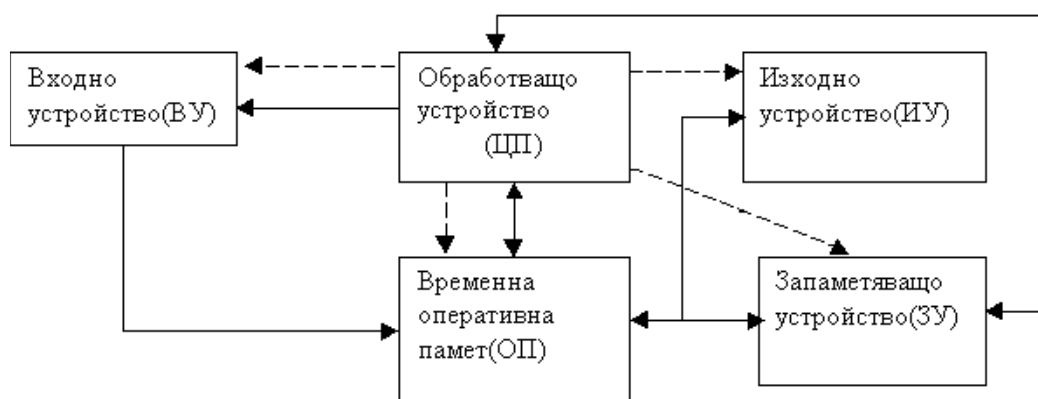


**Основни понятия. Функционална схема на компютър. Операционна система.
Езици за програмиране – класификация и предназначение.**

Компютърът е сложна електронно изчислителна машина, състояща се от отделни модули (елементи, части), чието общо предназначение е обработка на информация, (данни), представени в цифров вид. Тази обработка се извършва съгласно някакъв алгоритъм, представен в компютъра като програма.

Програмата е последователност от инструкции, които водят до решаването на някакъв конкретен проблем или клас от проблеми.

Функционална схема на персонален компютър:



Централният процесор (ЦП) управлява работата на всички останали устройства и изпълнява програмата, намираща се в паметта на компютъра.

Оперативната памет (ОП) е бърза памет, в която се съхраняват активните програми и данните, които те използват. Тя е енергозависима. При спиране на захранването, нейното съдържание се губи.

Запаметяващото устройство (ЗУ) е постоянната памет, която съхранява данните и програмите за тяхната обработка след изключване на захранването на компютъра. Най-често това е т.нар. "твърд диск" (хард диск).

Входното устройство (ВУ) служи за въвеждане на данните. Обикновено това са клавиатура и мишка.

Изходното устройство (ИУ) извежда обработените данни в подходящ вид. Обикновено това е монитор.

Основни концепции при работата на компютър с **еднопроцесорна архитектура**:

- * компютърът трябва да бъде електронно устройство и да използва двоичен код
- * структурата на компютъра трябва да включва аритметико-логическо устройство, устройство за управление, памет и устройства за въвеждане и извеждане на информация;
- * компютърът трябва да работи под управлението на програма, състояща се от инструкции, която се съхранява заедно с данните в една и съща линейна едномерна последователно адресируема памет;

- * съдържанието на паметта се адресира чрез номерата на отделните клетки, без да се взема предвид типа на данните в тях;
- * докато данните и инструкциите са в паметта, няма явно различие между тях, те се идентифицират само при изпълнение на програмите;
- * компютърът трябва да изпълнява инструкциите последователно една след друга, след прочитането им от паметта.

Програмата, записана в паметта на компютъра, представлява последователност от **машинни команди**, които може да изпълнява ЦП. Машинните команди са със стандартизиран формат, което позволява бързото и лесно дешифриране и подготовка за изпълнение. Пълният формат на машинните команди съдържа пет полета и има вида:

КОП	a	b	c	d
-----	---	---	---	---

КОП – код на операция;

a – адрес в паметта на първия аргумент;

b – адрес в паметта на втория аргумент;

c – адрес на резултата;

d – адрес на следващата команда.

Адресът на клетката е уникален номер, чрез който нейното съдържание може да се достигне от компютърната програма.

Компютрите могат да работят и да бъдат използвани само ако са снабдени със съответното **програмно осигуряване**. От своя страна то се разделя на две групи:

- * системно - използва се за управление на компютърните модули (Операционна Система)
- * приложно - специализирани програмни продукти за решаване на задачи от всекидневната дейност като текстообработка, създаване на изображения, работа с електронни таблици и др.

Операционната система (ОС) е сред най-сложните софтуерни продукти. Тя е посредник между човека-потребител и апаратните ресурси на компютъра.

