

Python тілінде программалар құру

11.1-дәріс. Сөздіктерді өңдеу

Сұрақтар

1. Сөздіктер құру.....	3
2. Сөздіктен мән алу.....	6
3. Сөздіктерде <i>in</i> және <i>not in</i> операторларын қолдану.....	8
4. Сөздікке элементтер қосу.....	10
5. Сөздік элементтерін өшіру.....	11
6. Сөздіктегі элементтер санын алу.....	13
7. Сөздіктегі мәліметтер типін араластыру.....	14
8. Бос сөздік құру.....	17
9. Сөздікті өңдеу кезінде for циклін қолдану.....	18
10. Бірсыпыра сөздік әдістері.....	19
11. Сөздікті карта топшасын бейнелеу үшін қолдану.....	31
12. Аттар мен туған күндерді сөздікте сақтау.....	39
13. Программаны орындау нәтижелері.....	50
14. Сұраққа жауап берейік.....	59

1. Сөздіктер құру

- **Сөздік (dictionary)** – бұл мәліметтер коллекциясын (тобын, жиынын) сақтайтын объект-контейнер. Сөздіктің әрбір элементі екі бөліктен: *кілттен* және *мәннен* тұрады. *Кілт* нақты бір *мәннің тұрған орнын анықтау* үшін қолданылады.
- Сөздік элементтері қосарланған *кілт/мән жұбынан* тұрады. Сөздіктегі **сөз** – *кілт*, ал оның **анықтамасы** – оның *мәні* болып табылады.
- Мысалы, программадағы адамның аты – кілт, оның телефон нөмірі – кілттің мәні. Егер адам аты белгілі болса, онда оның телефон нөмірін ала аламыз.
- Сөздік құру үшін жүйелік жақшалар (**{ }**) қолданылады. Мұндағы бір элемент *кілттен*, *қоснүктеден*, сонан соң *мәннен* тұрады. Олар үтірмен бөлінеді. Келесі нұсқау **phonebook** айнымалысына сөздікті меншіктеу арқылы оны анықтау өрнегі болып табылады:
**phonebook = { 'Крис' : '555-1111', 'Катя' : '555-2222',
 'Джоанна' : '555-3333' }**

Сөздіктер құру

- Сөздіктердегі мән кез келген типтегі объект бола алады, ал бірақ кілт мутацияланбайтын, яғни өзгермейтін объект болуы тиіс.
- Кілт ретінде **тіркестік мәндер, бүтін сандар, аралас сандар** немесе **кортеждер** де бола береді. Тізімдер немесе одан да басқа өзгеретін объектер кілт бола алмайды.
- Сөздік элементтері нақты бір реттілікпен орналаспайды. Мысал ретінде келесі нұсқауларды қарастырайық:

```
>>> phonebook = { 'Крис': '555-1111', 'Катя': '555-2222',
```

```
'Джоанна': '555-3333' } Enter
```

```
>>> phonebook Enter
```

```
{ 'Джоанна': '555-3333', 'Крис': '555-1111', 'Катя': '555-2222' }
```

```
>>>
```

- Мұнан сөздік элементтерінің құрылуы мен шығарылу реттіліктері бірдей емес екені көрініп тұр. Сөздіктер тізім, кортеж немесе тіркестік элементтері емес екені осыдан көрініп тұр.

Сөздіктер құру

- Мұнда нәтиже ретінде, сандық индекс арқылы мән алуды қолдануға болмайды. Мән алу үшін тек кілтті пайдалану керек.
- Сөздіктен мән алу үшін келесі форматтағы өрнек жазылады:
`сөздік_аты [кілт]`
- Бұл форматағы `сөздік_аты` – бұл сөздікке сілтеме жасайтын айнымалы, ал `кілт` – бұл қолданылып отырған кілт. Егер сөздікте *кілт* бар болатын болса, онда мұндағы өрнек сол кілтпен байланысты *мәнді* қайтарады.
- Ал егер сөздікте кілт жоқ болатын болса, онда **KeyError** (кілттің қатесі) аластамасы шақырылады. Келесі интерактивті мысал осы әрекетті көрсетеді.

2. Сөздіктен мән алу

```
1 >>> phonebook = {'Крис' : '555-1111', 'Катя' : '555-2222',  
    'Джоанна' : '555-3333'} Enter
```

```
2 >>> phonebook [ 'Крис' ] Enter
```

```
3 '555-1111'
```

```
4 >>> phonebook ['Джоанна'] Enter
```

```
5 '555-3333'
```

```
6 >>> phonebook [ 'Катя' ] Enter
```

```
7 '555-2222'
```

```
8 >>> phonebook ['Кэтрин'] Enter
```

```
Traceback (most recent call last) :
```

```
    File "<pyshell#5>", line 1, in <module>
```

```
phonebook [ 'Кэтрин' ]
```

```
KeyError: 'Кэтрин'
```

```
>>>
```

Сөздіктен мән алу

- Жоғарыдағы әрекеттерді толығырақ қарастырайық.
- 1-жол адам аттары (кілттер ретінде) мен телефон нөмірлері (мәндер ретінде) бар сөздік құрады.
- 2-жолдағы phonebook ['Крис'] өрнегі сөздіктен 'Крис' кілтімен байланысқан мәнді қайтарады. Ол 3-жолда көрсетіледі.
- 4-жолда phonebook ['Джоанна'] өрнегі сөздіктен 'Джоанна' кілтімен байланысты мәнді қайтарады. Ол 5-жолда көрсетіледі.
- 6-жолда phonebook ['Катя'] өрнегі сөздіктен 'Катя' кілтімен байланысты мәнді қайтарады. Ол 7-жолда көрсетіледі.
- 8-жолда phonebook ['Кэтрин'] өрнегі енгізіледі. Мұндай кілт сөздікте жоқ болғандықтан, **KeyError** аластамасы туындайды.
- Тіркестік мәндерді салыстыру операцияларында регистр дұрыс көрсетілуі тиіс, мысалы, **phonebook ['кэти']** сөздіктен **'Катя'** кілтін таба алмайды.

3.Сөздіктерде *in* және *not in* операторларын қолдану

- `KeyError` аластамасы туындамас үшін, кілттің алдын ала бар/жоқ екендігін тексеретін *in* операторын қолдану керек. Келесі интерактивті сеанс осыны көрсетеді:
 - 1 `>>> phonebook = {'Крис': '555-1111', 'Катя': '555-2222', 'Джоанна': '555-3333'}` Enter
 - 2 `>>> if 'Крис' in phonebook:` Enter
 - 3 `print (phonebook [ 'Крис' ] )` Enter Enter
 - 4
 - 5 `555-1111`
 - 6 `>>>`
- Мұнда `'Крис'` кілті бар екендігі тексеріледі де, оның кілтке сәйкес мәні `555-1111` көрсетіледі.

