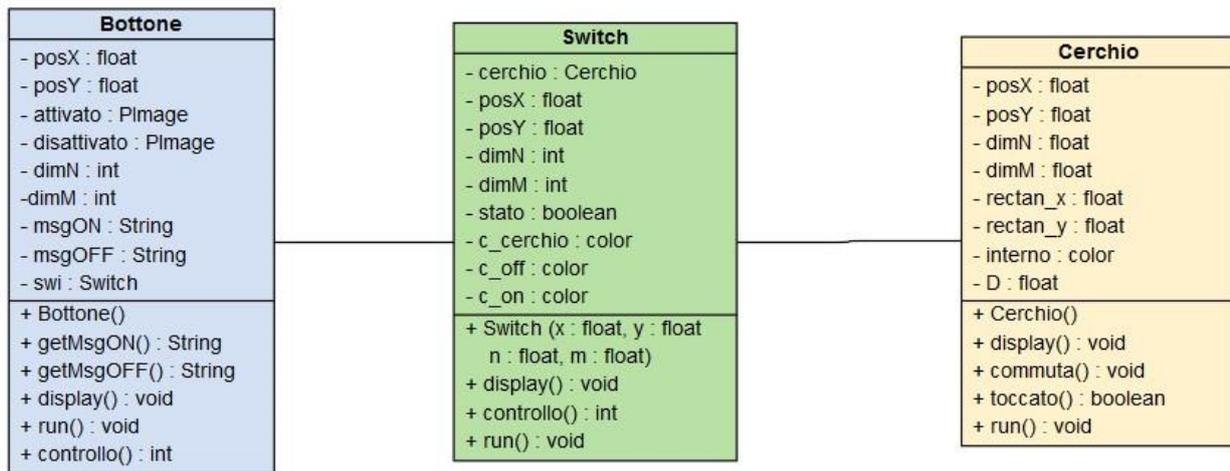


# NOTE DI RILASCIO

## MyHostica versione 2.0

### Architettura del software:



Il programma è composto dalle seguenti classi.

#### •CLASSE CERCHIO:

ATTRIBUTI → Abbiamo: posX, posY, dimN, dimM, interno (colore del cerchio), D(diametro), rectan\_x e rectan\_y(coordinate del rettangolo). posX e posY sono rispettivamente le coordinate del centro della circonferenza, dimN e dimM sono le dimensioni del rettangolo dove la circonferenza sarà appoggiata.

METODI → La classe contiene un metodo per visualizzare la circonferenza, chiamato display. Avremo poi un metodo chiamato commuta che serve per spostare la posizione del cerchio quando lo stato dello Switch cambia. Un altro metodo chiamato toccato, serve per verificare se il puntatore del mouse si trova nei limiti descritti ed infine vi è una funzione chiamata run per eseguire essenzialmente il display ed in caso altri metodi.

#### •CLASSE SWITCH:

ATTRIBUTI → Essi sono formati da: posX, pos, dimN, dimM e da uno stato. posX e posY sono le coordinate del centro della figura complessiva, lo stato indica se il pulsante è attivo o meno ed infine dimN e dimM sono rispettivamente le dimensioni della larghezza e della lunghezza della figura. Abbiamo inoltre altri attributi: delle variabili di tipo color per mantenere il colore del cerchio (c\_cerchio), il colore del bottone da spento (c\_off) e da accesso (c\_on). Inoltre questa classe interagisce con classe Cerchio quindi avremo un istanza di Cerchio chiamata cerchio.

METODI → La funzione display serve per visualizzare un rettangolo (con le altezze arrotondate) di dimensioni proporzionali alla risoluzione del pc (moltiplicata per delle costanti scelte opportunamente). Il rettangolo da visualizzare inoltre fa uso della classe cerchio disegnando così una circonferenza all' estremità (con diametro di poco superiore all' altezza del rettangolo) che deve essere di colore bianco con bordi neri. L' interno del rettangolo può essere di colore rosso o verde e può contenere la scritta "ON" o "OFF" (in base allo stato). Quando viene cliccato il cerchio lo switch cambia di stato, lo stesso succede se viene cliccato il rettangolo. Al cambiamento di stato deve corrispondere lo spostamento del cerchio nell' estremità opposta e la scritta all' interno del rettangolo deve passare da ON ad OFF e viceversa. Alla parola ON è assegnato il colore verde mentre alla parola OFF quello rosso.

#### •CLASSE BOTTONE:

ATTRIBUTI → È una classe con all' interno un oggetto Switch chiamato swi. Lo scopo di questa classe è quello di andare ad inserire un'opportuna immagine sovrainposta allo Switch così da completare il bottone, questa immagine sarà di tipo PImage e corrisponde alla variabile attivato/disattivato (in base allo stato dello switch). La classe è dotata di posX e posY che indicano il vertice dell'immagine. dimN e dimM indicano la larghezza e l'altezza dell'immagine. Infine abbiamo gli ultimi due attributi: msgOn e msgOff, sono due stringhe contenenti il messaggio da mandare alla scheda elettronica ATmega.

METODI → I metodi getMsgON() e getMsgOFF() ritornano rispettivamente le variabili msgON e msgOFF. La funzione display invece serve per visualizzare a schermo (di poco in alto rispetto allo switch) l'immagine del bottone.

Lo scopo dell'oggetto di tipo Switch invece è quello di far cambiare immagine (da disattivato ad attivato) ogni qual volta che cambia lo stato dello switch, questa azione avviene tramite l'apposito metodo controllo() che ritorna un valore intero per segnalare l'avvenuto cambiamento.

#### •CLASSE MYHOSTICA:

Infine abbiamo la classe MyHostica cioè il main che crea e dispone le immagini.

#### Programma per Android creato con Android Studio:

Il programma è composto essenzialmente da una sola classe, ovvero il main. Esso svolge tutte le funzioni (un'ottima miglioria sarebbe quella di suddividere i compiti creando più classi). La funzione principale è quella di creare un collegamento tra lo

smartphone e la scheda Arduino. L'interfaccia grafica è realizzata tramite file XML. Per ulteriori spiegazioni o chiarimenti contattare Giorgio Guardo tramite e-mail.

#### Programma per Windows creato con Arduino:

Nel sorgente sono presenti svariate funzioni, ogni funzione svolge un particolare compito: ad esempio accendere o spegnere la luce, attivare o disattivare una ventola etc. etc.

Ogni funzione ha un nome esplicativo ragion per cui non occorre altra descrizione. Inoltre, sono presenti dei commenti per dividere le zone in base al ruolo del codice.

#### **Collaudi:**

Il software per Windows è stato collaudato tramite un personal computer con SO Windows 10, aventi risorse: intel i5 6° generazione, 8GB RAM DDR3.

Il software per Android è stato collaudato tramite un tablet Samsung Galaxy Tab 3 con versione Android 4.2.2.

#### **Idee per ulteriori sviluppi:**

Una buona idea per migliorare l'app per Android sarebbe quella di implementare una ricerca Bluetooth con la quale scegliere la scheda manualmente (quindi con la possibilità di scegliere in corso d'opera tra una o più schede). Inoltre, bisognerebbe migliorare l'app stessa facendo uso di classi differenti, diversificando i compiti.