

Práctica 1 CPLP

1) 1951 - 1955: Durante este período surgieron los primeros lenguajes de programación de bajo nivel, conocidos como lenguajes tipo assembly. Estos lenguajes permitían a los programadores escribir instrucciones directamente en lenguaje de máquina, lo que mejoró significativamente la eficiencia y velocidad de los programas.

1956 - 1960: En este período surgieron varios lenguajes de programación de alto nivel, que permitían a los programadores escribir programas utilizando una sintaxis más natural y cercana al lenguaje humano. Entre ellos destacan FORTRAN, ALGOL 58, ALGOL 60 y LISP. Estos lenguajes se caracterizan por su capacidad para realizar cálculos científicos y matemáticos complejos.

1961 - 1965: Durante este período surgieron lenguajes de programación como COBOL, SNOBOL y JOVIAL, que se enfocaron en resolver problemas empresariales y de defensa. También se continuó trabajando en el desarrollo de lenguajes de programación para cálculo científico, como ALGOL 60.

1966 - 1970: En este período se desarrollaron lenguajes de programación orientados a objetos como SIMULA 67, así como lenguajes de programación para procesamiento de matrices como APL. También se mejoró FORTRAN, creando FORTRAN 66 y posteriormente FORTRAN 77.

1971 - 1975: Este período se caracteriza por el surgimiento de lenguajes de programación de propósito específico como Pascal y Prolog, así como la aparición de C, que se convirtió en uno de los lenguajes de programación más utilizados en la actualidad. Además, se desarrolló el lenguaje de programación Scheme, que se enfoca en la programación funcional.

1976 - 1980: Durante este período surgieron lenguajes de programación orientados a objetos como Smalltalk y Ada. También se mejoró FORTRAN, creando FORTRAN 77, y se desarrolló el lenguaje de programación ML, que se enfoca en la programación funcional.

1981 - 1985: En este período se mejoró Smalltalk, creando Smalltalk 80, y se desarrolló el lenguaje de programación Turbo Pascal, que se enfoca en la programación estructurada. También se desarrolló Postscript, un lenguaje de programación utilizado en la impresión y el diseño gráfico.

1986 - 1990: Durante este período se mejoró FORTRAN, creando FORTRAN 90, y se desarrolló C++, que es una extensión del lenguaje C que permite la programación orientada a objetos. También se desarrolló el lenguaje de programación SML, que se enfoca en la programación funcional.

1991 - 1995: En este período surgieron lenguajes de programación para la web como TCL, PERL y HTML. TCL se utiliza para crear scripts en servidores web, PERL se enfoca en el

procesamiento de texto y HTML es el lenguaje de marcado utilizado en la creación de páginas web.

1996 - 2000: En este período surgieron lenguajes de programación para la web como Java, Javascript y XML. Java se enfoca en la programación de aplicaciones web y el desarrollo de applets, Javascript se utiliza para crear páginas web interactivas y XML se utiliza para describir datos.

2) Python: fácil de adaptarse, de aprender, de leer, más dinámico.

JS

3) Desde lo personal Python tiene muy buenas características.

4) Python es un lenguaje de programación de alto nivel, que se caracteriza por su sintaxis clara y concisa, lo que lo hace fácil de aprender y de leer. A continuación se describen algunos de los tipos de expresiones que se pueden escribir en Python, así como las facilidades provistas para la organización del programa:

Tipos de expresiones:

Números: enteros, flotantes, complejos.

Cadenas de caracteres: texto.

Booleanos: True o False.

Variables: nombres que se utilizan para almacenar valores.

Listas, tuplas y conjuntos: colecciones de valores.

Diccionarios: colecciones de pares clave-valor.

Expresiones condicionales: permiten realizar operaciones según una condición.

Expresiones aritméticas: operaciones matemáticas como suma, resta, multiplicación y división.

Expresiones lógicas: permiten realizar operaciones booleanas como AND, OR y NOT.

Funciones: permiten crear bloques de código reutilizables que realizan tareas específicas.

Facilidades para la organización del programa:

Módulos: permiten organizar el código en archivos separados, que pueden ser importados y utilizados en otros programas.

Funciones: permiten crear bloques de código reutilizables que realizan tareas específicas.

Clases: permiten definir estructuras de datos y métodos que operan sobre ellas.

Excepciones: permiten manejar errores y situaciones inesperadas en el programa.

Indentación: Python utiliza la indentación para indicar bloques de código, lo que ayuda a mejorar la legibilidad del código.

Atributos del ejercicio anterior que posee Python:

Python es un lenguaje de programación de alto nivel, lo que significa que tiene una sintaxis clara y concisa que lo hace fácil de aprender y de leer.

Python es un lenguaje de programación orientado a objetos, lo que significa que permite la programación orientada a objetos y la creación de clases y objetos.

