FUNERS

주니어 임베디드SW 챌린저

-2차 기술 지원 교육-

2019. 07. 27

목차

FUNERS

- 1차 기술지원 내용 복습
- 프로그래밍 응용
  - 배열
  - 배열 값을 디스플레이에 출력하기
  - 배열에 색상 정보 저장하기
  - Class 활용하기
- 문제 해결을 위한 솔루션
  - 최단 경로 찾기
  - 최단 경로대로 주행하기
  - 경로 찾기 테스트 코드 실습
  - 실전 연습 시간

배열

배열

FUNERS

- **배열 – 사물함, 기차, 반, 여러 칸이 있는 그릇**
  - 여러 개의 데이터를 하나의 이름과 번호로 저장 할 수 있는 변수
  - 저장된 데이터의 번호는 0부터 시작한다.
  - 배열을 사용하지 않을 경우:
    - 1000개의 데이터 = 1000개 다른 이름을 가진 변수
  - 배열을 사용 할 경우 :
    - 1000개의 데이터 = 하나의 변수 이름과 + 번호, 반복문을 사용하여 데이터를 검색 할 수 있다.

배열 사용하기

FUNERS

- 선언 :
  - 배열이름 = [ ] (비어있는 배열)
  - 배열이름 = [1,2,3,4,5,6]
- 배열에 데이터 추가 및 제거 :
  - 배열이름.append(데이터)
  - 배열이름.remove(데이터)
- 배열의 데이터 읽기:
  - 배열이름[번호]
- 배열의 길이:
  - len(배열이름)
- 배열의 데이터를 크기순으로:
  - 배열이름.sort()

배열

FUNERS

- 배열의 선언과 사용

```
#선언
array = [10,123,4,5,6,7,2,4,5,13]
#데이터 출력
print(array)
#데이터 추가
array.append(10)
print(array)
#데이터 정렬
array.sort()
print(array)
#4번째 데이터 읽기
print(array[3])
#배열 크기
print(len(array))
```

```
#!/usr/bin/env pybricks-micropython

from pybricks import ev3brick as brick
from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,
                                 InfraredSensor, UltrasonicSensor, GyroSensor)
from pybricks.parameters import (Port, Stop, Direction, Button, Color,
                                 SoundFile, ImageFile, Align)
from pybricks.tools import print, wait, StopWatch
from pybricks.robotics import DriveBase

#배열 선언
array = [10,123,4,5,6,7,2,4,5,13]
#데이터 출력
print(array)
#데이터 추가
array.append(10)
print(array)
#데이터 정렬
array.sort()
print(array)
#4번째 데이터 읽기
print(array[3])
#배열 크기
print(len(array))
```

반복문으로 배열 탐색

FUNERS

• 배열의 선언과 사용

```
array = [10,123,4,5,6,7,2,4,5,13]

even_count = 0
for i in array :
    if i%2 == 0 :
        print(i, " = 짝수")
        even_count += 1

print("짝수의 갯수 = ", even_count)
```

```
#!/usr/bin/env pybricks-micropython

from pybricks import ev3brick as brick
from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,
                                 InfraredSensor, UltrasonicSensor, GyroSensor)
from pybricks.parameters import (Port, Stop, Direction, Button, Color,
                                 SoundFile, ImageFile, Align)
from pybricks.tools import print, wait, StopWatch
from pybricks.robotics import DriveBase

array = [10,123,4,5,6,7,2,4,5,13]

even_count = 0
for i in array :
    if i%2 == 0 :
        print(i, " = 짝수")
        even_count += 1

print("짝수의 갯수 = ", even_count)
```

최대/최소/평균값 구하기

배열

FUNERS

```
array = [10,123,4,5,6,7,2,4,5,13]
min_number = 1000
max_number = 0
avg_number = 0

for i in array :
    #최솟값 찾기
    if min_number > i :
        min_number = i
    #최댓값 찾기
    if max_number < i :
        max_number = i
    #평균 구하기1
    avg_number += i
#평균 구하기2
avg_number = avg_number/len(array)


print("최솟값 : ", min_number)
print("최댓값 : ", max_number)
print("평균 : ", avg_number)
```