ОС е съвкупност от програми, които се грижат за управлението на всички софтуерни и хардуерни компоненти в една компютърна система, както и за взаимодействието помежду им. Тя се грижи за разпределяне на ресурсите в системата, и за управление на достъпа до тях на отделните програми.

ОС има следните основни функции:

- 1) Осигуряване на среда за изпълнение на потребителските програми.
- 2) Конкретното управление на апаратните и информационните ресурси на компютърната система.
- 3) Интерфейс (връзка) между компютърната система и потребителя.

Най-общо казано **езикът за програмиране** служи като посредник между човека (програмист) и машината (компютър). Програмистът използва езика за програмиране, за да "накара"

компютъра да извърши серия от действия (инструкции). Тези серии от инструкции се наричат програми.

Съществуващите до сега езици за програмиране могат да се класифицират според хронологията на създаването им по следния начин:

I. Езици от ниско ниво

Машинен език - Машинният код е двоичен и е съставен от серия от инструкции, кодирани чрез нули и единици. Това е единственият "език", разбираем за компютрите, но не и за хората. На практика писането на машинен език е крайно неудобно и по тази причина се избягва.

Асемблер - превежда инструкциите, които един компютър е способен да изпълнява, във вид разбираем за хората. Различните инструкции на даден процесор се представят чрез кратки символни съкращения, а не чрез двоичен код. Асемблер е процесорно-зависим език за програмиране и причината за това е, че различните процесори имат различни инструкции. Тоест програма писана за една процесорна архитектура може и да не работи на компютър с различна процесорна архитектура.

II. Езици от високо ниво

- Процедурно-ориентирани езици - ползват думи от английския език и познатите математически символи. При процедурното програмиране кодът се разделя на няколко модула/пакета. Всеки модул е съставен от една или повече подпрограми. Една подпрограма може да е съставена от функции, процедури или методи - в зависимост от възприетата терминология за съответния език за програмиране. Такива езици са Fortran, Algol, (най-ранните) Basic, Pascal, C и др.;
- Обектно-ориентирани езици – те са от най-високо ниво до момента. Съдържат стандартни визуални обекти, които се използват при създаване на програмите. В най-общи линии това са езици, при които софтуерът е съставен от обекти, които отразяват взаимовръзките и свойствата на използваните структури, а не толкова тяхната обвързаност с компютърния код. Такива езици са: Python, Visual Basic, Delphi, C++, JAVA, C# и др.

Според своята приложимост езиците за програмиране се разделят на две групи:

- Универсални - те са подходящи за решаване на широк кръг задачи и за програмиране на всеки компютърен алгоритъм. Представители: Fortran, Pascal, C/C++, Java и др.
- Специализирани - подходящи са за решаването на специализирани задачи. Представители: езици за бази данни (SQL), за мултимедийни приложения (HTML) и др.

Транслатори

Транслаторите са компютърни програми, които преобразуват програма, написана на език от високо ниво, на машинен език. Всеки език, всяка негова версия имат свой транслатор. Така работата по превода на машинен език е предоставена на самия компютър, който изпълнява програмата-транслатор.

Видове транслатори:

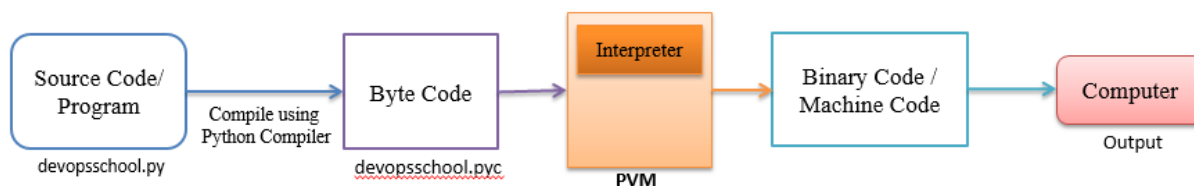
- Интерпретатори. При тях превода и изпълнението на програмата протичат успоредно. Всеки оператор се прекодира до машинни команди, които веднага се изпълняват. След това се преминава към следващия оператор. Програмата се превежда всеки път при стартирането и за изпълнение. Машинния превод не се запазва.
- Компилатори. Първо се прекодират всички оператори, за да се получи цялостна изпълнима програма. Тя се записва и може многократно да се изпълнява.

Език за програмиране Python:

- Python е интерпретируем език. Той няма компилатор в чист вид, който да произведе машинен код. При своето първо изпълнение програмният код на Python програма (файл с разширение .py) първо се компилира до междинен байткод (файл с разширение .pyc), който след това се превръща в машинни команди от интерпретатор - виртуална Питон машина (Virtual Python Machin). Машинните команди се четат и изпълняват от компютъра директно и резултатите се показват на потребителя. Ако в процеса на изпълнение възникне грешка, то изпълнението се спира със съобщение за грешка, което се показва на потребителя. Байткодът се генерира при първото изпълнение на програмата и може да бъде запазен в кеша за следващи изпълнения на програмата. Това повишава значително ефективността на процеса на разработка и тестване на дадено приложение.