Сөздіктерде *in* және *not in* операторларын қолдану

- Кілттің сөздікте жоқ екендігін де тексеруге болады, ол үшін *not in* операторын қолдана аламыз, мысалы:

```
1 >>> phonebook = {'Крис': '555-1111', 'Катя': '555-2222'} Enter
```

```
2 >>> if 'Джоанна' not in phonebook: Enter
```

```
3 print ('Джоанна табылмады. ') Enter Enter
```

```
4
```

```
5 Джоанна табылмады.
```

```
6 >>>
```

- Мұндағы *in* және *not in* операторлары арқылы кілттерді тексеру кезінде олардың регистрге тәуелді екендігін есте ұстау керек.

4. Сөздікке элементтер қосу

- Сөздіктер өзгермелі объектілер, сондықтан оларға жаңа кілт/мән жұптарын қоса аламыз, ол үшін төмендегі форматтағы тәрізді меншіктеу нұсқауын қолдану керек:

`сөздік_аты [кілт] = мән`

- Мұндағы `сөздік_аты` - бұл сөздікке сілтеме жасайтын айнымалы, `кілт` – бұл қолданылатын кілт. Егер мұндай кілт бұрыннан бар болса, онда оған сәйкес мән өзгертіледі. Егер ондай кілт жоқ болатын болса, онда ол өзінің мәнімен бірге сөздікке қосылады. Мысалы:

```
1 >>> phonebook = {'Крис': '555-1111', 'Катя': '555-2222',
```

```
   'Джоанна': '555-3333'} Enter
```

```
2 >>> phonebook['Джо'] = '555-0123' Enter
```

```
3 >>> phonebook['Крис'] = '555-4444' Enter
```

```
4 >>> phonebook Enter
```

```
5 { 'Крис': '555-4444', 'Джоанна': '555-3333', 'Джо': '555-0123',  
   'Катя': '555-2222'}
```

```
6 >>>
```

5. Сөздік элементтерін өшіру

- `del` нұсқауымен сөздікте бұрыннан бар кілт/мән жұбын өшіре аламыз. Оның форматы: `del сөздік_аты[кілт]`
- Мұнда `сөздік_аты` – сөздікке сілтеме жасап тұрған айнымалы, ал кілт – бұл қолданылатын кілт. Бұл нұсқаудан кейін кілт пен оның мәні сөздіктен өшіріледі. Егер ондай кілт болмаса, онда `KeyError` аластамасы туындайды:

```
1 >>> phonebook = { 'Крис': '555-1111', 'Катя' : '555-2222',  
  'Джоанна': '555-3333'} Enter  
2 >>> phonebook Enter  
3 {'Chris': '555-1111', 'Джоанна': '555-3333', 'Катя': '555-2222'}  
4 >>> del phonebook [ 'Крис'] Enter  
5 >>> phonebook Enter  
6 {'Джоанна': '555-3333', 'Катя': '555-2222'}  
7 >>> del phonebook [ 'Крис'] Enter  
8 Traceback (most recent call last) :  
9   File "<pyshell#5>", line 1, in <module>  
10       del phonebook [ 'Крис']  
11 KeyError: 'Крис'  
12 >>>
```

Сөздік элементтерін өшіру

- `KeyError` аластамасын туындатпас үшін `in` операторын қолданып, алдын ала керекті кілт сөздікте бар/жоқ екенін біліп алуға болады:

```
1 >>> phonebook = { 'Крис': '555-1111', 'Катя': '555-2222',
```

```
    'Джоанна': '555-3333'}) Enter
```

```
2 >>> if 'Крис' in phonebook: Enter
```

```
3 del phonebook['Крис'] Enter Enter
```

```
4
```

```
5 >>> phonebook Enter
```

```
6 {'Джоанна': '555-3333', 'Катя': '555-2222'})
```

```
7 >>>
```

6. Сөздіктегі элементтер санын алу

- Сөздіктегі элементтер санын алу үшін `len` ішкі функциясын қолдана аламыз. Келесі интерактивті сеанс осыны көрсетеді:

```
1 >>> phonebook = {'Крис': '555-1111', 'Катя' : '555-2222'}) Enter
```

```
2 >>> num_items = len(phonebook) Enter
```

```
3 >>> print(num_items) Enter
```

```
4 2
```

```
5 >>>
```

- Мұндағы 1-жолда екі элементі бар сөздік жасалады да, ол `phonebook` айнымалысына меншіктеледі.
- 2-жол `len` функциясын шақырып, оған аргумент ретінде `phonebook` айнымалысын береді. Бұл функция `num_items` айнымалысына меншіктелетін 2 мәнін қайтарады.
- 3-жол `num_items` айнымалысын `print` функциясына береді. 4-жолда функция нәтижесі экранға шығарылады.

7. Сөздіктегі мәліметтер типін араластыру

- Сонымен, сөздіктегі кілттер өзгермейтін объект болуы тиіс, дегенмен онымен байланысқан мәндер кез келген типтегі объектілер түрінде бола береді. Мысалы, төменде көрсетілгендей, мәндер тізім түрінде де бола береді. Бұл мысалдағы сөздікте кілт ретінде студенттер аттары, ал олардың мәндері ретінде – солардың емтихан бағалары алынған.

```
1 >>> test_scores = { 'Келіс' : [88, 92, 100], Enter
2                        'Лора' : [95, 74, 81], Enter
3                        'Сапа' : [72, 88, 91], Enter
4                        'Итен' : [70, 75, 78] } Enter
5 >>> test_scores Enter
6 { 'Келіс' : [88, 92, 100], 'Сапа' : [72, 88, 91], 'Итен' : [70, 75, 78], 'Лора' : [95, 74, 81] }
7 >>> test_scores['Сапа'] Enter
8 [72, 88, 91]
9 >>> kelic_scores = test_scores['Келіс'] Enter
10 >>> print(kelic_scores) Enter
11 [88, 92, 100]
12 >>>
```

Сөздіктегі мәліметтер типін араластыру

- Бір сөздіктегі мәндер әртүрлі типтерде болуы мүмкін. Мысалы, бір элемент мәні – тіркестік объект, екінші элемент мәні – тізім, ал тағы бірі – бүтін сан болуы мүмкін. Кілттер де әртүрлі типтерде бола береді, бірақ олар өзгермейтін болып қалуы тиіс. Келесі мысал сөздіктегі типтердің қалай араласып келе беретінін көрсетеді:

```
1 >>> mixed_up = { 'абв':1, 999: 'тада тада', (3, 6, 9): [3, 6, 9] } Enter
```

```
2 >>> mixed_up Enter
```

```
3 { (3, 6, 9): [3, 6, 9], 'абв': 1, 999: 'тада тада' }
```

```
4 >>>
```

- 1-жол сөздік құрады да, оны mixed_up (смесь) айнымалысына меншіктейді. Бұл сөздікте төмендегі элементтер бар.
- Бірінші элемент – 'абв' : 1 '. Бұл элементте кілт - 'абв' тіркесі, мәні –1 бүтін саны.
- Екінші элемент – 999: 'тада тада'. Бұл элемент кілті бүтін сан – 999, мәні – 'тада тада' тіркестік мәні.
- Үшінші элемент – (3, 6, 9): [3, 6, 9]. Мұндағы кілт – кортеж (3, 6, 9), ал мәні – [3, 6, 9] тізімі.

