**Національний технічний університет України**

**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра обчислювальної техніки**

**Програмування**

**Лабораторна робота №1**

**«Загальні поняття про технічні характеристики**

**та програмне забезпечення сучасних комп’ютерів»**

Виконав:

студент групи ІО-63

Братун А.Ю.

Залікова книжка № \_\_\_\_\_

Перевірив Новотарський М. А.

Київ 2016 р.

**Тема:** «Загальні поняття про технічні характеристики та програмне забезпечення сучасних комп’ютерів».

**Мета:** вивчення основних складових частин персональних та професійних комп’ютерів, рівнів програмного забезпечення сучасних комп’ютерів. Інсталяція алгоритмічної мови Python та програмного інтегрованого середовища розробки PyCharm.

**Загальне завдання**:

1. Вивчити матеріал лекцій 1 та 2.

2. Здати комп’ютерний тест №1.

3. Інсталювати алгоритмічну мову програмування Python.

4. Інсталювати IDE PyCharm Edu.

**Короткі теоретичні відомості:**

**Комп’ю́тер** (англ.Computer — «обчислювач») — обладнання або система, здатна виконувати задану чітко визначену змінювану послідовність операцій. Це найчастіше операції чисельних розрахунків і маніпулювання даними, однак сюди відносяться й операції вводу- виводу.

За кожну з названих операцій відповідає спеціальне обладнання комп'ютера:

• обладнання (пристрої) вводу

, • системний блок,

• обладнання (пристрої) виводу.

**Архітектуру сучасного комп'ютера визначають наступні принципи:** 1. Принцип програмного керування. Забезпечує автоматизацію процесу обчислень на комп'ютері. Згідно із цим принципом, для розв'язування кожної задачі складається програма, яка визначає послідовність дій комп'ютера. 2. Принцип програми, що зберігається в пам'яті. Згідно із цим принципом, команди програми подаються, як і дані, у вигляді чисел і обробляються так само, як і числа, а сама програма перед виконанням завантажується в оперативну пам'ять, що прискорює процес її виконання. 3. Принцип довільного доступу до пам'яті. Відповідно до цього принципу, програми й дані можуть записуватися в довільне місце оперативної пам'яті, що дозволяє звернутися за будь-якою заданою адресою (до конкретної ділянки пам'яті) без перегляду попередніх

**Програма** – це запис алгоритму розв'язування задачі у вигляді послідовності команд або операторів мовою, яку розуміє комп'ютер. Кінцевою метою будь-якої комп'ютерної програми є керування апаратними засобами.

**Міжпрограмний інтерфейс –** це розподіл програмного забезпечення на декілька зв'язаних між собою рівнів. Рівні програмного забезпечення є пірамідою, де кожний вищий рівень базується на програмному забезпеченні попередніх рівнів

**Інтегроване середовище розробки, ІСP/IDE (англ. Integrated development environment)** — комплекс програмних засобів, що використовують програмісти для розробки програмного забезпечення (ПО).

**Компіля́ція** — трансляція програми, складеної початковою мовою високого рівня, в еквівалентну програму на низькорівневій мові, близькій до машинного коду (абсолютний код, об’єктний модуль, іноді на мову асемблера). Вхідною інформацією для компілятора (початковий код) є опис алгоритму або програми предметно- орієнтованою мовою, а на виході компілятора — еквівалентний опис алгоритму машинно-орієнтованою мовою (об'єктний код).

**Семантика** – це той «зміст», який програміст вкладає у свою програму. Це початкова й вкрай неформалізована частина процесу програмування. Саме в перекладі змісту у формальний опис і криється найбільша складність програмування.

**Синтаксис** – це засіб, за допомогою якого програміст викладає свій задум у вигляді, зрозумілому інтерпретатору.

**Синтаксис** – повністю формалізована частина: його можна описати формальною мовою синтаксичних діаграм (що й робиться в інструкції користувача). Прагматика – це ті дії, які виконує інтерпретатор з виконання завдання, представленого в синтаксично правильному вигляді. Виявом прагматики є сам інтерпретатор мови. Саме він читає записане відповідно до синтаксису "послання" і перетворює його в дії за закладеним в ньому алгоритмом.

**Транслятори бувають двох типів**:

• інтерпретатори,

• компілятори.

**Інтерпретатор** читає один оператор програми, аналізує його й відразу виконує, після чого переходить до обробки наступного оператора.