История на Python:

Python е обектно ориентиран език за програмиране, създаден от Гуидо ван Росум в началото



на 90-те години на XX век. Кръстен е на телевизионното шоу на BBC „Monty Python’s Flying Circus“. От тогава до днес той се е утвърдил като един от най-широко използваните езици за програмиране поради редицата си очевидни предимства:

- Това е език с общо предназначение и с него могат да се пишат много разнообразни приложения от съвсем прости до такива извършващи сложни научни анализи в области като системи с изкуствен интелект и машинно самообучение
- Интерпретаторът на Python е абсолютно безплатен и се разпространява свободно. За да го ползвате не е нужно да плащате нещо, но можете абсолютно легално да продавате създадените от вас програми.
- Има разработени Python интерпретатори за почти всички известни операционни системи (Windows, Linux, UNIX, MacOS, Android, iPhone OS и др.)
- Това е един от най-простите езици за програмиране. Една от основните идеи при създаването му е да има ясен и изчистен синтаксис, който да позволява лесна четимост и разбираемост на кода. Това го прави идеален избор за първи език за програмиране за начинаещи програмисти.
- Python позволява динамично определяне на типа на променливите. Типът не се декларира предварително, а се определя от записаното съдържание в променливата.

- За оформяне на блокове в кода на програмата се използва отстъпи от полето в дясно, а не специални символи като фигурни скоби например. Това прави синтаксиса на програмите много подреден.
- Python се разпространява с редица стандартни библиотеки, използването на който може много да разшири функционалността на разработваните приложения. Освен тези библиотеки има и множество разработени от независими разработчици, които след инсталация също лесно могат да бъдат внедрени в код на Python програми и да разширят още повече възможностите им.

От къде да вземем Python?

<https://www.python.org/downloads/>

Източници на информация за език за програмиране Python:

<https://www.w3schools.com/python/>

<https://www.python.org/doc/>

<https://realpython.com/>

КОЛИСНИЧЕНКО Д., Python практическо програмиране, Асеновци, Второ издание, 2019 г.

AMOS D., BADER D, A Practical Introduction to Python 3, Forth Edition, Real Python, 2020.

НАКОВ С. & КОЛЕКТИВ., Основи на програмирането с Python, 2018.

Първа програма на Python:

При инсталирането на Python интерпретатор заедно с него се инсталира и базовата му среда за разработка на програми, написани на този език. Тя се нарича IDLE (Integrated Development and Learning Environment). В средата за разработка IDLE пишем код, компилираме и изпълняваме програмите.

Задание: Напишете програма на Python, която поздравява потребителя като използвате средата IDLE.

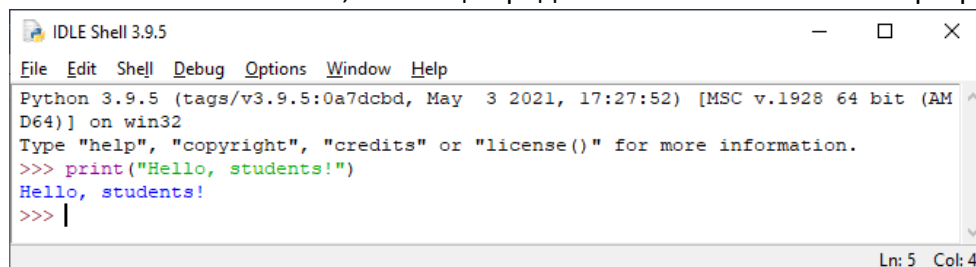
От Старт менюто на ОС Windows намираме и стартираме средата за разработки

Start-> IDLE(Python 3.x)

В отворения прозорец въвеждаме:

```
print("Hello, students!")
```

Натискаме клавиш Enter, което ще предизвика изпълнението на програмата:

The image shows a screenshot of the IDLE Shell 3.9.5 window. The title bar reads "IDLE Shell 3.9.5". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the following content: "Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32", "Type 'help', 'copyright', 'credits' or 'license()' for more information.", the command ">>> print('Hello, students!')", the output "Hello, students!", and the prompt ">>> |". The status bar at the bottom right indicates "Ln: 5 Col: 4".

```
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello, students!")
Hello, students!
>>> |
```

print() е вградена функция, която може да се използва за разпечатване на текстови съобщения върху екрана. Текстът на съобщението се загражда със специални маркери – кавички "Text" или единични апострофи 'Text'. В средата за разработки текстовете се оцветяват в зелено, имената на функциите – в лилаво, а резултатите от изпълнението на програмата – в синьо.