Сөздіктегі мәліметтер типін араластыру

- Келесі жолдарда практикалық мысал көрсетілген. Онда қызметкер туралы мәліметтері бар сөздік жасалған:

```
1 >>> employee = {'фιο': 'Кевин Смит', 'ИД':12345, 'ставка' :25.75} Enter
```

```
2 >>> employee Enter
```

```
3 { 'ИД': 12345, 'ставка': 25.75, 'фιο': 'Кевин Смит' }
```

```
4 >>>
```

- 1-жолда сөздік құрылып, ол `employee` айнымалысына меншіктеледі. Бұл сөздікте төмендегідей элементтер бар.
- Бірінші элемент – `'фιο' : 'Кевин Смит'`. Мұндағы кілт - тіркестік литерал `'фιο'`, оның мәні – `'Кевин Смит'` тіркестік мәні.
- Екінші элемент- `'ид' :12345`. Бұл элементтегі кілт – тіркестік литерал - `'ид'`, оның мәні – `12345` бүтін саны.
- Үшінші элемент – `'ставка': 25.75`. Мұндағы кілт - тіркестік литерал `'ставка'`, оның мәні – нақты сан `25.75`.

8. Бос сөздік құру

- Кейде бос сөздік жасалып, оған программа жұмысы барысында элементтер қосылып отырады. Бос сөздік құру үшін бос жүйелік жақшалар алынады:

```
1 >>> phonebook = {} Enter
2 >>> phonebook['Крис'] = '555-1111' Enter
3 >>> phonebook['Катя'] = '555-2222' Enter
4 >>> phonebook['Джоанна'] = '555-3333' Enter
5 >>> phonebook Enter
6 { 'Крис' : '555-1111', 'Катя' : '555-2222', 'Джоанна' : '555-3333' }
7 >>>
```

- Мұнда 1-жолда бос сөздік құрылып, ол **phonebook** айнымалысына меншіктеледі. 2-4-жолдар сөздікке кілт/мән жұбын қосады, ал 5-жол сөздікті шығарады.
- Бос сөздік құру үшін **dict()** ішкі әдісін қолдануға да болдады:
phonebook = dict()
- Бұл нұсқаудан кейін **phonebook** айнымалысы бос сөздікке сілтеме жасап тұратын болады.

9. Сөздікті өңдеу кезінде for циклін қолдану

- Сөздіктегі барлық кілттерді қарастырып шығу үшін **for** циклі қолданылады:

for айнымалы **in** сөздік:

нұсқау

нұсқау

...

- Бұл форматтағы **айнымалы** – бұл айнымалы аты, **сөздік** – сөздік аты. Бұл цикл сөздіктің әрбір элементі үшін бір рет орындалып, цикл ішінде айнымалыға кілт меншіктеледі:

```
1 >>> phonebook = {'Крис' : '555-1111', Enter
2                  'Катя' : '555-2222', Enter
3                  'Джоанна' : '555-3333'} Enter
4 >>> for key in phonebook: Enter
5     print(key) Enter Enter
6 Крис
7 Катя
8 Джоанна
9 >>> for key in phonebook: Enter
10    print(key, phonebook[key]) Enter Enter
11 Крис 555-1111
12 Катя 555-2222
13 Джоанна 555-3333
14 >>>
```

1-3-жолдарда 3 элементі бар сөздік құрылды.

4-5-жолдар сөздіктің әрбір элементі үшін **for** циклін орындайды да, онда **key** айнымалысына кілт меншіктеледі. 5-жол **key** айнымалысын шығарады.

6-8-жолдар цикл жұмысының нәтижесін көрсетеді.

9-10-жолдар циклде сөздік элементтерін шығарады, онда **key** айнымалысына кілт меншіктеліп отырады.

11-13-жолдар цикл жұмысының нәтижесін көрсетеді.

10. Бірсыпыра сөздік әдістері

- Сөздік объектілерімен жұмыс істейтін көптеген әдістер бар. Солардың жиі пайдаланылатын бірсыпырасын қарастырайық, олар 9.1-кестеде көрсетілген.

9.1 кесте. Сөздіктерді өңдеу әдістері

Әдіс	Сипаттамасы
<code>clear ()</code>	Сөздіктің мәліметін өшіреді
<code>get ()</code>	Берілген кілтпен байланысқан мәнді алады. Егер кілт табылмаса, онда бұл әдіс аластама шақырмай, оның орнына, келісім бойынша алынатын мәнді қайтарады
<code>items ()</code>	Сөздіктегі барлық кілттерді және солармен байланысты мәндерді кортеж тізбектері түрінде қайтарады
<code>keys ()</code>	Сөздіктегі барлық кілттерді кортеж тізбектері түрінде қайтарады
<code>pop()</code>	Сөздіктегі берілген кілтпен байланысқан мәнді қайтарады да, осы кілт/мән жұбын өшіреді. Егер кілт табылмаса, онда бұл әдіс келісім бойынша алынатын мәнді қайтарады
<code>popitem()</code>	Сөздіктегі кездейсоқ таңдалып алынған кілт/мән жұбын кортеж түрінде қайтарады да, сөздіктен осы кілт/мән жұбын өшіреді.
<code>values ()</code>	Сөздіктегі барлық мәндерді кортеж тізбектері түрінде қайтарады

Бірсыпыра сөздік әдістері

`clear()` әдісі

- `clear()` әдісі сөздіктегі барлық элементтерді өшіріп, оны бос қалдырады. Оның жазылу форматы:

`сөздік.clear ()`

- Төмендегі жұмыс сеансы бұл әдістің жұмысын көрсетеді:

```
1 >>> phonebook = { 'Крис' : '555-1111', 'Катя' : '555-2222' } Enter
```

```
2 >>> phonebook Enter
```

```
3 {'Крис': '555-1111', 'Катя': '555-2222'}
```

```
4 >>> phonebook.clear() Enter
```

```
5 >>> phonebook Enter
```

```
6 {}
```

```
7 >>>
```

- Мұнда 4-жолдан кейін `phonebook` сөздігінде ешқандай элемент қалған жоқ.

