Шешім қабылдау құрылымдары. Шартты нұсқаулығы , if , if-else , if-elif-else.

Шешім қабылдау – ақпарат алмасу сияқты кез келген басќару қызметінің құрамдас бөлігі. Шешім қабылдау қажеттігі басқарушының алға қойған маќсатының қалыптасуы мен сол мақсатқа жетуіндегі іс-қимылын түгелдей қамтиды

Шартты (нұсқаулық) оператор:

if шарт:

1 нұсқаулар блогы

else:

2 нұсқаулар блогы

Шартты оператор келесі алгоритм бойынша жұмыс істейді:

Алдымен IF қызметтік сөзінен кейін орналасқан логикалық өрнектің (шарттың) мәні есептеледі. Егер оның нәтижесі ақиқат болса, жаңа қатарда 4 бос орынмен басталатын 1 нұсқаулар блогы орындалады, егер нәтижесі жалған болса, онда ELSE сөзінен кейін орналасқан 2 нұсқаулар блогы орындалады.

Нұсқаулар блогы қажетті операторлар санынан тұруы мүмкін, бірақ олардың барлығы міндетті түрде 4 бос орыннан кейін жазылуы керек.

1 мысал. Берілген екі санның үлкенін экранға шығару.

x= int(input('Введите первое число'))

y= int(input('Введите второе число'))

if x > y:

print('Большее число', x)

else:

print('Большее число', y)

Шартты нұсқауда else сөзі мен келесі блок болмауы да мүмкін. Мұндай нұсқау толық емес тармақталу деп аталады. Мысалы, егер:

2 мысал. Пернетақтадан х саны еншізіледі және біз оны абсолюттік өлшемге, осы санның модуліне алмастырғымыз келеді. Мұны былай жасауға болады: егер сан теріс болса, «унарлық минус» деген амалды қолдануымыз, ал егер сан оң болса, онда ештеңе өзгертпестен шығаруымыз керек.

x = int(input())

if x < 0:

x = -x

print(x)

Бұл мысалда х айнымалыға –x мәні меншіктеледі, бірақ тек x<0 болғанда ғана. Ал print(x) нұсқауы тексерілетін шартқа тәуелсіз, әрқашанда орындалады.

2 мысал: Пайдаланушы мектептегі сынып нөмірін енгізеді, бағдарлама оның қай сыныпқа жататыны туралы хабарлама шығарады.

n=int(input('Введите номер класса'))

if n<4:

print('Младшие классы')

else:

if n<9:

print('Средние классы')

else:

print('Старшие классы')

Салыстыру операторлары

Әдетте, тексерілетін шарт ретінде төмендегі салыстыру операторларының бірін есептеу нәтижесі қолданылады:

< кіші — шарт дұрыс, егер бірінші операнд екіншісінен кіші болса.

> үлкен — шарт дұрыс, егер бірінші операнд екіншісінен үлкен болса.

<= кіші немесе тең.

>= үлкен немесе тең.

== теңдік. Шарт дұрыс, егер екі операнд тең болса.

!= теңсіздік. Шарт дұрыс, егер екі операнд тең болмаса.

Мысалы, (x\*x < 1000) шарты «x\*x мәні 1000 нан кіші» дегенді, ал (2 \* x != y) шарты «х айнымалысының мәні у айнымалының мәніне тең емес» дегенді білдіреді.

Питонда салыстыру операторларын тізбекке біріктіруге (көптеген басқа да бағдарламалау тілдеріне қарағанда, мұнда ол үшін логикалық байланыстарды қолдану керек) болады, мысалы, a == b == c немесе 1 <= x <= 10.

Сонда 2 мысалды дұрысырақ жазуға болатын еді :

n=int(input('Введите номер класса'))

if 1<=n<=4:

print('Младшие классы')

else:

if 4print('Средняя школа')

else:

print('Старшие классы')

Салыстыру операторлары арнайы bool логикалық типінің мәнін қайтарады. Логикалық типтің мәндері екі мәндердің бірін қабылдауы мүмкін: True (ақиқат) немесе False (жалған). Егер логикалық True мәнін int типіне өзгертсек, онда 1 болады, ал егер False болса, онда 0 береді. Кері жағдайда 0 саны False өзгереді, ал кез келген нөл емес сан True өзгереді. Str типін bool типіне кері өзгерту кезінде, бос қатар False, ал кез келген бос емес қатар True мәніне өзгереді.