- 배열의 최대, 최소, 평균 구하기

```
#!/usr/bin/env pybricks-micropython

from pybricks import ev3brick as brick
from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,
                                 InfraredSensor, UltrasonicSensor, GyroSensor)
from pybricks.parameters import (Port, Stop, Direction, Button, Color,
                                 SoundFile, ImageFile, Align)
from pybricks.tools import print, wait, StopWatch
from pybricks.robotics import DriveBase

array = [10,123,4,5,6,7,2,4,5,13]
min_number = 1000
max_number = 0
avg_number = 0

for i in array :
    #최솟값 찾기
    if min_number > i :
        min_number = i
    #최댓값 찾기
    if max_number < i :
        max_number = i
    #평균 구하기1
    avg_number += i
#평균 구하기2
avg_number = avg_number/len(array)

print("min number : ", min_number)
print("max number : ", max_number)
print("avg number : ", avg_number)
```

```
min number :  2
max number :  123
avg number :  17.9
```

디스플레이에 표시하기

디스플레이에 표시하기

FUNERS

- 디스플레이 출력:
  - brick.display.text("text",(x,y))
    - ㅡ. X,Y를 입력하지 않으면 첫 번째 행부터 순서대로 텍스트를 출력합니다.
- 디스플레이 지우기:
  - brick.display.clear()

디스플레이에 표시하기

FUNERS

- **2차 배열 – 아파트, 책장, 체스판**
  - 배열이 들어있는 배열
  - 2개의 번호가 필요하다
    - 바깥쪽 배열에 저장된 배열들의 번호
    - 안쪽 배열에 저장된 데이터들의 번호
  - 선언 :
    - 배열이름 = [[1,2,3],[4,5,6],[7,8,9]]

디스플레이에 표시하기

FUNERS

- 2차 배열 표시하기

```
array = [[0,1,2,3,4,5,6],
[7,8,9,10,11,12],
[13,14,15,16,17,18]]
brick.display.clear()

for index1 in range(0,len(array)) :
    string = ""
    for index2 index range(0,len(array[index1])) :
        string += str(array[index1][index2])
        string += " "
    brick.display.text(string)

wait(5000)
```

```
#!/usr/bin/env pybricks-micropython

from pybricks import ev3brick as brick
from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,
                                 InfraredSensor, UltrasonicSensor
from pybricks.parameters import (Port, Stop, Direction, Button, C
                                 SoundFile, ImageFile, Align)
from pybricks.tools import print, wait, StopWatch
from pybricks.robotics import DriveBase

array = [[0,1,2,3,4,5,6],
         [7,8,9,10,11,12],
         [13,14,15,16,17,18]]
brick.display.clear()

for index1 in range(0,len(array)) :
    string = ""
    for index2 in range(0,len(array[index1])) :
        string += str(array[index1][index2])
        string += " "
    brick.display.text(string)

wait(5000)
```

```
0 1 2 3 4 5 6
7 8 9 10 11 12
13 14 15 16 17 18
```

색상 정보 저장하기

색상 정보 저장하기

FUNERS

- **배열을 사용하여 저장**
  - 배열을 사용하여 색상 정보 저장
- **교차로 개수를 구한 뒤 읽은 순서대로 저장**
  - array = [ ] #교차로 정보를 저장할 배열 선언
  - column = 3 , row = 3 (가로 세로 교차로 개수)
  - total = row*column (전체 교차로 개수)

색상 정보 저장하기 – 색상 정보 저장

FUNERS

```
#로봇 모터 및 센서 선언
left_motor = Motor(Port.B)
right_motor = Motor(Port.C)
color_sensor = ColorSensor(Port.S3)

column = 3 #교차로의 가로 갯수
row = 3 #교차로의 세로 갯수
total = row*column #전체 교차로 갯수

색상 정보 저장
array = [] #각 교차로 정체시간 리스트
for index in range(0, total):
    left_motor.run_angle(35,36,Stop.BRAKE,False)
    right_motor.run_angle(35,36,Stop.BRAKE,True)
    if color_sensor.color() == 1: #녹색을 감지
        array.append(1)
    elif color_sensor.color() == 4: #노란색을 감지
        array.append(2)
    elif color_sensor.color() == 5: #빨간색을 감지
        array.append(3)
    else:
        array.append(8) #다른 색상을 감지

    left_motor.stop(Stop.BRAKE)
    right_motor.stop(Stop.BRAKE)
    brick.sound.beep()
```

```
8  from pybricks.tools import print, wait, StopWatch
9  from pybricks.robotics import DriveBase
10
11 #로봇 모터 및 센서 선언
12 left_motor = Motor(Port.B)
13 right_motor = Motor(Port.C)
14 color_sensor = ColorSensor(Port.S3)
15
16 column = 3 #교차로의 가로 갯수
17 row = 3 # 교차로의 세로 갯수
18 total = row*column #전체 교차로 갯수
19
20 ########################색상 정보 저장
21 array = [] #각 교차로 정체시간 리스트
22 for index in range(0, total):
23     left_motor.run_angle(35,36,Stop.BRAKE,False)
24     right_motor.run_angle(35,36,Stop.BRAKE,True)
25
26     if color_sensor.color() == 1:#녹색을 감지
27         array.append(1)
28     elif color_sensor.color() == 4:#노란색을 감지
29         array.append(2)
30     elif color_sensor.color() == 5:#빨간색을 감지
31         array.append(3)
32     else:
33         array.append(8) #다른 색상을 감지
34
35     left_motor.stop(Stop.BRAKE)
36     right_motor.stop(Stop.BRAKE)
37     brick.sound.beep()
```