Бірсыпыра сөздік әдістері

get() әдісі

- Сөздіктен мән алу үшін `[ ]` операторының баламасы ретінде `get()` әдісін қолдануға болады. Ол берілген кілт табылмаса да, аластама шақырмайды. Бұл әдістің жазылу форматы:

`сөздік.get (кілт, келісім_бойынша_мән)`

- Мұндағы `сөздік` – бұл сөздік аты, `кілт` – бұл сөздікте ізделетін кілт, `келісім_бойынша_мән` – кілт табылмағанда, қайтарылатын мән. Бұл әдіс шақырылғанда, берілген кілтпен байланысқан мән шығарылады. Егер берілген кілт табылмаса, онда келісім бойынша берілген мән қайтарылады. Келесі мысалда осы әдіс жұмысы көрсетіледі:

```
1 >>> phonebook = {'Крис': '555-1111', 'Катя': '555-2222'} Enter
```

```
2 >>> value = phonebook.get('Катя', 'Жазба табылмады') Enter
```

```
3 >>> print(value) Enter
```

```
4 555-2222
```

```
5 >>> value = phonebook.get('Энди', 'Жазба табылмады') Enter
```

```
6 >>> print(value) Enter
```

```
7 Жазба табылмады
```

```
8 >>>
```

Бірсыпыра сөздік әдістері

- Мұның жұмысын қарастырайық.
 - 2-жолдағы нұсқау `phonebook` сөздігіндегі `'Катя'` кілтін іздейді. Ол кілт табылды, сондықтан онымен байланысқан мән қайтарылады да, ол сонан кейін `value` айнымалысына меншіктеледі.
 - 3-жолда `value` айнымалысы `print` функциясына беріледі. Функция 4-жолда шығарылады.
 - 5-жолдағы нұсқау `phonebook` сөздігінен `'Энди'` кілтін іздейді. Ол кілт табылмайды, сондықтан `value` айнымалысына `'Жазба табылмады'` тіркесі меншіктеледі
 - 6-жол `value` айнымалысын `print` функциясына береді, оның нәтижесі 7-жолда шығарылады.

Бірсыпыра сөздік әдістері

items() әдісі

- items() әдісі сөздіктің барлық кілттері мен солармен байланысты мәндерін қайтарады. Олар **сөздік бейнесі** деп аталатын **ерекше типтегі тізбек** түрінде қайтарылады.
- Сөздік бейнесіндегі әрбір элемент **кортеж** болып табылады және әрбір кортеж кілт пен сонымен байланысты мәнді сақтайды. Мысалы, біз келесідей сөздік құрдық делік:

```
phonebook = {'Крис': '555-1111', 'Катя': '555-2222', 'Джоанна': '555-3333'}
```

- Егер `phonebook.items()` әдісін шақырсақ, ол келесі тізбекті қайтарады:

```
[ ('Крис', '555-1111'), ('Катя', '555-2222'), ('Джоанна', '555-3333') ]
```
- Мұндағы тізбектегі бірінші элемент – `('Крис', '555-1111')` кортежі;
- екінші элемент – `('Катя', '555-2222')` кортежі;
- үшінші элемент - `('Джоанна', '555-3333')` кортежі болып табылады.

Бірсыпыра сөздік әдістері

- Тізбектегі кортеждерді біртіндеп қарастыру үшін **for** циклін қолдана аламыз:

```
1 >>> phonebook = {'Крис': '555-1111', Enter
```

```
2 'Катя' : '555-2222', Enter
```

```
3 'Джоанна': '555-3333'} Enter
```

```
4 >>> for key, value in phonebook.items(): Enter
```

```
5 print(key, value) Enter Enter
```

```
6
```

```
7 Крис 555-1111
```

```
8 Катя 555-2222
```

```
9 Джоанна 555-3333
```

```
10 >>>
```

- 1-3-жолдар 3 элементі бар сөздік құрып, оны **phonebook** айнымалысына береді.
- 4-5-жолдарда **for** циклі кілт/мән жұбын сақтап, кортеждер тізбегін қайтаратын **phonebook.items()** әдісін шақырады. Бұл цикл әрбір кортеж үшін бір қадам жасап, сол қадамдардың әрбірінде кортеж мәндері **key** және **value** айнымалыларына меншіктеледі. 5-жол **key** айнымалысының, сонан соң **value** айнымалысы мәндерін шығарады. 8-10-жолдарда цикл нәтижесі шығарылады.

Бірсыпыра сөздік әдістері

keys() әдісі

- `keys()` әдісі сөздіктің барлық кілтін *сөздік бейнесі* деп аталатын *ерекше типтегі тізбек* түрінде қайтарады.
- Сөздік бейнесіндегі әрбір элемент сөздік кілті болып табылады. Мысалы, біз мынадай сөздік құрдық делік:

```
phonebook = {'Крис': '555-1111', 'Катя': '555-2222', 'Джоанна': '555-3333'}
```

- Егер енді `phonebook.keys()` әдісін шақыратын болсақ, онда ол төмендегі тізбекті қайтарады:

```
[ ' Крис ' , ' Катя ' , ' Джоанна ' ]
```

- Келесі интерактивті жұмыс сеансы `keys()` әдісінен қайтарылған тізбекке `for` циклін қалай қолдануға болатынын көрсетеді :

```
1 >>> phonebook = {'Крис': '555-1111', Enter
```

```
2 'Катя': '555-2222', Enter
```

```
3 'Джоанна': '555-3333'} Enter
```

```
4 >>> for key in phonebook.keys(): Enter
```

```
5 print (key) Enter Enter
```

```
6
```

```
7 Крис
```

```
8 Катя
```

```
9 Джоанна
```

```
10 >>>
```

Бірсыпыра сөздік әдістері

pop() әдісі

- **pop()** әдісі берілген кілтпен байланысқан мәнді қайтарады да, осы кілт/мән жұбын сөздіктен өшіреді. Егер кілт табылмаса, онда әдіс келісім бойынша тағайындалған мәнді қайтарады. Мұның жалпы жазылу форматы:
сөздік.pop (кілт, келісім_бойынша_мән)
- Мұндағы **сөздік** – бұл сөздік аты, **кілт** – бұл сөздіктен ізделіп отырған кілт, **келісім_бойынша_мән** – ол кілт табылмаған жағдайда қайтарылатын мән. Бұл әдіс шақырылғанда, ол берілген кілтпен байланысқан мәнді қайтарады да, осы кілт/мән жұбын сөздіктен өшіреді. Егер кілт сөздіктен табылмаса, онда әдіс келісім бойынша тағайындалған мәнді қайтарады. Мысал келтірейік:

```
1 >>> phonebook = { 'Крис': '555-1111', Enter
2                   'Катя' : '555-2222', Enter
3                   'Джоанна': '555-3333'} Enter
4 >>> phone_num = phonebook.pop('Крис', 'Жазба табылмады') Enter
5 >>> phone_num Enter
6 '555-1111'
7 >>> phonebook Enter
8 {'Катя': '555-2222', 'Джоанна': '555-3333'}
9 >>> phone_num = phonebook.pop('Энди', 'Жазба табылмады') Enter
10 >>> phone_num Enter
11 Жазба табылмады
12 >>> phonebook Enter
13 {'Катя': '555-2222', 'Джоанна': '555-3333'}
14 >>>
```

Бірсыпыра сөздік әдістері

- Осы жұмыс сеансына қысқаша түсінік берейік.
 - 1-3-жолдар үш элементі бар сөздік құрады да, оны `phonebook` айнымалысына меншіктейді.
 - 4-жол `phonebook.pop()` әдісін шақырып, оған іздеу кілті ретінде `'Крис'` сөзін береді. Сонда `'Крис'` кілтімен байланысқан мән қайтарылып, `phone_num` айнымалысына меншіктеледі де, `'Крис'` кілті бар кілт/мән жұбы сөздіктен өшіріледі.
 - 5-жол `phone_num` айнымалысына меншіктелген мәнді көрсетеді де, нәтижесі 6-жолда шығарылады. Оның `'Крис'` кілтімен байланысқан мән екеніне назар аударыңыздар.
 - 7-жол `phonebook` сөздігінің мазмұнын көрсетеді. Оның нәтижесі 8-жолда көрсетілген. Енді `'Крис'` кілті бар кілт/мән жұбы сөздіктен өшіріледі.
 - 9-жол `phonebook.pop()` әдісін шақырып, оған іздеу кілті ретінде `'Энди'` сөзін береді. Бұл кілт табылмайды, сондықтан `phone_num` айнымалысына `'Жазба табылмады'` тіркестік литералы меншіктеледі.
 - 10-жол `phone_num` айнымалысының мәнін көрсетеді. Ол 11-жолда шығарылады.
 - 12-жол `phonebook` сөздігінің мәліметтерін көрсетеді. Нәтижесі 13-жолда шығарылып тұр.