Python tiene una gran cantidad de bibliotecas y módulos disponibles, lo que lo hace útil para una amplia gama de aplicaciones.

Atributos del ejercicio anterior que no posee Python:

Python no fue uno de los primeros lenguajes de programación de bajo nivel, como los lenguajes tipo assembly que surgieron en los años 50.

5) Ada es un lenguaje de programación diseñado específicamente para aplicaciones de sistemas embebidos, aviones, sistemas militares y otros sistemas críticos de alta seguridad y fiabilidad. A continuación se describen algunas de las características más relevantes de Ada:

Tipos de datos:

- Ada tiene un conjunto de tipos de datos estáticos y fuertemente tipados, que incluyen enteros, flotantes, caracteres, booleanos, entre otros.
- Ada también proporciona tipos definidos por el usuario, que permiten a los programadores definir sus propios tipos de datos personalizados.
- Los tipos de datos en Ada son altamente expresivos y pueden ser muy específicos, lo que los hace adecuados para aplicaciones críticas que requieren un alto nivel de precisión y control.

Tipos abstractos de datos - paquetes:

- Ada soporta la programación orientada a objetos y proporciona tipos abstractos de datos a través de paquetes.
- Los paquetes permiten al programador encapsular datos y funciones relacionados, proporcionando así una mayor modularidad y abstracción en el diseño del programa.
- Los paquetes también permiten la ocultación de información, lo que significa que se puede restringir el acceso a ciertos datos y funciones sólo a ciertos módulos.

Estructuras de datos:

- Ada proporciona una amplia variedad de estructuras de datos, incluyendo arrays, listas, conjuntos y registros.

- Ada también soporta estructuras de datos dinámicas, como arrays y listas dinámicas, que pueden crecer o reducirse durante la ejecución del programa.

Manejo de excepciones:

- Ada proporciona un mecanismo de manejo de excepciones robusto, que permite al programador controlar y gestionar situaciones excepcionales, como errores de tiempo de ejecución o fallos en la memoria.
- Los bloques de manejo de excepciones en Ada permiten al programador definir qué hacer en caso de que se produzca una excepción, lo que puede incluir la recuperación del error, la notificación al usuario o la terminación controlada del programa.

Manejo de concurrencia:

- Ada proporciona un modelo de concurrencia basado en el uso de procesos y tareas.
- Los procesos en Ada representan entidades separadas de ejecución, mientras que las tareas son procesos ligeros que comparten el mismo espacio de direcciones.
- Ada proporciona un conjunto completo de primitivas de sincronización para la gestión de la concurrencia, incluyendo semáforos, monitores y eventos.

6) Java fue creado originalmente por James Gosling y su equipo en Sun Microsystems a principios de la década de 1990. Fue diseñado para ser un lenguaje de programación seguro, portátil y de alto rendimiento que pudiera ejecutarse en cualquier plataforma de hardware. La intención principal de Java era ser utilizado en dispositivos electrónicos, pero finalmente se convirtió en un lenguaje popular para la programación de aplicaciones empresariales y de escritorio.

Java introdujo varios cambios importantes en la web, incluyendo la posibilidad de crear applets de Java, que son pequeñas aplicaciones de software que se ejecutan dentro del navegador web del usuario. Los applets de Java permitieron a los desarrolladores crear experiencias más ricas y dinámicas en la web, como animaciones interactivas y juegos en línea.

Java también introdujo la máquina virtual de Java (JVM), que permite que el código Java se ejecute en cualquier plataforma que tenga una implementación de la JVM. Esto significa que los desarrolladores pueden escribir una vez su código en Java y luego ejecutarlo en cualquier plataforma, lo que ha hecho que Java sea un lenguaje muy popular para aplicaciones empresariales que se ejecutan en múltiples plataformas.

Sí, Java es un lenguaje dependiente de la plataforma en la que se ejecuta, lo que significa que el código Java debe ser compilado específicamente para la plataforma de destino. Sin embargo, la portabilidad de Java se logra mediante el uso de la JVM, que es una capa de abstracción de hardware que permite que el mismo código Java se ejecute en diferentes plataformas sin necesidad de ser recompilado.

7) Java está basado en varios lenguajes, incluyendo C++, Smalltalk y Objective-C. Los creadores de Java, liderados por James Gosling en Sun Microsystems, tomaron como base los conceptos y características de estos lenguajes para diseñar uno nuevo que tuviera un enfoque en la portabilidad, la seguridad y la facilidad de programación. Además, Java también incorpora ideas de otros lenguajes como Eiffel, Modula-2 y Mesa. En resumen, Java es un lenguaje de programación que se inspiró en una variedad de otros lenguajes y adoptó lo mejor de cada uno para crear un lenguaje moderno y altamente popular.