**Компілятор** спочатку читає, аналізує й переводить на машинний код усю програму й тільки після завершення всієї трансляції ця програма виконується.

**Інтегроване середовище розробки, ІСP/IDE** (англ. Integrated development environment) — комплекс програмних засобів, що використовують програмісти для розробки програмного забезпечення (ПО).

**Компіля́ тор** — програма або технічний засіб, що виконує компіляцію.

**Компіля́ ція** — трансляція програми, складеної початковою мовою високого рівня, в еквівалентну програму на низькорівневій мові, близькій до машинного коду (абсолютний код, об’єктний модуль, іноді на мову асемблера). Вхідною інформацією для компілятора (початковий код) є опис алгоритму або програми предметно- орієнтованою мовою, а на виході компілятора — еквівалентний опис алгоритму машинно-орієнтованою мовою (об'єктний код).

**Компілювати** — проводити трансляцію машинної програми з предметно-орієнтованої мови на машинно-орієнтовану мову.

**Інтерпрета́тор** — програма (різновид транслятора), що виконує інтерпретацію.

**Інтерпрета́ ція** — пооператорний (покомандний, порядковий) аналіз, обробка й відразу виконання початкової програми або запиту ( на відміну від компіляції, при якій програма транслюється без її виконання.

Мова Python підтримує три основні **парадигми** програмування:

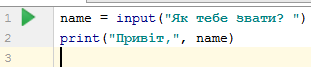
1. імперативне (процедурний, структурний, модульний підходи),

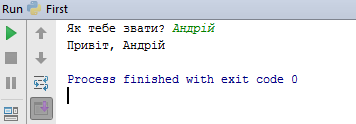
2. об'єктно-орієнтоване,

3. функціональне програмування.

**Літерал** (англ. literal) — запис у вхідному (початковому) коді комп'ютерної програми, що є фіксованим значенням. Літералами також називають представлення значення деякого типу даних.

**текст програми**

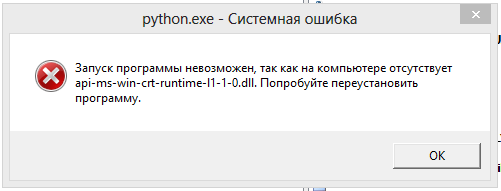
****

****

**Аналіз проблем, які виникли при виконанні лабораторної роботи**

**Проблема перша**

Труднощі виникли під час встановлення Python на ОС Windows 8, після встановлення , exe файл не відкривався через відсутність dll файлу



переглянувши доволі багато інформації в пошуковій системі Google, знайшлося багато рішень, деякі з них допомогли запустити exe файл без помилки у сусіда з моєї кімнати, але в мене помилка виникала знову і знову. Витративши ще багато часу та терпіння, спробувавши багато різних варіантів рішень, мені допомогло ось що:   
1. Зайти в «панель управления» далі в розділі програми і компоненти видалити всі версії Visual Studio

2. Закачати файл vc\_redist.x64 ( для 32-біт ОС скачати vc\_redist.x86) можна найти в пошуковій система або скачати за посиланням

<https://www.microsoft.com/ru-RU/download/details.aspx?id=48145> .

Встановити файл ( в мене зайняло це хвилин 25-35, на кінці завантаження буде написано що йде оновлення Windows, завантаження ніби «залагає», але потрібно чекати)

**Контрольні питання**

**ЗАПИТАННЯ 1**

1. **Блок живлення (БП)**. Це обладнання, що перетворює мережеву напругу змінного струму 220 вольтів у ряд низьких напруг постійного струму. За його охолодження відповідає вентилятор. Краще, якщо БП оснащений великим вентилятором (не менш, ніж 120 мм). Чим більший вентилятор, тем тихіше він працює й тем краще охолоджує. Блок живлення призначений для живлення материнської плати, плат розширення, вінчестера, CD/Dvd- дисководу, тобто всього обладнання системного блоку.

2. **Привод CD/Dvd-диска**. Оптичний привод — обладнання для запису й відтворення CD/Dvd-дисків. До нього надходять електричні шнури від блоку живлення й шина даних від материнської плати. DVDROM (RW) служить для читання й запису лазерних CD/Dvd-дисків. Компакт-диски можуть містити комп'ютерні програми й ігри, музичні твори, відеофільми і т. д.

3. **Процесор**. Це велика мікросхема, на яку кріпиться кулер (металевий радіатор з вентилятором). Процесор — головна мікросхема комп'ютера, призначена для виконання різних арифметичних і логічних операцій (інструкцій і програм). Установлюється процесор на материнську плату за допомогою спеціального конектора. 4.