Бірсыпыра сөздік әдістері

popitem() әдісі

- `popitem()` әдісі кездейсоқ таңдалып алынған кілт/мән жұбын қайтарады да, соларды сөздіктен өшіреді. Кілт/мән жұбы кортеж түрінде қайтарылады. Осы әдістің жалпы жазылу форматы:

`сөздік.popitem()`

- Қайтарылатын кілт пен мәнді жеке айнымалыға меншіктеу үшін қолданылатын меншіктеу операторы төмендегідей форматта жазылады:

`k, v = сөздік.popitem()`

- Мұндай меншіктеу типі еселік меншіктеу деп аталады, өйткені мән бірден бірнеше айнымалыға меншіктеледі. Жоғарыдағы форматта `k` мен `v` – айнымалылар. Бұл нұсқаудан кейін `k` айнымалысына сөздіктен **кездейсоқ таңдалып алынған кілт**, ал `v` айнымалысына – сол кілтпен байланысқан мән меншіктеледі. Сонан соң осы кілт/мән сөздіктен өшіріледі. Келесі интерактивті жұмыс сеансы осыны көрсетеді.

Бірсыпыра сөздік әдістері

```
1 >>> phonebook = {'Крис': '555-1111', Enter
2 'Катя' : '555-2222', Enter
3 'Джоанна': '555-3333'} Enter
4 >>> phonebook Enter
5 {'Крис': '555-1111', 'Катя': '555-2222', 'Джоанна': '555-3333'}
6 >>> key, value = phonebook.popitem() Enter
7 >>> print(key, value) Enter
8 Крис 555-1111
9 >>> phonebook Enter
10 {'Катя': '555-2222', 'Джоанна': '555-3333'}
11 >>>
```

- Мұндағы 1-3 жолдар 3 элементі бар сөздік құрып, оны **phonebook** айнымалысына меншіктейді.
- 4-жол сөздік мәліметін шығарады, ол 5-жолда көрсетілген.
- 6-жол **phonebook.popitem()** әдісін шақырады. Осы әдістен алынған кілт пен мән **key** және **value** айнымалыларына меншіктеледі. Сөздіктен кілт/мән өшіріледі.
- 7-жол **key** және **value** айнымалыларына меншіктелген мәндерді шығарады. Нәтижесі 8-жолда көрсетілген.
- 9-жол сөздік мәліметін шығарады. Нәтижесі 10-жолда тұр. Мұндағы 6-жолда **popitem ()** әдісінен қайтарылған кілт/мән өшірілгеніне назар салыңыздар.

Бірсыпыра сөздік әдістері

values() әдісі

- values() әдісі сөздіктің барлық мәндерін (кілттерсіз) сөздік бейнесі, яғни ерекше тізбек түрінде қайтарады. Сөздік бейнесіндегі әрбір элемент сөздік мәні болып табылады. Мысалы, біз келесідей сөздік құрдық делік:

```
phonebook = { 'Крис': '555-1111', 'Катя': '555-2222', 'Джоанна': '555-3333' }
```

- Егер phonebook.values() әдісін шақырсақ, онда ол мынадай тізбек қайтарады:
['555-1111', '555-2222', '555-3333']
- Келесі интерактивті сеанс values() әдісінен қайтарылатын тізбекке қалай for циклін қолдануға болатынын көрсетеді:

```
>>> phonebook = ('Крис' : '555-1111', Enter
```

```
'Катя': '555-2222', Enter
```

```
'Джоанна': '555-3333') Enter
```

```
>>> for val in phonebook.values(): Enter
```

```
print(val) Enter
```

```
555-1111
```

```
555-3333
```

```
555-2222
```

```
>>>
```

11.Сөздікті карта топшасын бейнелеу үшін қолдану

- Кейбір ойын карталарымен байланысты ойындарда карталарға сандық мәндер тағайындалады. Мысалы, блек-джек деген ойында карталарға сандық мәндер былай беріледі:
 - сандық карталарға оларда жазылған сандық мәндер меншіктеледі. Мысалы, қарғаның екілігіне 2, ал қиықтың бестігіне 5 мәні беріледі;
 - валетке, дамаға және корольге 10 мәні беріледі;
 - тұзға ойыншының қалауына қарай 1 немесе 11 тағайындалады.
- Бұл бөлімде ойын картасының стандартты топшасын бейнелеу үшін сөздіктерде карталарға сандық мәндер (блек-джек ойыны тәрізді) берілетін программа қарастырылады. Мұнда барлық тұздарға 1 мәні беріледі. Кілт/мән жұптарында картаның аты кілт болады да, картаның сандық көрсеткіші – оның мәні болып саналады.

Сөздікті карта топшасын бейнелеу үшін қолдану

- Мысалы, табанның дамасы үшін кілт/мән жұбы мынадай болады:
'Дама табан' : 10
- Ал қиықтың сегіздігі үшін ол мынадай болып беріледі:
'8-дік қиық': 8
- Программа қолданушыға таратылатын карталар санын енгізуді ұсынады, кездейсоқ түрде оларды ойыншыларға таратып береді. Сонан соң таратылған карталардың аттары және де олардың сандық мәндері шығарылады.
- 9.1 программада осыған сәйкес код келтірілген. Программа үш функция бөлінген: `main` (басты), `create_deck` (топша құру) және `deal_cards` (картаны тарату). Бұлардың барлығын бірден емес біртіндеп қарастырайық, алдымен басты `main` функцияны келтіреміз.

Сөздікті карта топшасын бейнелеу үшін қолдану

```
1  # Программа 9.1 (card_dealer.py). Басты функция
2  # Бұл программа карталар ретінде сөздікті пайдаланады.
3  def main():
4      # Карта топшасын құру.
5      deck = create_deck()
6
7      # Таратылатын карталар санын алу.
8      num_cards = int(input('Неше карта таратылады?'))
9
10     # Карталарды тарату.
11     deal_cards(deck, num_cards)
12
```

- 4-жол `create_deck` функциясын шақырады. Ол карта топшасы үшін кілт/мән жұптары бар сөздік құрады да, сөздікке сілтеме қайтарады. Сілтеме `deck` (топша) айнымалысына меншіктеледі.
- 6-жол қолданушыға таратылатын карталар санын енгізуді сұрайды. Енгізілген мән бүтін `int` типіне түрлендіріліп, `num_cards` айнымалысына меншіктеледі.
- 8-жол `deal_cards` функциясын шақырып, оған аргумент ретінде `deck` және `num_cards` айнымалыларын береді. `deal_cards` функциясы топшадан берілген карталар санын таратады. Енді `create_deck()` функциясын құрамыз.