Тасбақа графикасының позициясын , түсін , бұрышын анықтау.

Жылы компьютерлік графика, тасбақа графикасы болып табылады векторлық графика туысқанды қолдану меңзер («тасбақа«) а Декарттық жазықтық. Тасбақа графикасы Логотиптің бағдарламалау тілі.

Тасбақаның үш қасиеті бар: орналасу орны, бағдарлау (немесе бағыт) және қалам. Қаламның да атрибуттары бар: түсі, ені және қосу / өшіру күйі.

Тасбақа өз орнына қатысты командалармен қозғалады, мысалы «10 бос орын алға жылжу» және «солға 90 градусқа бұрылу». Тасбақа көтеретін қаламды оны қосу, түсін немесе енін орнату арқылы басқаруға болады. Студент тасбақаның қозғалысын, егер олар тасбақа болса, не істейтіндерін елестету арқылы түсіне алады (және болжап, дәлелдей алады). Сеймур Паперт бұл «дене синтоникалық» ойлау деп атады.

Толық тасбақа графикалық жүйесі басқару ағыны, процедуралар мен рекурсияны қажет етеді: көптеген тасбақаларды сызу бағдарламалары орындалмайды. Осы блоктардан квадраттар, үшбұрыштар, шеңберлер және басқа құрама фигуралар сияқты күрделі пішіндер жасауға болады. Тасбақа графикасының идеясы, мысалы, а Линденмайер жүйесі генерациялау үшін фракталдар.

Тасбақа графикасы көбінесе Логотиптің бағдарламалау тілі.[2] Сеймур Паперт өзінің нұсқасын қолдау үшін 1960 жылдардың соңында Логотипке тасбақа графикасына қолдау көрсетті тасбақа робот, роботтың корпусына орнатылған немесе оған бекітілген кішкене тартылатын қаламды қолданып, оған берілген сызу функцияларын орындауға арналған, пайдаланушының жұмыс бекетінен басқарылатын қарапайым робот. Тасбақаның геометриясы (х,ж) бағытталған Декарттық геометрия, бірінші кезекте векторсияқты координаттық-адрестік жүйелермен салыстырғанда (яғни бастапқы нүктеден салыстырмалы бағыт пен қашықтық) негізделген PostScript. Тәжірибелік мәселе ретінде дәстүрлі модельдің орнына тасбақа геометриясын қолдану тасбақа роботының нақты қозғалыс логикасын имитациялайды. Тасбақа дәстүрлі түрде және көбінесе кескіндемелік түрде үшбұрыш немесе тасбақаның белгішесі түрінде ұсынылады (бірақ оны кез-келген белгішемен көрсетуге болады).

Бүгін Python бағдарламалау тілінің стандартты кітапханасында Turtle графикалық модулі бар.[3] Логотиптің алдыңғы нұсқасы сияқты, Python тасбақасын енгізу бағдарламашыларға екі өлшемді кеңістікте бір немесе бірнеше тасбақаны басқаруға мүмкіндік береді. Стандартты Python синтаксисі, басқару ағыны және деректер құрылымы тасбақа модулімен қатар қолданыла алатындықтан, тасбақа Python оқитын бағдарламашылар үшін тілдің негіздерімен танысудың танымал тәсілі болды.

Тасбақа графикасының идеяларын үш өлшемді кеңістікті қамтуға болады. Бұған бірнеше түрлі координаттар модельдерінің бірін қолдану арқылы қол жеткізіледі. Егер тасбақа цилиндрлік координаттарда жұмыс істейтін болса, онда оның жазықтықта орналасуы мен айдары болады, ал оның жазықтығы тік осьтің айналасында айналуы мүмкін. Бұл көбінесе тасбақаның екі түрлі бағыт бұрышы болатындығынан көрінеді, бірі жазықтықта, екіншісі жазықтықтың бұрышын анықтайды. Әдетте ұшақтың бұрышын өзгерту тасбақа қозғалмайды.