Performance Task: Create - Applications from Ideas Program Code

Entire Project

Computational Artifact

AP Computer Science Principles Explore Performance Task

AP Computer Science Principles Create Performance Task Written Responses

AP-Computer-Science-Principles-Performance-Tasks-Repository

3. **Program Code**

Let the blue ovals represent the segment of program code that implements the algorithm I created for my program that integrates other algorithms and integrates mathematical and/or logical concepts.

Let the yellow rectangles represent the segment of program code that represents an abstraction I developed.

```
#include <Wire.h> //built-in to the Arduino IDE
#include <LiquidCrystal.h> //built-in to the Arduino IDE
#include "pitches.h" //acquired from https://gist.github.com/mikeputnam/2820675
#define BLUE 2 //PWM slot 2 - blue channel 🔵
#define GREEN 3 //PWM slot 3 - green channel 🔵
#define RED 4 //PWM slot 4 - red channel 🔵
//5 is occupied by passive buzzer
LiquidCrystal lcd(8, 9, 10, 11, 12, 13); //LCD pins

int i; //for control purposes

int tempPin = 0; 🔵

int melody[] = {
  NOTE_C7, NOTE_C7, NOTE_C7, NOTE_C7, NOTE_C7, NOTE_C7, NOTE_C7, NOTE_C7
🔵 🟨
};

int duration = 500; //constant integer for the "music" 🔵

void setup() {
  Serial.begin(9600);
  pinMode(RED, OUTPUT); 🔵 🟨
  pinMode(GREEN, OUTPUT); 🔵 🟨
  pinMode(BLUE, OUTPUT);
  digitalWrite(RED, HIGH);
  digitalWrite(GREEN, LOW);
  digitalWrite(BLUE, LOW);
  lcd.begin(16, 2); //max number for lcd displays
  lcd.print("Setting up Miner");
  delay(500);
  Serial.println("Initialize DS3231");
  //The below print statements are just for fun! Have fun translating them!
  Serial.println("Настройка шахтёра.");
  delay(500);
  Serial.println("Скачивание большего количества оперативной памяти из полностью открытого исходного кода, 100-процентного бесплатного веб-сайта memes 'downloadmoreram.com'!");
  delay(500);
  Serial.println("RAM успешно запущена, я имею в виду скачанный.");
  delay(500);
  Serial.println("У вас есть оперативная память!");
  delay(500);
  Serial.println("Congratulations!");
  delay(500);
  Serial.println("Configuring miner.");
  delay(500);
  Serial.println("Downloading more VRAM from 'downloadmorevram.com'!");
  delay(500);
  Serial.println("Successful.");
  delay(500);
  Serial.println("You have downloaded 32 quadrillion petabytes of VRAM!");
  Serial.println("Look at that, you can finally get 60 frames per second on Fortnite!");
  delay(500);
  Serial.println("Доступ к Биткойнному кошельку!");
  delay(500);
  Serial.println("Перевод в СССР!");
  delay(500);
  Serial.println("Все ваши биткойны ушли!");
  delay(500);
  Serial.println("Но вы не знаете, что, если вы не перевели эту программу!");
  delay(500);
  Serial.println("Accessing personal information.");
  delay(100);
  Serial.println("Accessing cryptocurrency wallet.");
  delay(500);
  Serial.println("Transfer initiated.");
  delay(500);
}

int redValue;
int greenValue;
int blueValue;

void loop() {
  int PHrate = (random(100, 999));
#define delayTime 5
  redValue = 255;
  greenValue = 0;
  blueValue = 0;
  lcd.setCursor(0, 1);
  lcd.print("Elapsed: ");
  lcd.println(millis() / 1000);
  lcd.setCursor(0, 0);
  lcd.print(PHrate);
  lcd.print(" PH/s        ");

  int tempReading = analogRead(tempPin);
  double tempK = log(10000.0 * ((1024.0 / tempReading - 1)));
  tempK = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * tempK * tempK )) * tempK );
  float tempC = tempK - 273.15;
  float tempF = (tempC * 9.0) / 5.0 + 32.0; //acquired from https://ideone.com/fork/tS9IX1

  Serial.print("Mining rig temperature in Celsius is ");
  Serial.println(tempC);
  Serial.print("Mining rig temperature in Fahrenheit is ");
  Serial.println(tempF);

  Serial.print(PHrate);
  Serial.println(" PH/s");
  Serial.print("Time elapsed in seconds is ");
  Serial.println(millis() / 1000);
  Serial.println();

  for (int thisNote = 0; thisNote < 8; thisNote++) {
    tone(5, melody[thisNote], duration);
    delay(1000);
  }

  for (int i = 0; i < 255; i += 1) {
    redValue -= 1;
    greenValue += 1;
    analogWrite(RED, redValue);
    analogWrite(GREEN, greenValue);
    delay(delayTime);
  }

  redValue = 0;
  greenValue = 255;
  blueValue = 0;

  for (int i = 0; i < 255; i += 1) {
    greenValue -= 1;
    blueValue += 1;
    analogWrite(GREEN, greenValue);
    analogWrite(BLUE, blueValue);
    delay(delayTime);
  }

  redValue = 0;
  greenValue = 0;
  blueValue = 255;

  for (int i = 0; i < 255; i += 1) {
    blueValue -= 1;
    redValue += 1;
    analogWrite(BLUE, blueValue);
    analogWrite(RED, redValue);
    delay(delayTime);
  }
}
```