Сөздікті карта топшасын бейнелеу үшін қолдану

```
13 # Программа 9.1 (жалғасы). create_deck функциясы
14 # create_deck функциясы карта топшасын бейнелейтін сөздікті қайтарады.
15 def create_deck():
16     # Әрбір карта мен оның мәні кілт/мән жұптары түрінде
17     # сақталатын сөздік құру.
18     deck = {'Қарғаның тұзы': 1, '2-лік қарға': 2, '3-тік қарға': 3,
19             '4-тік қарға': 4, '5-тік қарға': 5, '6-лық қарға': 6,
20             '7-лік қарға': 7, '8-дік қарға': 8, '9-дық қарға': 9,
21             '10-дық қарға': 10, 'Валет қарға': 10,
22             'Дама қарға': 10, 'Король қарға': 10,
23
24             'Табанның тұзы': 1, '2-лік табан': 2, '3-тік табан': 3,
25             '4-тік табан': 4, '5-тік табан': 5, '6-лық табан': 6,
26             '7-лік табан': 7, '8-дік табан': 8, '9-дық табан': 9,
27             '10-дық табан': 10, 'Валет табан': 10,
28             'Дама табан': 10, 'Король табан': 10,
29
```

Сөздікті карта топшасын бейнелеу үшін қолдану

```
30     'Шыбынның тұзы': 1, ' 2-лік шыбын': 2, ' 3-тік шыбын': 3,  
31     '4-тік шыбын': 4, ' 5-тік шыбын': 5, ' 6-лық шыбын': 6,  
32     '7-лік шыбын': 7, ' 8-дік шыбын': 8, ' 9-дық шыбын': 9,  
33     '10-дық шыбын': 10, 'Валет шыбын': 10,  
34     'Дама шыбын ': 10, ' Король шыбын ': 10,  
35  
36     'Қиықтың тұзы': 1, ' 2-лік қиық': 2, ' 3-тік қиық': 3,  
37     '4-тік қиық': 4, ' 5-тік қиық': 5, ' 6-лық қиық': 6,  
38     '7-лік қиық': 7, ' 8-дік қиық': 8, ' 9-дық қиық': 9,  
39     '10-дық қиық': 10, 'Валет қиық': 10,  
40     'Дама қиық': 10, ' Король қиық': 10}  
41  
42     # Карта топшасын қайтару.  
43     return deck  
44
```

- 18-40-жолдардағы код кілт/мән жұптарынан тұратын сөздік құрады, олар стандартты ойын топшасы карталарын бейнелейді. 22, 29 және 35-бос жолдар программалық кодты оқуды жеңілдету үшін берілген.
- 43-жол сөздікке сілтеме қайтарады. Енді `deal_cards` функциясын қарастырамыз.

Сөздікті карта топшасын бейнелеу үшін қолдану

```
45 # Программа 9.1 (жалғасы). deal_cards функциясы
46 # deal_cards функциясы топшасындағы
47 # берілген карталар санын таратады.
48
49 def deal_cards(deck, number):
50     # Қолдағы карталар саны үшін жинақтауышты инициалдау.
51     hand_value = 0
52
53     # Таратылатын карталар саны топшадағы карталар
54     # санынан артық емес екендігіне көз жеткізу.
55     if number > len(deck):
56         number = len(deck)
57
58     # Карталарды таратып, олардың мәндерін жинақтау.
59     for count in range(number):
60         card, value = deck.popitem()
61         print(card)
62         hand_value += value
63     # Қолдағы карталардың сандық шамасын көрсету.
64     print('Қолдағы карталардың сандық шамасы: ', hand_value)
65
66 main() # Басты функцияны шақыру.
```

Сөздікті карта топшасын бейнелеу үшін қолдану

- `deal_cards` функциясы екі аргумент қабылдайды: таратылатын карталар саны және олар таратылатын топша. 50-жол жинақтайтын `hand_value` айнымалысын (қолдағы карталар мәндерінің сомасы) 0-мен инициалдайды. 54-жолдағы `if` нұсқауы таратылатын карталар санының топшадағы карталар санынан артық емес екендігін анықтайды. Егер бұл дұрыс болса, онда 55-жол таратылатын карталар санын топшадағы карталар санымен бірдей қылады.
- 58-жолдан басталатын `for` циклі әрбір таратылатын карта үшін бір қадам жасайды. Цикл ішіндегі 59-жол `popitem()` әдісін шақырады, ал ол, өз кезегінде, `deck` сөздігінен кез келген кілт/мән жұбын қайтарады да, сол жұпты сөздіктен өшіреді. Сөздіктегі элементтерді өшірмей таңдап алу үшін `random()` әдісін қолдану керек.
- Сонан соң кілт `card` айнымалысына, ал оның мәні – `value` айнымалысына меншіктеледі. 60-жол карт атын көрсетеді, 61-жол карта мәнін соларды жинақтайтын `hand_value` айнымалысына қосады.

Сөздікті карта топшасын бейнелеу үшін қолдану

- Цикл жұмысы аяқталған соң, 64-жол қолдағы карталар комбинациясының сандық мәндері қосындысын көрсетеді.

Программаның нәтижелері (енгізілген мәліметтер қарайтылған)

Неше карта таратылады? 5

Король қиық

Дама қиық

Валет қиық

10-дық қиық

9-дық қиық

Қолдағы карталардың сандық шамасы: 49

12. Адам аттары мен туған күндерін сөздікте сақтау

- Бұл бөлімде біз сөздікте сіздердің достарыңыз бен солардың туылған күндерін сақтайтын программаны қарастырамыз, Сөздіктегі әрбір жазба досыңыздың атын кілт ретінде, ал оның туылған күнін мән ретінде қабылдайды. Бұл программаны достарыңызды атын енгізіп, туылған күнін анықтау үшін қолдана аласыздар.
- Программа төмендегі әрекеттердің бірін таңдау мүмкіндігін беретін менюді көрсететін болады:
 1. Туылған күнді табу.
 2. Жаңа туылған күнді енгізу.
 3. Туылған күнді өзгерту.
 4. Туылған күнді өшіру.
 5. Программадан шығу.

