

dayType Project Report, Pseudocode, and Algorithm

URL:

<https://github.com/The-Lone-Saiyan/DayType-Project>

Planning for DayType Class

View as an ADT

Name: DayType

Domain {Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday}

Operations:

- a. Set the day.
- b. Print the day.
- c. Return the day.
- d. Return the next day.
- e. Return the previous day.
- f. Set the day by adding a certain number of days to the current day. For example, if the current day is Monday and we add 4 days, the new day is Friday. Similarly, if today is Tuesday and we add 13 days, the new day is Monday.
- g. Include the appropriate constructors.

Attributes of a DayType object: Name of the day of the week(name) only be one of the values from the restricted domain(above).

Class Invariant – Day must match one of the items in the domain. The days are ordered where Sunday comes first, then Monday, then Tuesday, etc.

Member Functions:

SetDay(string aDay) – Needs Algorithm

- Pre-Condition: The day is supplied and must be one of the designated days.
- Post-Condition: The day is stored as an instance variable if valid, otherwise the day is set to “Sunday”.

Print()

- Post-Condition: Output the day value.

GetDay()

- Post-Condition: Returns the day as a string.

PreviousDay() – Needs Algorithm

- Post-Condition: Returns the day as a string before this day(instance variable).

NextDay() – Needs Algorithm

- Post-Condition: Returns the day as a string after this day(instance variable).

AddDays(int NumDays) – Needs Algorithm

- Pre-Condition: Value for days to add is supplied and must be a positive integer.
- Post-Condition: The day NumDays in the future is computed and assigned to the instance variable.

DayType(string = “Sunday”)

Can be used as both a default constructor and constructor with default parameters. Allows for declarations like the following: DayType day; or DayType(“Monday”);

Algorithms:**SetDay(string aDay)**

1. Declare an array to hold the acceptable strings.
2. Search the array with the input as the target.
3. If the target is found, assign that value to the instance variable.
4. Else, assign the default day to the instance variable.

Index = 0

Day = DayofWeek[0]

For each day in the week, with index 0 – 6

 If(DayofWeek[0] == Target)

 Day = Target

What about DayofWeek[7] = {“Sunday”, “Monday”, “Tuesday”, “Wednesday”, “Thursday”, “Friday”, “Saturday”};

Store this array as a static variable.

PreviousDay()

//Find the index for the instance variable.

Day = DayofWeek[0]

For each day in the week, with index 0 – 6

 If(DayofWeek[0] == Day)

Location = Index

Break

NewIndex = (index - 1)

If NewIndex < 0 then NewIndex = 6

Return DayofWeek[NewIndex]

NextDay()

//Find the index for the instance variable.

Day = DayofWeek[0]

For each day in the week, with index 0 - 6

If(DayofWeek[0] == Day)

Location = Index

Break

NewIndex = (index + 1)

If NewIndex > 6 then NewIndex = 0

Return DayofWeek[NewIndex]

AddDays(int NumDays)

Find the index of the current day.

NewIndex = (Index + NumDays) % 7

Sunday (0) What day is in 10 days? $(0 + 10) \% 7 = 3$ (Wednesday)

What day is in 7 days? $(0 + 7) \% 7 = 0$ (Sunday)

DayType
-day: string +DayofWeek[7]: string
+SetDay(string): void +GetDay() const: string +Print() const: void +NextDay(): string +PreviousDay(): string +AddDays(int): void +DayType(int = "Sunday")

1. What did you find most challenging with this program?

Getting the programs to display output correctly.

2. What problems did you encounter and how did you solve them?

NextDay and PreviousDay functions weren't outputting the proper display along with AddDays. Couldn't get AddDays to display the correct string variable instead of the default.

NextDay and PreviousDay were solved by debugging with cout seeing how far the program was making it to the for loop. AddDays is currently not solved.

3. What did you learn from writing this program?

How to make a header file and classes and to create objects.