

Рачунари и софтвер

Програмирање омогућава да се рачунари (хардвер) користи да би се реализовао одређени задатак.

Програмер (softver developer) је особа која има вештине и знање неопходне за дизајнирање, креирање и тестирање компјутерских програма.

Хардвер чине сви физички делови (компоненте) рачунарског система (CPU, меморија, улазни и излазни уређаји).

Процесори су у виду чипова микропроцесора и задужени за сву обраду података који су ушли у рачунар преко улазних уређаја.

Меморија служи са чување података и резултата обраде података.

Улазни и излазни уређаји служе за унос и приказ резултата обраде података.

Софтвер чине програми који су смештени у меморији рачунара и који се састоје од низа инструкција које процесор обрађује и извршава.

Софтвер може бити системски (оперативни систем, антивируси, алати за програмирање) или апликативни (Office пакет).

Програми и програмирање

У рачунару се подаци смештају у виду низа 0 и 1 које представљају бит информације.

Слова су такође низ битова пошто се свако слово претвара у облик броја помоћу ASCII (American Standard Code for Information Interchange) кода (нпр слово А има бројну вредност 65).

Низ битова процесор препознаје као податке пошто је његов језик машински језик (низ битова).

Такође процесор добија на своје улазе програмске инструкције у виду низа битова, извршава их и на своје излазе шаље низ битова који престављају резултате унетих инструкција.

Овај процес се назива дохвати-декодуј-изврши (fetch-decode-execute) циклус.

Пошто је низ инструкција тешко писати у машинском језику измишљени су разни програмски језици да олакшају процес писања програма или програмирања.

Прво је осмишљен асемблерски језик а затим и низ програмских језика вишег нивоа.

Сваки програмски језик има посебне команде или службене речи које се могу користити само на тачно одређен начин.

Службене речи за програмски језик Пајтон:

<code>and</code>	<code>del</code>	<code>from</code>	<code>None</code>	<code>True</code>
<code>as</code>	<code>elif</code>	<code>global</code>	<code>nonlocal</code>	<code>try</code>
<code>assert</code>	<code>else</code>	<code>if</code>	<code>not</code>	<code>while</code>
<code>break</code>	<code>except</code>	<code>import</code>	<code>or</code>	<code>with</code>
<code>class</code>	<code>False</code>	<code>in</code>	<code>pass</code>	<code>yield</code>
<code>continue</code>	<code>finally</code>	<code>is</code>	<code>raise</code>	
<code>def</code>	<code>for</code>	<code>lambda</code>	<code>return</code>	

Сваки програмски језик има и операторе који извршавају различите операције над подацима (операндима).

Синтакса и семантика

Програмски језик има своју синтаксу и семантику.

Синтакса је скуп правила који се морају стриктно поштовати приликом писања програма.

Синтаксна правила указују како се морају користити службене речи, оператори и други симболи у програму.

Синтаксна правила су граматичка правила програмског језика: како је нешто написано.

Синтаксна правила су фиксна и непромењива током кодовања и зато програмер мора да их познаје.

Семантика је наука која изучава значење исписаних линија кода програмског језика.

Дешава се да је код правилно синтаксно написан али није семантички па се програм не даје добре резултате.

Семантичка правила су граматичка правила програмског језика: зашто је нешто написано.

Семантика указује на логичност, сврсисходност и ефикасност делова кода у програму.

Семантичке грешке се често не пријављују као грешке при куцању кода већ се откривају тек после извршења програма.

Појединачне инструкције које се користе за писање програма се називају искази (statements).

Исказ у програмирању се састоји од службених речи, оператора, симбола и других елемената који ако су правилно поређани реализују операцију и дају очекивани резултат.

Компајлери и интерпретатори

Пошто процесор разуме само машински језик (низ 0 и 1), програми написани у програмским језицима морају да се преводе на машински језик пре њиховог извршавања.

Компајлер је програм који преводи цео програм написан у програмском језику у посебан програм на машинском језику.

Пајтон користи интерпретер (преводилац) који преводи инструкције (делове програма) у машински језик и одмах их и извршава.

Компајлер реализује компајлирање (превођење) и извршавање кода у два одвојена корака.

Интерпретери комбинују превођење и извршавање програма инструкција по инструкција па зато они ни не праве посебан програм у машинском коду.

Компајлери су нешто спорији, док интерпретери не чувају међурезултате кодова у програму.

Изворни код (source code) или само код, су искази које програмер пише у програмском језику.