Адам аттары мен туған күндерін сөздікте сақтау

- Программа жұмысты алдымен бос сөздіктен бастайды, сондықтан сіз менюдің 2 пунктін таңдап, жаңа жазба енгізесіз. Сіз бірнеше жазба енгізгеніңізден кейін, менюдің 1-пунктін таңдап, белгілі бір адамның туылған күнін іздеп таба аласыз.
- Ал 3 пункт енгізілген туған күнді өзгерту үшін керек. 4 пункт қате терілген туған күнді дұрыстайды, 5 пункт программадан шығарады.
- 9.2 программада осыған сәйкес код берілген. Программа алты функцияға бөлінген: `main` (басты), `get_menu_choice` (меню пунктін алу), `look_up` (іздеп табу), `add` (қосу), `change` (өзгерту) және `delete` (өшіру).
- Барлық программаны толық келтіруден бұрын, алдымен глобальді константалар мен басты функцияны `main` қарастырып шығайық.

Адам аттары мен туған күндерін сөздікте сақтау

9.2 программа (birthdays.py). Басты функция

1 # Бұл программа аттар мен туылған күндерді

2 # сақтау үшін сөздікті қолданады.

3

4 # Меню пункттеріне арналған глобальдік константалар

5 LOOK_UP = 1

6 ADD = 2

7 CHANGE = 3

8 DELETE = 4

9 QUIT = 5

10

11 # Басты функция.

12 def main():

13 # Бос сөздік жасау.

14 birthdays = {}

15

Адам аттары мен туған күндерін сөздікте сақтау

```
16     # Қолданушыны таңдауға арналған айнымалыны инициалдау.
17     choice = 0
18
19     while choice != QUIT:
20         # Қолданушының таңдаған меню пунктін алу.
21         choice = get_menu_choice()
22
23         Таңдалып алынған әрекет нұсқасын өңдеу.
24         # if choice == LOOK_UP:
25             look_up(birthdays)
26         elif choice == ADD:
27             add (birthdays)
28
29         elif choice == CHANGE:
30             change(birthdays)
31         elif choice == DELETE:
32             delete(birthdays)
```

Адам аттары мен туған күндерін сөздікте сақтау

- 5-9-жолдарда жарияланған константалар қолданушы таңдап алған меню пункттерін тексеру үшін қолданылады. Басты функциядағы 14-жол `birth-days` айнымалысы сілтеме жасайтын бос сөздік құрады. 17-жол `choice` айнымалысын 0 мәнімен инициалдайды. Бұл айнымалы қолданушы таңдап алған меню пунктін сақтап тұрады.
- 19-жолда басталатын `while` циклі қолданушының программадан шығар кезіне дейін қайталанады. Цикл ішіндегі 21-жол `get_menu_choice` функциясын шақырады. `get_menu_choice` функциясы менюді шығарып, қолданушы таңдаған пунктті қайтарады. Қайтарылған мән `choice` айнымалысына меншіктеледі.
- 24-31-жолдардағы `if-elif` нұсқауы қолданушы таңдап алған меню пунктін өңдейді. Егер ол 1 пунктті таңдаса, онда 25-жол `look_up` функциясын шақырады. Егер ол 2 пунктті таңдаса, онда 27-жол `add` функциясын шақырады. Егер ол 3 пунктті таңдаса, онда 29-жол `change` функциясын шақырады. Егер ол 4 пунктті таңдаса, онда 31-жол `delete` функциясын шақырады.

Адам аттары мен туған күндерін сөздікте сақтау

```
# Программа 9.2 (жалғасы). get_menu_choice функциясы
33 # get_menu_choice функциясы меню шығарады және қолданушы
34 # таңдап алған меню пунктiнiң дұрыс екендiгiн тексередi.
35 def get_menu_choice():
36     print ()
37     print ('Достар және олардың туылған күндері')
38     print('-----')
39     print('1. Туылған күндерді табу')
40     print ('2. Жаңа туылған күнді қосу')
41     print('3. Туылған күнді өзгерту')
42     print('4. Туылған күнді өшіру')
43     print ( '5. Программадан шығу')
44     print()
45
46     # Қолданушы таңдап алған меню пунктін алу.
47     choice = int ( input ('Таңдап алынған меню пунктін енгізіңіз: ' ) )
48
```

Адам аттары мен туған күндерін сөздікте сақтау

```
49     # Таңдап алынған меню пунктінің дұрыс екендігін тексеру.  
50     while choice < LOOK UP or choice > QUIT:  
51         choice = int(input('Таңдап алынған меню пунктін енгізіңіз: '))  
52  
53     # Қолданушы таңдап алған меню пунктін қайтару.  
54     return choice  
55
```

- 36-44-жолдар экранға меню шығарады. 47-жол қолданушыға таңдалған пунктті енгізуді ұсынады. Енгізілген мән int типіне келтіріледі де, **choice** айнымалысына меншіктеледі. 50-51-жолдардағы while циклі қолданушы енгізген мәннің дұрыстығын тексереді де, қажет болса, оны қайта енгізуді сұрайды. Қолданушы менюдің дұрыс мәнін енгізген соң, функция өз жұмысын аяқтайды.

Адам аттары мен туған күндерін сөздікте сақтау

Ары қарай look_up функциясы жұмыс істейді.

```
# Программа 9.2 (жалғасы). look_up функциясы
56 # look_up функциясы birthdays сөздігінен
57 # керекті атты іздеп табады.
58 def look_up(birthdays):
59 # Керекті атты алу.
60 name = input ('Ат енгізіңіз: ')
61
62 # Оны сөздіктен іздеп табу.
63 print(birthdays.get(name, 'Табылмады.'))
64
```

- look_up функциясының міндеті дос адамның туылған күнін табу. Ол аргумент ретінде сөздікті қабылдайды. 60-жол ат енгізуді сұрайды, 63-жол осы атты сөздіктің get функциясына аргумент етіп береді. Егер ат табылса, онда сонымен байланысты мән қайтарылады (достың туылған күні) да, экранға шығарылады. Егер ат табылмаса, онда 'Табылмады.' сөзі шығарылады.

Адам аттары мен туған күндерін сөздікте сақтау

Ары қарай add функциясын қарастырамыз .

```
# Программа 9.2 (жалғасы). add функциясы
65 # add функциясы birthdays сөздігіне
66 # жаңа жазба қосады.
67 def add(birthdays):
68     # Атты және туылған күнді алу.
69     name = input ('Атты енгізіңіз: ' )
70     bday = input ('Туылған күнді енгізіңіз: ' )
71
72     # Егер ат болмаған болса, онда оны қосу.
73     if name not in birthdays:
74         birthdays[name] = bday
75     else:
76         print('Бұл жазба бұрыннан бар.')
77
```

Add функциясының міндеті сөздікке жаңа жазба қосу. Ол аргумент ретінде сөздікті қабылдайды. 69- және 70-жолдар қолданушыға жаңа ат пен туылған күнді енгізуді ұсынады. 73-жолдағы if нұсқауы сөздікте осы ат бар екендігін анықтайды.

Егер ол болмаса, онда 74-жол жаңа ат пен туылған күнді сөздікке қосады. Әйтпесе 76-жолда бұл жазбаның бұрыннан бар екені жазылады.

Адам аттары мен туған күндерін сөздікте сақтау

Ары қарай `change` функциясы келтіріледі.

78 # `change` функциясы `birthdays` сөздігіндегі

79 # бұрынғы жазбаны өзгертеді.