**Материнська плата**. Системна (материнська) плата – це друкована плата, у яку встановлюються основні компоненти комп'ютера (процесор, пам'ять, контролери зовнішнього обладнання і конектори (зокрема, слоти) для підключення додаткового обладнання). Інакше кажучи, материнська плата — це велика друкована плата, на якій усе монтується, яка є сполучною ланкою між іншими пристроями ПК (центральним процесором, оперативною пам'яттю, вінчестером і т. д.). Від неї відходять шини даних до дисководу, жорсткого диска. До материнської плати підведена електрика від блоку живлення, шнури вентиляторів, шнури від лицьової частини системного блоку. Бюджетні комп'ютери зазвичай містять вбудовану звукову карту, яка дозволяє відтворювати звук, синтезований комп'ютером, а також записувати цифрований звук з якістю Cd-дисків.

5. **Відеокарта**. Відеокарта призначена для перетворення даних, вироблюваних комп'ютером, у сигнал, відображуваний монітором. У дешевих (бюджетних) ПК материнська плата вже містить вбудовану відеокарту. У більш продуктивних моделей ПК відеокарта купується окремо й установлюється на материнську плату, іноді до неї підведений шнур від блоку живлення. Кулер відеокарти кріпиться на ній же, при покупці відеокарти він найчастіше вже встановлений.

6. **Жорсткий диск (вінчестер)**. Пристрій для зберігання основної інформації на ПК. Вінчестер (накопичувач на жорстких дисках) є місцем зберігання операційної системи, програм, додатків, баз даних, ігор і т. д. На відміну від ОЗУ, при вимиканні живлення інформація, записана на вінчестер, не губиться.

7. **Оперативна пам'ять (ОЗУ)**. Це мікросхема, у якій тимчасово зберігаються дані й команди, необхідні процесору для виконання ним різних операцій. Пам'ять встановлюється у спеціальні слоти на материнській платі.

8**. Кулер системного блоку**. Установлюється в задній (іноді у верхній) частині системного блоку на гвинти або спеціальні гумові кріплення. Його шнур живлення підключають до материнської плати.

**ЗАПИТАННЯ 2**

**Міжпрограмний інтерфейс** – це розподіл програмного забезпечення на декілька зв'язаних між собою рівнів. Рівні програмного забезпечення є пірамідою, де кожний вищий рівень базується на програмному забезпеченні попередніх рівнів.

Схематично структура програмного забезпечення наведена на рис. 1. Прикладний рівень Службовий рівень Системний рівень Базовий рівень

**Базовий рівень програмного забезпечення**

Базовий рівень є найнижчим рівнем програмного забезпечення. Відповідає за взаємодію з базовими апаратними засобами. Базове програмне забезпечення міститься в складі базового апаратного забезпечення й зберігається в спеціальних мікросхемах постійного запам'ятовувального пристрою (ПЗУ), утворюючи базову систему вводу-виводу BIOS. Програми й дані записують в ПЗУ на етапі виробництва й не можуть бути змінені під час експлуатації.

**Системний рівень програмного забезпечення**

Системний рівень є перехідним. Програми цього рівня забезпечують взаємодію інших програм комп'ютера з програмами базового рівня й безпосередньо з апаратним забезпеченням. Від програм цього рівня залежать експлуатаційні показники всієї обчислювальної системи. При приєднанні до комп'ютера нового обладнання на системному рівні повинна бути встановлена програма, що забезпечує для інших програм взаємозв'язок з обладнанням. Конкретні програми, призначені для взаємодії з конкретним обладнанням, називають драйверами. Інший клас програм системного рівня відповідає за взаємодію з користувачем. Завдяки йому, можна вводити дані в обчислювальну систему, керувати її роботою й одержувати результат у зручній формі. Це засоби забезпечення користувацького інтерфейсу, від них залежить зручність і продуктивність роботи з комп'ютером. Сукупність програмного забезпечення системного рівня утворює ядро операційної системи комп'ютера. Наявність ядра операційної системи – це перша умова для можливості практичної роботи користувача з обчислювальною системою. Ядро операційної системи виконує такі функції: керування пам'яттю, процесами вводу-виводу, файловою системою, організація взаємодії й диспетчеризація процесів, облік використання ресурсів, обробка команд і т. д.

