Faculty of Science, Engineering and Technology

Introduction to Programming

Tutorial Task 4.2: Using Records and Enumerations

Overview

Effectively organising your data makes programs much easier to develop. By using records and enumerations you can start to model the entities associated with your programs.

Purpose:	Learn to use records and enumerations to start managing the data in your programs.
Task:	Create a program that reads in a data about an entity, stores it in a record in your program, and then prints this back out to the terminal.
Time:	This task should be completed before the start of week 5.
Resources:	▪ Chapter 7 of the Programming Arcana ▪ Swinburne CodeCasts (YouTube Channel, iTunesU) • Using records to model entities • Using enumerations to list options • Using pointers to refer to values in memory • Pointing to functions and procedures (optional) ▪ Syntax Videos • Pass by Reference (Var Parameters), Pass by Reference (Const Parameters), Pass by Reference (Out Parameters), Type Declarations, Sub Range Types, Enumerated Types, Record Declarations, Pointers

Submission Details

You must submit the following files to Doubtfire:

- Program source code, and screenshot of the program in action.

Make sure that your task has the following in your submission:

- Code must follow the Pascal coding convention used in the unit (layout, and use of case).
- You are using a record and enumeration to store the values.
- The code must compile and the screenshot show it working, and the validations in action.

Instructions

In this program you will need to **declare** a **record** and an **enumeration**, as well as create functions and procedures to read and write these new types you have created.

The following steps will help you implement your chosen program.

1. Start by creating a new program file in Sublime Text.
2. Implement the basic outline of the program with a Main procedure.
3. Use the **Terminal User Input** unit so that you have access to ReadString and ReadInteger.
4. Declare the **enumeration** using the list of Genre options provided (see below).

Tip: Place type declarations at the top of the file. This makes them easy to find, and ensures they are available to all of you functions and procedures.

5. Declare the record. It will have a number of fields, each of which is like a variable declaration within the record.

Hint: If you have created **Read Integer Range**, it may be useful in this program. It is not essential, but can simplify some parts of these programs.

6. Create the function to **read** in a value of the Genre from the user (see below). The easiest option is to show the user a list of options, and then read in an integer corresponding to the list item. You can then use a case statement (or type casting) to convert the integer into the appropriate value from the enumeration.

Hint: For example, you can assign the name of a Artist using:

```
result.name := ReadString('Enter artist''s name: ');
```

7. Note how in the TerminalUserInput fucntions a **while** loop is used. This shows an error message *while* the user has not entered a valid value, and continues to ask them to enter the data until it is valid. In your music player you may need to use a while loop in this way to ensure the user enters valid data.

Hint: Validating any field is similar to the validations in ReadInteger and ReadInteger-Range, so use a similar pattern when you need to.

8. Create the procedure to **print** a record value to the Terminal. This will accept a parameter, and print the values from its fields to the terminal in the format indicated. Remember to use an if statement to check which, if any, of the additional details you need to write.

Tip: You can use **Write** to write each part to the terminal and then end with a **WriteLn**. This will enable you to write the different parts to the one line.

9. Implement Main, which will need to declare a local variable of the record's type and use the functions and procedures you have created to read and print its value.

10. Use the above four declarations to guide you on starting to create your own Music player program (see Pass Task 3.1 for details). It must have the following features:

- An enumeration of music genres with at least 3 options.
- An album record with at least the following fields:
    - Artist name
    - Genre (use your enumeration for this)
    - Folder location (eg: "c:\my_music\Album1\")
    - Year released (eg: 2015)
- (i.e you need to use different data types: string, enumeration and integer).
- A read function to read in the record.
- Use the TerminalUserInput functions as best as possible to ensure you get valid data for fields.
- A print procedure to output a record's data to the terminal.

Remember: In Main you will create a variable of the record's type. The *read* function is called to provide the values you assign to the record's fields. The *print* procedure is called to output the value from the variable.

Submit your completed program to Doubtfire with a screen capture of the terminal with the program's output.