80 `def change(birthdays):`

81 # Керекті атты алу.

82 `name = input ( 'Ат енгізіңіз: ' )`

83

84 `if name in birthdays:`

85 # Жаңа туылған күнді алу.

86 `bday = input ( 'Жаңа туылған күнді енгізіңіз: ' )`

87

88 # Жазбаны жаңалау.

89 `birthdays[name] = bday`

90 `else:`

91 `print ( 'Бұл ат табылмады. ' )`

92

Мұндағы `change` функциясының міндеті сөздікке жаңа жазба қосу. Ол аргумент ретінде сөздікті қабылдайды. 82-жол атты қабылдайды. 84-жолдағы `if` нұсқауы сөздікте осы ат бар/жоқ екендігін анықтайды.

Егер ол болса, онда 86-жол жаңа туылған күнді алады да, оны сөздікте сақтайды. Егер ол ат сөздікте болмаса, 91-жол соған сәйкес мәлімет шығарады.

Адам аттары мен туған күндерін сөздікте сақтау

Ары қарай `delete` функциясы келтіріледі.

```
# Программа 9.2 (жалғасы). delete функциясы
93 # delete функциясы birthdays сөздігінен
94 # жазбаны өшіреді.
95 def delete(birthdays):
96     # Ізделетін атты алу.
97     name = input ('Атты енгізіңіз: ')
98
99     # Егер ат табылса, онда бұл жазбаны өшіру.
100    if name in birthdays:
101        del birthdays[name]
102    else:
103        print ('Ат табылмады.')
104
105 # Басты функцияны шақыру.
106 main ()
```

Мұндағы `delete` функциясының міндеті сөздіктегі ат пен туған күнді өшіру. Ол аргумент ретінде сөздікті қабылдайды. 97-жол атты қабылдайды. 100-жолдағы `if` нұсқауы сөздікте осы ат бар/жоқ екендігін анықтайды.

Егер ол болса, онда 101-жол оны өшіреді. Егер ол ат сөздікте болмаса, 103-жол соған сәйкес мәлімет шығарады.

13. Программаны орындау нәтижелері

Программа нәтижелері (енгізілген мәліметтер қарайтылған)

Достар және олардың туылған күндері

1. Туылған күнді табу
2. Туылған күнді қосу
3. Туылған күнді өзгерту
4. Туылған күнді өшіру
5. Программадан шығу

Таңдалған пунктті енгізіңіз : **2** Enter

Атты енгізіңіз : **Кэмерон** Enter

Туылған күнді енгізіңіз : **10/12/1990** Enter

Программаны орындау нәтижелері

Достар және олардың туылған күндері

1. Туылған күнді табу
2. Туылған күнді қосу
3. Туылған күнді өзгерту
4. Туылған күнді өшіру
5. Программадан шығу

Таңдалған пунктті енгізіңіз: **2** Enter

Атты енгізіңіз: **Кэтрин** Enter

Туылған күнді енгізіңіз: **5/7/1989** Enter

Программаны орындау нәтижелері

Достар және олардың туылған күндері

1. Туылған күнді табу
2. Туылған күнді қосу
3. Туылған күнді өзгерту
4. Туылған күнді өшіру
5. Программадан шығу

Таңдалған пунктті енгізіңіз: **1** Enter

Атты енгізіңіз : **Кэмерон** Enter

10/12/1990

Программаны орындау нәтижелері

Достар және олардың туылған күндері

1. Туылған күнді табу
2. Туылған күнді қосу
3. Туылған күнді өзгерту
4. Туылған күнді өшіру
5. Программадан шығу

Таңдалған пунктті енгізіңіз : **1** Enter

Атты енгізіңіз : **Кэтрин** Enter

5/7/1989

Программаны орындау нәтижелері

Достар және олардың туылған күндері

1. Туылған күнді табу
2. Туылған күнді қосу
3. Туылған күнді өзгерту
4. Туылған күнді өшіру
5. Программадан шығу

Таңдалған пунктті енгізіңіз : **3** Enter

Атты енгізіңіз: **Кэтрин** Enter

Введите новый день рождения: **5/7/1988** Enter

Программаны орындау нәтижелері

Достар және олардың туылған күндері

1. Туылған күнді табу
2. Туылған күнді қосу
3. Туылған күнді өзгерту
4. Туылған күнді өшіру
5. Программадан шығу

Таңдалған пунктті енгізіңіз : **1** Enter

Атты енгізіңіз : **Кэтрин** Enter

5/7/1988

Программаны орындау нәтижелері

Достар және олардың туылған күндері

1. Туылған күнді табу
2. Туылған күнді қосу
3. Туылған күнді өзгерту
4. Туылған күнді өшіру
5. Программадан шығу

Таңдалған пунктті енгізіңіз : **4** Enter

Атты енгізіңіз : **Кэмерон** Enter

Программаны орындау нәтижелері

Достар және олардың туылған күндері

1. Туылған күнді табу
2. Туылған күнді қосу
3. Туылған күнді өзгерту
4. Туылған күнді өшіру
5. Программадан шығу

Таңдалған пунктті енгізіңіз : **1** Enter

Атты енгізіңіз : **Кэмерон** Enter

Табылмады

Программаны орындау нәтижелері

Достар және олардың туылған күндері

1. Туылған күнді табу
2. Туылған күнді қосу
3. Туылған күнді өзгерту
4. Туылған күнді өшіру
5. Программадан шығу

Таңдалған пунктті енгізіңіз : **5** Enter

14. Сұраққа жауап берейік

9.1. Сөздіктегі элементтің екі бөлігі бар. Олар қалай аталады?

9.2. Сөздіктің қай элементі өзгермейді (мутацияланбайды)?

9.3. Мысалы, 'старт' : 14 7 2 сөздік элементі болсын. Қайсысы кілт, қайсысы мән болып саналады?

9.4. Мысалы, `employee` атты сөздік құрылды делік. Келесі нұсқау не істейді?

```
employee['id'] = 54321
```

9.5. Келесі код фрагменті не істейді?

```
stuff = {1: 'aaa', 2: 'ббб', 3: 'ввв'}    print(stuff[3])
```

9.6. Сөздікте кілт/мән жұбы бар екені қалай анықталады?

9.7. Мысалы, `inventory` атты сөздік құрылды делік. Келесі нұсқау не істейді?

```
del inventory[654]
```

Сұраққа жауап берейік

9.8. Келесі код фрагменті не істейді?

```
stuff = {1: 'aaa', 2: 'ббб', 3: 'ввв'}  
print(len(stuff))
```

9.9. Келесі код фрагменті не істейді?

```
stuff = { 1: 'aaa', 2: 'ббб', 3: 'ввв' }  
for k in stuff: print(k)
```

9.10. Сөздіктердегі `pop()` және `popitem()` әдістерінің айырмашылығы неде?

9.11. Сөздіктегі `items()` әдісі нені қайтарады?

9.12. Сөздіктегі `keys()` әдісі нені қайтарады?

9.13. Сөздіктегі `values()` әдісі нені қайтарады?

СОҢЫ...