색상 정보 저장하기 – 색상 정보 출력

FUNERS

- 색상 정보 출력

```
z=1
# 읽은 색상표시 브릭 LCD 표시 (G=원할,Y=서행,R=정체)
for i in array:
    if array[i] == 1:
        brick.display.text("G",(z*8,7))
    elif array[i] == 2:
        brick.display.text("Y",(z*8,7))
    elif array[i] == 3:
        brick.display.text("R",(z*8,7))
    elif array[i] == 8:
        brick.display.text("F",(z*8,7))
    z=z+1

wait(10000) #10초간 대기
```

```
39  z=1
40  # 읽은 색상표시 브릭 LCD 표시 (G=원할,Y=서행,R=정체)
41  for i in array:
42      if array[i] == 1:
43          brick.display.text("G",(z*8,7))
44      elif array[i] == 2:
45          brick.display.text("Y",(z*8,7))
46      elif array[i] == 3:
47          brick.display.text("R",(z*8,7))
48      elif array[i] == 8:
49          brick.display.text("F",(z*8,7))
50      z=z+1
51
52  wait(10000) #10초간 대기
```

클래스 활용하기

클래스

FUNERS

- **클래스(class) – 설계도, 청사진, 형틀**
  - 객체지향 언어에서의 핵심 요소 중 하나
  - 객체란 세계를 구성하는 요소들의 정의
  - 프로그램의 런(Run)타임에서 객체(object)를 만들기 위한 설계도의 역할
  - 변수, 메소드, 클래스 등을 포함한다.
- **좋은 점?**
  - 필요한 기능을 가진 최적화된 객체를 만들고 사용 할 수 있다.
  - 논리적으로 직관적인 프로그램을 짤 수 있다.

클래스

FUNERS

- **클래스(class) 사용하기**
  - 클래스의 정의 – class 클래스이름:
  - 클래스의 생성자 – def __init__(self, 파라미터들):
  - 클래스의 메소드 – def 메소드이름(self, 파라미터들):

클래스

FUNERS

- 클래스 배열의 저장과 표시

```
from random import randint

class StoreColor: #클래스 정의
    def __init__(self, row, column, color):
        self.row = row
        self.column = column
        self.color = color
```

```
def printColorArray(colorArray): #클래스배열의 정보 표시하기
    for row in colorArray:
        string = ""
        for storeColor in row:
            string += "[" + str((storeColor.row, storeColor.column))
            string += ", " + str(storeColor.color) + "]"

        print(string)


colorArray = []

for row in range(0, 3): #무작위 클래스 배열 생성
    colorArray.append([])
    for column in range(0, 3):
        Color = randint(1,3)
        colorArray[row].append(StoreColor(row, column,Color))

printColorArray(colorArray)
```

최단 경로 찾기

최단 경로 찾기

FUNERS

- 최단 경로의 정의
  - 시작은 좌측 최상단, 도착은 우측 최하단
  - 출발 구역과 1번 회전 교차로 사이에 위치한 9개의 색상 정보를 읽어 도로에 있는 교차로의 상황을 파악한 뒤 최적의 경로를 예측 후 주행

색상	초록	노랑	빨강
해당 번호 교차로의 교통 상황	원활	서행	정체
해당 번호 교차로를 1회전하는 데 걸리는 시간	4분	8분	12분

최단 경로 찾기

FUNERS

- **최단 경로를 찾는 과정**

1. 교차로의 교통 상황들을 모두 저장한다.
2. 교차로 간 교통 상황을 비교한다.
3. 시작점을 기준으로 가장 가까운 지점을 계산하여 표시한다.
4. 주행 중 같은 교차로 내에서의 이동 할 때와 다음 교차로로 이동할 때를 고려하여 주행하여야 한다.