8) Los applets son pequeñas aplicaciones de software escritas en Java que se ejecutan en un navegador web y se utilizan para agregar interactividad y dinamismo a las páginas web. Los applets pueden ser utilizados para crear animaciones, juegos, gráficos, formularios interactivos, y otras funciones. Los applets de Java fueron muy populares en los primeros días de la web, ya que permitían a los desarrolladores crear contenido web dinámico y atractivo sin la necesidad de utilizar tecnologías propietarias de un solo proveedor.

Por otro lado, los servlets son componentes de software escritos en Java que se ejecutan en un servidor web y procesan solicitudes de clientes HTTP (por ejemplo, navegadores web). Los servlets son utilizados para generar contenido dinámico y personalizado para los usuarios que visitan un sitio web. Los servlets permiten a los desarrolladores crear aplicaciones web complejas y escalables utilizando el lenguaje de programación Java.

En resumen, los applets son aplicaciones de software escritas en Java que se ejecutan en el navegador web del usuario, mientras que los servlets son componentes de software escritos en Java que se ejecutan en un servidor web y procesan solicitudes de clientes HTTP. Ambos son herramientas útiles en la creación de aplicaciones web en Java, pero su uso ha evolucionado con el tiempo y en la actualidad se utilizan tecnologías más modernas como JavaScript y las APIs de servidor.

9) La estructura básica de un programa en C es la siguiente:

```
#include <stdio.h> // Biblioteca estándar de entrada/salida

int main void // Función principal

    // Código del programa
    return 0; // Valor de retorno
}
```

En la primera línea se incluye la biblioteca estándar de entrada/salida (**stdio.h**), que proporciona funciones para leer y escribir datos desde/hacia la consola y archivos. A continuación, se define la función principal (**main**) que es el punto de entrada del programa.

Esta función devuelve un valor entero (**int**) que se utiliza para indicar el estado de finalización del programa (0 significa que todo salió bien).

Dentro de la función **main** se escribe el código del programa, que puede incluir declaraciones de variables, operaciones aritméticas, llamadas a funciones, estructuras de control de flujo (por ejemplo, **if**, **while**, **for**, **switch**), y otros elementos de programación.

Sí, en C es posible anidar funciones dentro de otras funciones. Esto significa que una función puede ser definida dentro de otra función, y luego ser llamada desde dentro de la función contenedora. Sin embargo, el anidamiento de funciones no es muy común en la práctica, ya que puede hacer que el código sea más difícil de leer y mantener.

10) En C, una expresión es una combinación de valores, variables, operadores y llamadas a funciones que se evalúa para producir un resultado. Las expresiones pueden ser utilizadas en diferentes contextos, como asignaciones de variables, parámetros de funciones, condiciones de estructuras de control, entre otros.

El lenguaje C proporciona una amplia variedad de operadores para construir expresiones, incluyendo operadores aritméticos (+, -, *, /, %), operadores de comparación (<, >, <=, >=, ==, !=), operadores lógicos (&&, ||, !), operadores bit a bit (&, |, ^, <<, >>), entre otros.

Además, en C es posible realizar conversiones entre diferentes tipos de datos utilizando operadores de casting. Por ejemplo, si se quiere convertir un número entero a un número en coma flotante, se puede hacer lo siguiente:

```
int x = 5
float y = float
```

El manejo de expresiones en C es muy flexible y permite construir expresiones complejas utilizando diferentes operadores y llamadas a funciones. Sin embargo, es importante tener en cuenta la precedencia de los operadores y el orden de evaluación de las expresiones para evitar errores de programación.

11) Python, Ruby y PHP son lenguajes de programación de propósito general que pueden ser utilizados para desarrollar diferentes tipos de programas, desde scripts simples hasta aplicaciones complejas. A continuación, se describen algunos de los tipos de programas que se pueden escribir con cada uno de estos lenguajes:

Python:

- Aplicaciones web: Python es ampliamente utilizado en la creación de aplicaciones web, gracias a frameworks como Django, Flask o Pyramid.
- Procesamiento de datos: Python es una opción popular para el procesamiento y análisis de datos, gracias a bibliotecas como NumPy, Pandas o Scikit-learn.

- Desarrollo de juegos: Python es utilizado para el desarrollo de juegos, gracias a frameworks como Pygame.
- Automatización de tareas: Python es una excelente opción para la automatización de tareas y scripts de sistema.

Python es un lenguaje de programación multiparadigma que admite diferentes estilos de programación, incluyendo programación orientada a objetos, programación funcional y programación imperativa.

Ruby:

- Aplicaciones web: Ruby es conocido por su uso en el desarrollo web, gracias a frameworks como Ruby on Rails y Sinatra.
- Desarrollo de juegos: Ruby también es utilizado en el desarrollo de juegos, gracias a frameworks como Gosu.
- Automatización de tareas: Al igual que Python, Ruby es una buena opción para la automatización de tareas.