**Службовий рівень**

Програми цього рівня взаємодіють як з програмами базового рівня, так і з програмами системного рівня. Призначення службових програм (утиліт) полягає в автоматизації робіт з перевірки й настроювання комп'ютерної системи, а також для поліпшення функцій системних програм. Деякі службові програми (програми обслуговування) відразу входять до складу операційної системи, доповнюючи її ядро, але більшість є зовнішніми програмами й розширюють функції операційної системи. Тобто, у розробці службових програм відслідковуються два напрямки: інтеграція з операційною системою й автономне функціонування.

**Прикладний рівень програмного забезпечення**

Програмне забезпечення цього рівня є комплексом прикладних програм, за допомогою яких виконуються конкретні задачі (виробничі, творчі, розважальні і навчальні). Між прикладним і системним програмним забезпеченням існує тісний взаємозв'язок. Універсальність обчислювальної системи, доступність прикладних програм і широта функціональних можливостей комп'ютера безпосередньо залежать від типу наявної операційної системи, системних засобів, поміщених у її ядро й взаємодії комплексу людина- програма-устаткування.

**ЗАПИТАННЯ 3**

**приклади програм системного рівня**

До системного програмного забезпечення відносять програми, що забезпечують стабільну роботу ОС; засоби, що взаємодіють із апаратними складовими ОС (драйвери); засоби, що дозволяють оптимізувати роботу ОС, засоби для роботи із файловою системою, засоби для захисту даних від пошкоджень; а також засоби для розробки програмного забезпечення. Часто деякі із програм, що обслуговують ОС називають службовими програмами або утилітами (тут під абревіатурою ОС мається на увазі обчислювальна система).

Розглянемо детальніше основні види системного програмного забезпечення і дамо їх коротку характеристику.

**Диспетчери файлів** (файлові менеджери). За допомогою програм даного класу виконується більшість операцій, пов’язаних з обслуговуванням файлової структури: копіювання, переміщення і переіменування файлів, створення каталогів (папок), знищення файлів та каталогів, пошук файлів та навігація по файловій структурі. Базові програмні засоби, призначені для цього, як правило, входять у склад програм системного рівня і встановлюються разом із операційною системою. Однак для підвищення зручності роботи з комп’ютером більшість користувачів встановлює додаткові службові програми.

**Засоби стиснення даних (архіватори).** Цей клас програм призначений для роботи з архівами. Архівація даних спрощує їх зберігання за рахунок того, що великі групи файлів та каталогів зводяться в один архівний файл. При цьому підвищується і ефективність використання носіїв, оскільки архівні файли, як правило, мають підвищену щільність запису інформації. Архіватори часто використовують для резервного копіювання цінних даних.

**Засоби перегляду і відтворення**. На практиці для роботи з файлами потрібно завантажити їх у прикладну програму, за допомогою якої вони були створені. Це дає можливість переглядати документи та вносити у них зміни. Але в тих випадках, коли потрібно лише переглянути документи без редагування, зручно використовувати більш прості та універсальні засоби, що дозволяють переглядати документи різних типів. У випадках, коли мова йде про звукозапис чи відеозапис, замість терміну перегляд використовують термін відтворення документів.

**Засоби забезпечення комп’ютерної безпеки**. До цієї досить широкої категорії відносяться засоби пасивного та активного захисту даних від пошкодження, а також засоби захисту від несанкціонованого доступу, перегляду та зміни даних.

У якості засобів пасивного захисту використовують службові програми, призначені для резервного копіювання даних. Часто вони володіють і базовими властивостями диспетчерів **архівів (архіваторів).** У якості засобів активного захисту застосовують антивірусне програмне забезпечення. Для захисту даних від несанкціонованого доступу, їх перегляду та зміни служать спеціальні системи, основані на криптографії.

**Засоби діагностики**. Призначені для автоматизації процесів діагностики програмного та апаратного забезпечення. Вони виконують необхідні перевірки і видають зібрану інформацію у зручному та наглядному вигляді. Їх використовують не тільки для усунення неполадок, але і для оптимізації роботи комп’ютерної системи.

**Засоби оптимізації роботи ОС**. Призначенні для підвищення швидкодії ПК за рахунок більш ефективного використання ресурсів ОС.

Інструментальні засоби для розробки програмного забезпечення використо­вуються висококваліфікованими програмістами для створення нових прикладних програм, розширення функціональних можливостей вже існуючих програм, а також для виправлення допущених помилок.