

# Práctica 3

## Tipos de datos en Picky

Grado en Ingeniería en Sistemas Audiovisuales y Multimedia

GSyC, Universidad Rey Juan Carlos

22 de septiembre de 2017

### 1. Introducción

En esta práctica vamos a aprender a manejar los distintos *apartados* de un programa en el lenguaje Picky. Declararemos *constantes* y trabajaremos con ellas mediante el uso de *operadores* y *funciones*.

Para realizar esta práctica crearemos un archivo `apartados.p` en el que implementaremos el código de nuestro programa. El programa va a contener los distintos apartados:

- Comentario con autor y fecha.
- Nombre del programa.
- Definir tipos.
- Constantes.
- Funciones.
- Programa principal.

1. Al principio del programa incluiremos nuestro nombre y apellidos como autores del programa:

```
/*  
 * Segundo programa en Picky.  
 * Autor: pon tu nombre aquí.  
 * Fecha: fecha de implementación (DD/MM/AA)  
 */
```

2. El siguiente paso es incorporar el nombre de nuestro programa:

```
/*  
 * Nombre del programa.  
 */  
program nombreprograma;
```

3. Crearemos el apartado para definir los tipos:

```
/*  
 * Tipos de datos  
 */  
types:
```

4. Crearemos el apartado para las constantes:

```
/*  
 * Constantes  
 */  
const:
```

5. Creamos el apartado de funciones:

```
/*  
 * Funciones y procedimientos.  
 */
```

6. Creamos el apartado del programa principal

```
/*  
 * Programa principal.  
 */  
procedure main()  
/*Espacio para variables globales*/  
  
{  
  
}
```

## 1.1. Variables vs Constantes

En los lenguaje de programación nos podemos encontrar con dos espacios de memoria donde almacenar valores:

- Constantes: *una vez declaradas mantienen su valor durante toda la ejecución del programa.*
- Variables: *pueden cambiar su valor tantas veces como deseen y en algunos casos incluso de tamaño de almacenamiento.*

En Picky las constantes se definen al inicio del código en el apartado de constantes y las variables globales al inicio del programa main.

Vamos a definir dos espacios de memoria en un programa que calcule el perímetro de un círculo. En uno almacenaremos el valor de Pi (3.1415926) que no va a cambiar nunca por lo que será una constante y en otro el valor radio que puede cambiar en función del tamaño de la circunferencia por lo que será una variable.

Por lo tanto en el apartado de constantes escribiremos:

$$Pi = 3.1415926;$$

Al inicio del código del programa declararemos la variable radio:

$$radio: int;$$

## 1.2. Tipos de datos

El lenguaje Picky cuenta con cuatro tipos primitivos:

- Enteros: int

- Reales: float
- Caracteres: char
- Valores de verdad: bool

### 1.3. Tipos definidos

Cuando los tipos primitivos se nos quedan cortos a la hora de definir el programa podemos definir enumeraciones dentro del apartado de tipos de datos. Por ejemplo si necesitamos crear un tipo que sea *días de la semana*:

```
/* Dias de la semana */
TipoDiaSem = (Lun, Mar, Mier, Jue, Vie, Sab, Dom);
```

Este apartado se tratará de forma más detallada en próximas prácticas.

### 1.4. Las funciones

Para evitar que el programa sea muy complejo se utilizan las funciones. Las funciones reciben unos parámetros de entrada, ejecuta una secuencia de líneas de código y retorna una salida. En Picky las funciones se definen en el apartado de funciones y procedimientos.

Cada función tiene dos partes, por un lado donde es definida y otra que es donde se produce la llamada. La primera línea de la definición comienza con la palabra *function* seguida del nombre de la función, entre paréntesis los tipos datos de entrada y finalmente tipo de entrada de salida:

```
function nombredefunción(datoentrada: tipo): tipodato de salida
```

Tras esa primera línea se pone la tarea a ejecutar entre llaves. tras la ejecución se colocará *return* seguido del resultado a devolver. A continuación se muestra un ejemplo con la función que calcula la longitud de la circunferencia con un valor *r* de radio que se proporciona en la entrada.

```
/*
 * Funciones y procedimientos.
 */
function longitudcirculo(r: int): float {
return 2.0 * Pi * r;
}
```

La llamada de esta función se realizará en el programa principal:

```
longitudcirculo(radio);
```

## 2. Ejercicios planteados

En este apartado se plantean varios ejercicios, por cada uno de ellos se debe crear una función que dados unos valores se produzca una salida con la solución. Se deben añadir las constantes y las variables que se consideren. Se hará una llamada a la función en el programa principal mostrando un mensaje con el resultado como se indica a continuación. 'Ejecutada la función *nombredelafunción* con el valor de entrada *valordeentrada* con el resultado: *resultado*'

1. Función *longitudcirculo* que nos devuelve el valor de la longitud de la circunferencia dando el radio como valor de entrada.

2. Función *areacirculo* que nos devuelve el valor del área de la circunferencia dando el radio como valor de entrada.
3. Función *areacuadrado* que nos devuelve el valor del área de un cuadrado dando el lado como valor de entrada.
4. Función *areacorona* que calcula el área de una corona circular dando el radio de interno y el externo como valores de entrada.
5. Función *volumencilindro* que calcula el volumen de un cilindro dado su radio y su altura.

### 3. Entrega

En esta práctica nuestro objetivo es que se comprendan los distintos apartados de un programa. Tendremos que crear un programa que resuelva los ejercicios planteados que compile y se ejecute mostrando correctamente el resultado.

La entrega de esta práctica se hará de forma no presencial, dejando todo el código fuente contenido en un fichero `apellido-nombre-P03.p` que habrá que dejar adjunto en la tarea (Moodle).

Fecha de entrega: **28 de septiembre**

Se podrá entregar hasta las 23:59h de este día.