

ПРИПРЕМА ЧАСА
РАЧУНАРСТВО И ИНФОРМАТИКА ЗА ШЕСТИ РАЗРЕД

Наставна тема:	Рачунарство
Редни број часа:	16.
Наставна јединица:	Пајтон - филтрирање
Тип часа:	Вежба
Циљ часа:	Упознавање ученика са појмом <i>филтрирања</i> колекција података и начину на који се оно може извести у оквиру програмског језика Пајтон
Исходи часа:	<ul style="list-style-type: none"> • Ученик разуме појам филтрирања елемената колекције података • Ученик уме да из листе података издвоји елементе који задовољавају одређени услов • Ученик уме да од елемената листе који задовољавају одређени услов формира нову листу • Ученик уме да препозна проблемску ситуацију и примени филтрирање у решавању проблемске ситуације
Облици рада:	фронтални, индивидуални, рад у пару, метода практичног рада
Наставне методе:	демонстративна, метода разговора, метода усменог излагања, метода практичног рада
Место реализације часа:	рачунарска кабинет

Ток часа

Уводни део часа (5 мин)

У уводном делу часа објаснити појам филтрирање података. Објаснити ученицима да је то нешто са чим су се већ сигурно срили али нису можда знали да се тако зове.

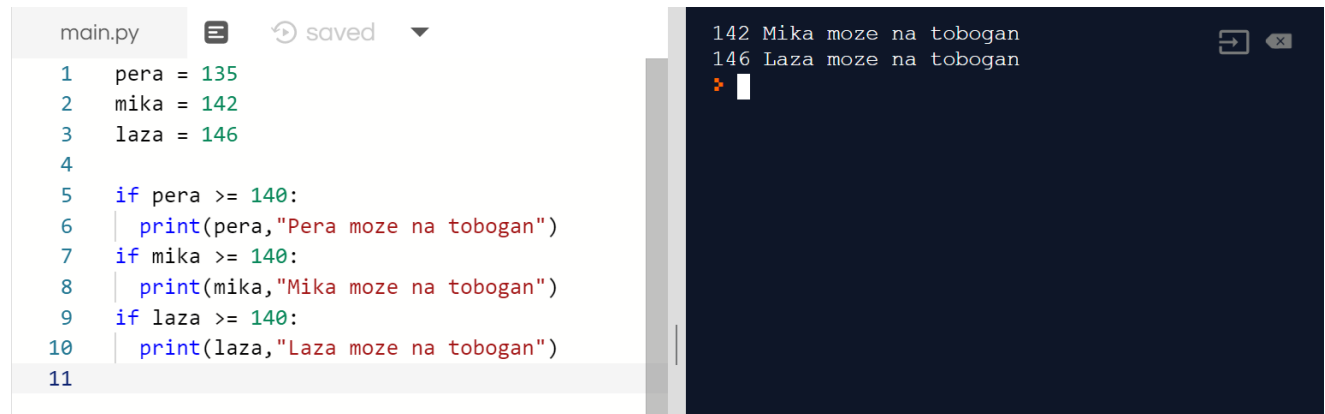
Филтрирање података објаснити на примеру тобогана. Сигурно су приметили да на неке тобогане могу да иду само она деца која су виша од 140цм. Дакле како из неког скупа бројева или из неке листе бројева издвојити бројеве који задовољавају одређени критеријум.

ГЛАВНИ ДЕО ЧАСА (35 МИНУТА)

Задатак 1: Написати програм који ће за дате висине у сантиметрима Пере,Мике и Лазе исписати коментар да ли се свако од њих појединачно може спуштати низ велики тобоган, при чему је услов за спуштање низ тобоган висина која је најмање 140 центиметара.

Решење:

(Слика 1)



```
main.py saved
1  pera = 135
2  mika = 142
3  laza = 146
4
5  if pera >= 140:
6      print(pera, "Pera moze na tobogan")
7  if mika >= 140:
8      print(mika, "Mika moze na tobogan")
9  if laza >= 140:
10     print(laza, "Laza moze na tobogan")
11
```

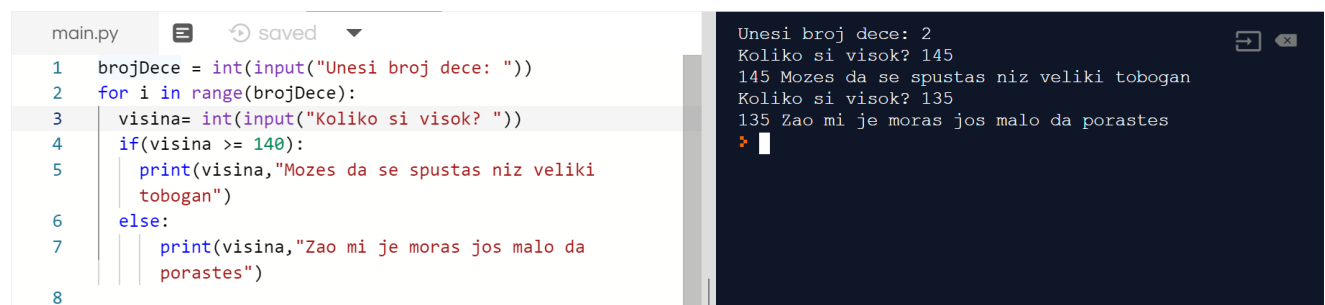
```
142 Mika moze na tobogan
146 Laza moze na tobogan
>
```