최단 경로 주행

FUNERS

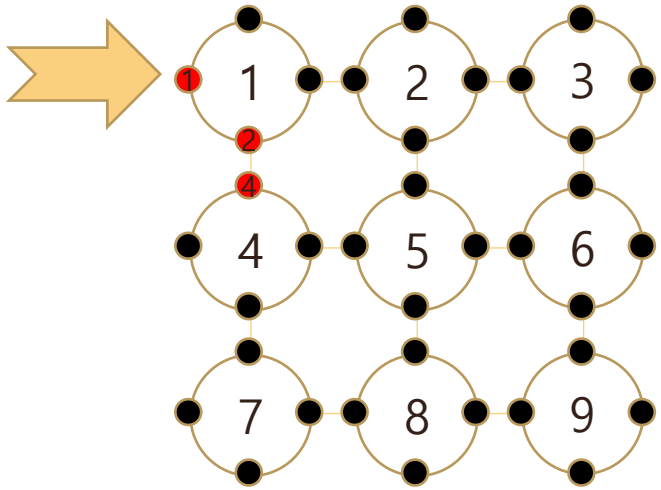
- **로봇의 이동은 회색 선을 따라 이동하여야 한다.**
  - 출발 구역에서 출발한 HW는 도착 구역에 도착할때까지 반드시 회색 선을 따라 이동해야 함.

- **회전 교차로에서는 반 시계 방향으로만 이동할 수 있다.**

최단 경로 찾기

FUNERS

- **현재 교차로에서 다음 교차로로 이동하기**
  - 현재 위치가 1일 때 4로 이동 하려면?

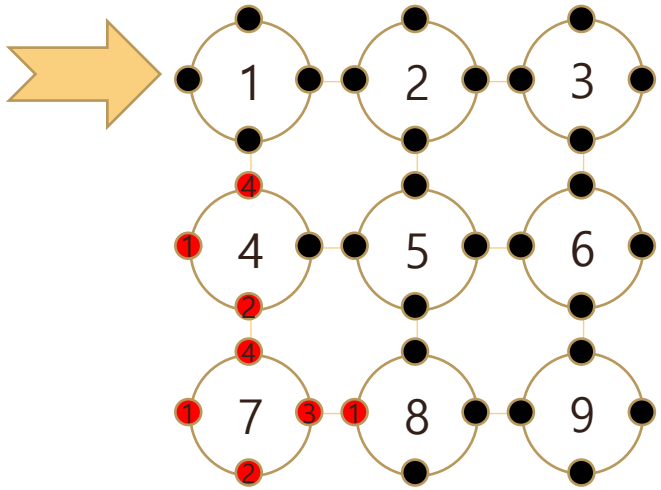


1. 1번 교차로 1번 지점에서 1번 교차로의 2번 지점으로 이동
2. 1번 교차로의 2번 지점에서 4번 교차로의 4번 지점으로 이동

최단 경로 찾기

FUNERS

- **현재 교차로에서 다음 교차로로 이동하기**
  - 현재 위치가 4일 때 8로 이동 하려면?



1. 4번 교차로 4번 지점에서 4번 교차로의 1번 지점으로 이동
2. 4번 교차로의 1번 지점에서 4번 교차로의 2번 지점으로 이동
3. 4번 교차로의 2번 지점에서 7번 교차로의 4번 지점으로 이동
4. 7번 교차로의 4번 지점에서 7번 교차로의 1번 지점으로 이동
5. 7번 교차로의 1번 지점에서 7번 교차로의 2번 지점으로 이동
6. 7번 교차로의 3번 지점에서 8번 교차로의 1번 지점으로 이동

실전 연습 시간

마무리

FUNERS

- 참고 사이트
  - EV3 Micro Python 학습 및 API : https://education.lego.com/en-us/support/mindstorms-ev3/python-for-ev3
  - Visual studio code 설치 관련: https://youtu.be/cqtRqsI6xMc
  - EV3와 PC의 연결 관련 : https://youtu.be/TNXqizQTZhs

- 교육, 기술 지원 및 경기장 구매 문의

  퓨너스 (www.funers.com) T.070-8670-8911