Ruby es un lenguaje de programación orientado a objetos que admite programación funcional y programación imperativa.

PHP:

- Aplicaciones web: PHP es uno de los lenguajes más utilizados en el desarrollo de aplicaciones web, gracias a frameworks como Laravel, Symfony o CodeIgniter.
- Desarrollo de CMS: PHP es ampliamente utilizado en la creación de sistemas de gestión de contenidos como WordPress, Joomla o Drupal.
- Procesamiento de formularios: PHP es utilizado en la recepción y procesamiento de formularios HTML.
- Automatización de tareas: PHP también puede ser utilizado para la automatización de tareas y scripts de sistema.

PHP es un lenguaje de programación imperativa con características de programación orientada a objetos. Aunque originalmente fue diseñado para el desarrollo de aplicaciones web, en la actualidad se puede utilizar para desarrollar diferentes tipos de aplicaciones.

- 12) Python:
 - Tipado dinámico: no es necesario declarar el tipo de datos de las variables.
 - Alta legibilidad del código debido a su sintaxis limpia y clara.
 - Soporte para múltiples paradigmas de programación, incluyendo programación orientada a objetos, programación funcional y programación procedural.
 - Amplia biblioteca estándar que facilita tareas comunes de programación.
 - Interprete interactivo que permite probar y ejecutar el código de manera interactiva.

- Es utilizado en una amplia variedad de áreas como el desarrollo web, el análisis de datos, la inteligencia artificial, entre otros.
- Ruby:
 - Tipado dinámico: no es necesario declarar el tipo de datos de las variables.
 - Orientado a objetos: todo en Ruby es un objeto.
 - Sintaxis simple y expresiva que favorece la legibilidad del código.
 - Metaprogramación: permite modificar el comportamiento de los objetos en tiempo de ejecución.
 - Soporte para programación funcional y programación procedural.
 - Frameworks populares como Ruby on Rails para el desarrollo web.
- PHP:
 - Tipado débil: no es necesario declarar el tipo de datos de las variables.
 - Diseñado específicamente para la creación de sitios web dinámicos y aplicaciones web.
 - Soporte para múltiples bases de datos, incluyendo MySQL y PostgreSQL.
 - Amplia biblioteca estándar y gran cantidad de frameworks para el desarrollo web como Laravel, Symfony, CodeIgniter, entre otros.
 - Lenguaje de script del lado del servidor, lo que permite la generación de contenido HTML dinámico.
- Golang:
 - Lenguaje diseñado específicamente para la enseñanza de programación.
 - Basado en el paradigma de programación imperativa.
 - Sintaxis sencilla y legible que facilita su aprendizaje.
 - Entorno gráfico que facilita la visualización y la comprensión del código.
- Processing:
 - Lenguaje diseñado para la creación de gráficos y animaciones interactivas.
 - Basado en Java, pero con una sintaxis simplificada que lo hace más accesible para artistas y diseñadores.
 - Entorno de desarrollo integrado (IDE) que incluye una amplia biblioteca de funciones y herramientas gráficas.
 - Multiplataforma: puede ser ejecutado en diferentes sistemas operativos, incluyendo Windows, macOS y Linux.

13) JavaScript es un lenguaje de programación que soporta múltiples paradigmas, entre ellos el paradigma orientado a objetos, el paradigma funcional y el paradigma de programación imperativa.

En cuanto a su clasificación, JavaScript es un lenguaje de programación interpretado, ya que no requiere de un proceso de compilación previo para poder ejecutarse. Además, es un lenguaje de programación de alto nivel, es decir, que posee una sintaxis más cercana al lenguaje humano que al lenguaje de máquina.

- 14) Tipado dinámico: en JavaScript, el tipo de una variable puede cambiar durante la ejecución del programa.
- Orientado a eventos: en JavaScript, se pueden manejar eventos y realizar acciones en función de ellos.
- Funciones de primera clase: las funciones en JavaScript son tratadas como cualquier otro tipo de dato y pueden ser asignadas a variables, pasadas como argumentos a otras funciones o retornadas como resultado de una función.
- Sintaxis similar a otros lenguajes: la sintaxis de JavaScript está influenciada por otros lenguajes de programación como C y Java, lo que hace que sea fácil de aprender para aquellos que ya tienen experiencia en programación.
- Objetos y prototipos: en lugar de tener clases como en la programación orientada a objetos tradicional, JavaScript utiliza objetos y prototipos para definir la estructura y el comportamiento de los objetos.
- Excepciones: en JavaScript se pueden manejar excepciones con bloques try-catch, lo que permite controlar errores y continuar la ejecución del programa en caso de que ocurran errores.
- Variables: en JavaScript, las variables se declaran con la palabra clave "var", aunque también se pueden declarar con "let" y "const" en versiones más modernas del lenguaje.