретен пример ако страните а и b се поголеми од 0, тогаш можеме да ја пресметаме плоштината на правоаголникот. Ако пак една од тие две страни е помала од 0, тоа значи дека еден од исказите не е вистинит или можеби и двата искази не се вистинити, тогаш треба да екранот да ни се појави информација“Страните на правоаголникот не можат да бидат негативни“ .

Ако имам пари(исказ 1 е вистинит) и ако продавницата работи(исказ 2 е вистинит), тогаш ќе купам намирници(резултат)

Ако имам пари (исказ 1 е вистинит) **И** продавницата не работи(исказ 2 не е вистинит), **тогаш** не можам да купам намирници(резултат)

Ако немам пари(исказ 1 не е вистинит) **И** продавницата работи (исказ 2 е вистинит), **тогаш** не можам да купам намирници(резултат)

Ако немам пари(исказ 1 не е вистинит) **И** продавницата не работи(исказ 2 не е вистинит), **тогаш** не можам да купам намирници (резултат)

Задача за вас е да ги решите двата примери со структура за одлучување

За оние кои сакаат да пробаат нешто повеќе :

Задача+: Определи дали внесенит број n е парен или непарен.  
Упатство: проверуваш дали внесениот број n %2 дава остаток 1 или 0, затоа што операцијата % го дава остатокот од делењето на бројот n со бројот2.

Овде ви ја давам и табелата со аритметичките операции во Python

|  |  |  |  |
| --- | --- | --- | --- |
| Аритметичка операција | знак | Операција(пример) | резултат |
| Собирање | + | 2+3 | 5 |
| Одземање | - | 15-6 | 9 |
| Множење | \* | 2\*6 | 12 |
| Делење | / | 45/2  45/5 | 22.5  9.0 |
| Целоброен количник | // | 7/3  14/3 | 2  4 |
| Остаток при делење | % | 14%3  7%2 | 2  1 |

Каде реални броеви се сите не цели броеви, пример 2,5 или 4,7....но во програмскиот јазик Python децималната запирка се заменува со точка, па така горе наведените броеви ќе бидат 2.5 или 4.7