Слика 1

Задатак 2: Направити програм који симулира да ли одређено дете може да се спушта низ велики тобоган или не. Свако дете долази испред тобогана изговара своју висину и добија обавештење „Можеш да се спушташ низ велики тобоган“ ако је његова висина већа од 140 центиметара или обавештење „Жао ми је мораш још мало да порастеш“ ако је висина мања од 140 центиметара. Најпре се учитава број деце а затим висина за свако дете.

Решење:

(Слика 2)



```
main.py saved
1  brojDece = int(input("Unesi broj dece: "))
2  for i in range(brojDece):
3      visina= int(input("Koliko si visok? "))
4      if(visina >= 140):
5          print(visina, "Mozes da se spustas niz veliki tobogan")
6      else:
7          print(visina, "Zao mi je moras jos malo da porastes")
8
```

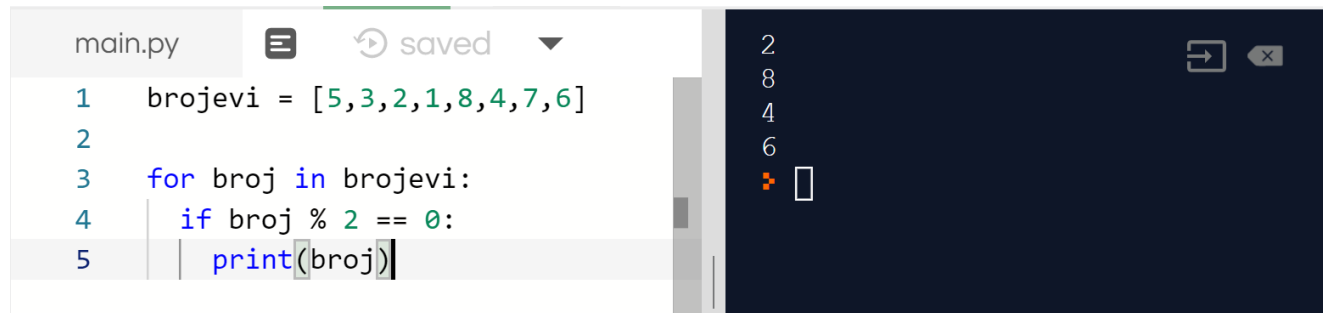
```
Unesi broj dece: 2
Koliko si visok? 145
145 Mozes da se spustas niz veliki tobogan
Koliko si visok? 135
135 Zao mi je moras jos malo da porastes
>
```

Слика 2

Задатак 3: Напиши програм, који за задату листу бројева исписује оне који су парни.

Решење:

(Слика 3)



The screenshot shows a Python IDE with a file named 'main.py'. The code in the editor is as follows:

```
1 brojevi = [5,3,2,1,8,4,7,6]
2
3 for broj in brojevi:
4     if broj % 2 == 0:
5         print(broj)
```

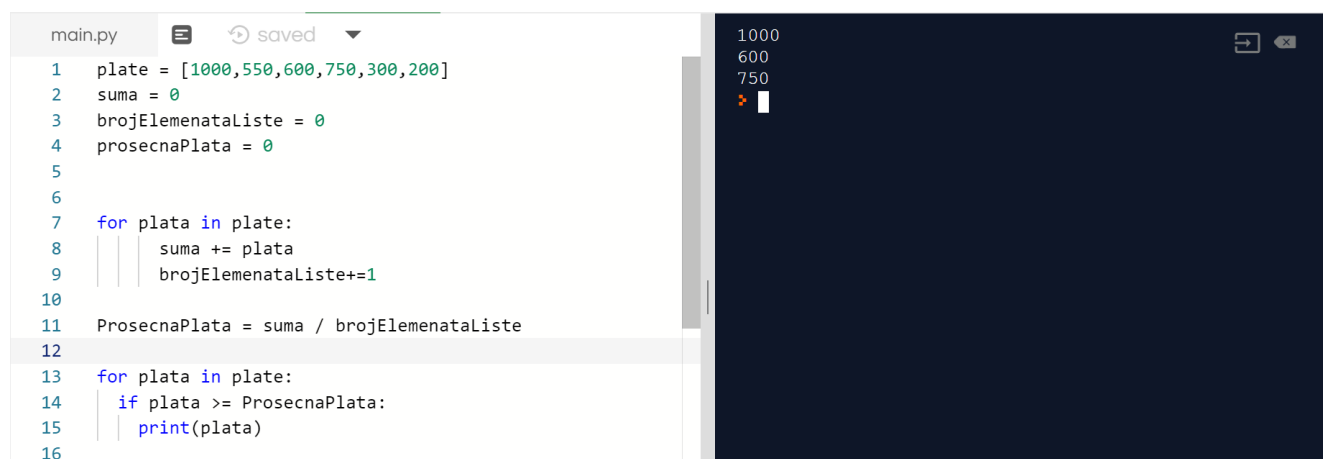
The output window on the right displays the results of the program execution, showing the even numbers from the list: 2, 8, 4, and 6, each on a new line.