Ови исказу се преводе преко компајлера или интерпретера у машински код који процесор извршава.

Али ако постоји грешка у синтакси кода, јавља се синтаксна грешка које не дозвољава да се код до краја изврши.

Тада програмер мора да очистио код од синтаксних грешака да би се код могао извршити.

Пајтон окружење

Пајтон је објектно оријентисани, интерпретирани програмски језик који се може користити на многим пословима у облику од кратких скрипти до кода за целокупне апликације. Да би се састављали програми у програмском језику Пајтон неопходно је инсталирати апликације које то омогућавају.

Апликације који су потребне за куцање кода и креирање програма су Пајтон интерпретер и Пајтон едитор.

За потребе наставе користиће се Microsoft Visual Studio 2017 Community edition, са инсталираним Пајтон модулима.

Ова апликација омогућава истовремено рад са Пајтоном у интерпретерском и у скрипт моду.

Пајтон интерпретер

Интерпретер омогућава да се унесе команде у једној линији у програмском језику Пајтон.

Прва слободна линија почиње са >>>. Овај низ симбола се назива **промпт** (command prompt). То је знак да је интерпретер спреман да прихвати укуцавање команде у новој линији. По извршењу задате наредбе, поново се појављује >>>.

За овакав начин рада се каже да је секвенцијалан – појави се >>>, укуца се линија са командама, притисне се ЕНТЕР, добије се резултат, појави се >>> итд.

```
Python 3.6 (64-bit) interactive
>>> 5 + 5
10
>>> 3 - 5
-2
>>> print 2 + 5
File "<stdin>", line 1
  print 2 + 5
  ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print(2 + 5)?
>>> print (2 + 5)
7
>>>
```

Задатак 012: Коришћење интерпретера

Пример 01) Куцање броја у интерпретеру

Откуцати следећи ред:

```
>>> 12345
```

Добија се резултат:

```
12345
>>> |
```

Из примера се види да куцање бројева и притисак на ЕНТЕР даје као резултат испис у новом реду укуцаног броја, прелазак у још један нови ред и постављање треперећег курсора на крају.

Пример 02) Куцање неслужбене речи у интерпретеру

```
>>> hvala
```

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'hvala' is not defined
>>>
```

Из примера се види да куцањем непознатог низа знакова, који нису под наводницима или апострофом, изазива реакцију у виду поруке о грешци; Пајтон не препознаје укуцани низ знакова.

Пример 03) Куцање службене речи у интерпретеру

```
>>> class
      ...
      File "<stdin>", line 1
      class
        ^
      SyntaxError: invalid syntax
```

Сада је откуцана службена реч, коју интерпретер препознаје и која има посебно значење у Пајтон програмском језику. Проблем је што она није коришћена на начин који синтакса Пајтон програмског језика захтева па је добијена порука о погрешној синтакси (граматичкој нетачности).

Задатак 013: Рад у интерпретеру

1. Укуцати своје име и презиме у интерпретеру и притиснути ЕНТЕР. Објаснити шта се појавило на екрану.
2. Укуцати празно место па притиснути ЕНТЕР. Поновити поступак два пута. Шта се дешава на екрану интерпретера ?
3. а) Укуцати print и притиснути ЕНТЕР.
б) Укуцати input и притиснути ЕНТЕР.
в) Укуцати crazy и притиснути ЕНТЕР.
Објаснити шта се десило после сваког уноса и зашто су коментари различити.
4. Укуцати "tekst" и притиснути ЕНТЕР. Укуцати 'tekst' и притиснути ЕНТЕР. Укуцати tekst и притиснути ЕНТЕР. Објаснити добијене излазе у интерпретеру.
5. Укуцати 'tekst''tekst' и притиснути ЕНТЕР. Укуцати "tekst""tekst" и притиснути ЕНТЕР. Да ли постоје разлике у резултатима и зашто?
6. Укуцати 'tekst' , 10 празних места, 'tekst' и притиснути ЕНТЕР. Како објашњавате добијени резултат ? Да ли интерпретер види празна места (белине) ?

Задаци и питања за утврђивање знања: Провежбати следеће задатке:

Инсталирати Microsoft Visual Studio Community 2017 са потребним Пајтон пакетом и пријавити се помоћу свог Microsoft аcount. Стартовати MS Visual Studio 2017 и провежбати рад у Пајтон интерактивном моду. За помоћ у коришћењу ових апликација користити материјал са школског сајта: [uvod u visual studio 2017](#) и [uputstvo za kurs python i igre](#).