Слика 3

Задатак 4: Написати програм који за дату листу плата, исписује само оне које су веће од просека.

Решење:

(Слика 4)



The screenshot shows a Python IDE with a file named 'main.py'. The code in the editor is as follows:

```
1 plate = [1000,550,600,750,300,200]
2 suma = 0
3 brojElemenataListe = 0
4 prosečnaPlata = 0
5
6
7 for plata in plate:
8     suma += plata
9     brojElemenataListe+=1
10
11 ProsečnaPlata = suma / brojElemenataListe
12
13 for plata in plate:
14     if plata >= ProsečnaPlata:
15         print(plata)
16
```

The output window on the right displays the results of the program execution, showing the salaries that are greater than or equal to the average: 1000, 600, and 750, each on a new line.

Слика 4

Филтрирање помоћу компрехенције

На пример, скуп позитивних елемената скупа целих бројева Z се може означити са:

$$\{x \mid x \in Z, x > 0\}$$

У језику Пајтон подржан је веома сличан запис:

```
{x for x in S if P(x)}
```

Означава скуп елемената x који припадају S и за које важи услов P . Слично,

```
[x for x in S if P(x)]
```

означава листу таквих елемената.

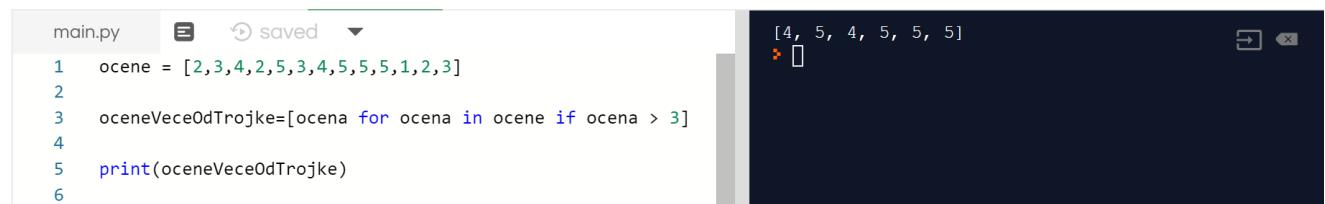
На пример, листа парних елемената из листе l се може изградити са:

```
[x for x in l if x % 2 == 0]
```

Задатак 5: Дата је листа са оценама ученика. Направити листу оних оцена које су веће од тројке.

Решење:

(Слика 5)



```
main.py  saved
1  ocene = [2,3,4,2,5,3,4,5,5,5,1,2,3]
2
3  oceneVeceOdTrojke=[ocena for ocena in ocene if ocena > 3]
4
5  print(oceneVeceOdTrojke)
6
```

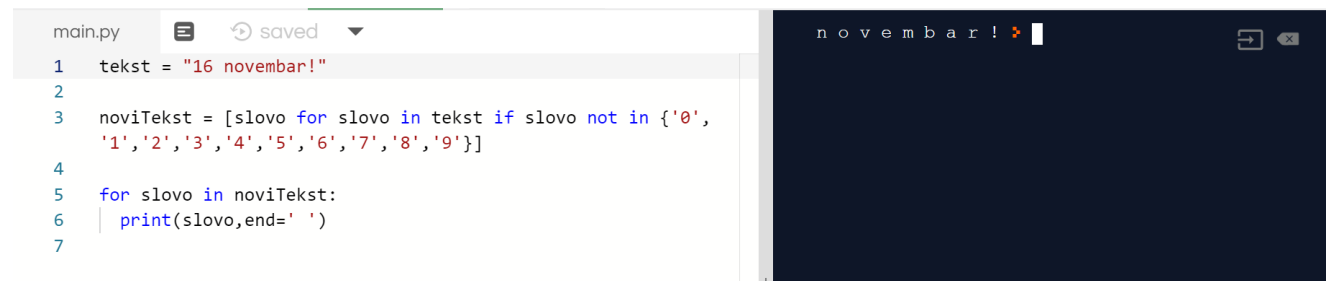
The output of the code is displayed in the console: [4, 5, 4, 5, 5, 5]

Слика 5

Задатак 6: Написати програм који би задати текст исписао тако што би изоставио сваку појаву броја.

Решење:

(Слика 6)



The screenshot shows a Python IDE with a file named 'main.py'. The code in the editor is as follows:

```
1 tekst = "16 novembar!"
2
3 noviTekst = [slovo for slovo in tekst if slovo not in {'0',
4               '1', '2', '3', '4', '5', '6', '7', '8', '9'}]
5
6 for slovo in noviTekst:
7     print(slovo, end= ' ')
```

The output window on the right shows the result of the program: "n o v e m b a r !".

Слика 6

Закључни део часа (5 мин)

Поново прокоментарисати са ученицима шта је филтрирање података. Разјаснити могуће нејасноће.