

29. Desenvolva um tipo de dado abstrato que represente um cilindro. Inclua as funções de inicializações necessárias e as operações que retornem sua altura, seu raio, a área da base, a área lateral, a área total e o volume do cilindro.

Resposta codificada em C, disponível em <https://onlinegdb.com/yBuU-M1MC>

```
/**
 * Arquivo Cylinder.h
 */

/**
 * Encapsulamento dos atributos do cilindro
 */
typedef struct cylinder Cylinder;

/**
 * Construtor do cilindro
 *
 * @param height a altura do cilindro
 * @param radius o raio do cilindro
 * @return a instância do cilindro
 */
Cylinder* create (double, double);

/**
 * Destrutor do cilindro
 *
 * @param instance a instância do cilindro
 */
void destroy (Cylinder*);

/**
 * Retornar a altura do cilindro
 *
 * @param instance a instância do cilindro
 * @return a altura do cilindro
 */
double getHeight (Cylinder*);

/**
 * Retornar o raio do cilindro
 *
 * @param instance a instância do cilindro
 * @return o raio do cilindro
 */
double getRadius (Cylinder*);

/**
 * Retornar a área da base do cilindro
 *
 * @param instance a instância do cilindro
 * @return a área da base do cilindro
 */
double baseArea (Cylinder*);
```

```
/**
 * Retornar a área lateral do cilindro
 *
 * @param instance a instância do cilindro
 * @return a área lateral do cilindro
 */
double lateralArea (Cylinder*);

/**
 * Retornar a área total do cilindro
 *
 * @param instance a instância do cilindro
 * @return a área total do cilindro
 */
double surfaceArea (Cylinder*);

/**
 * Retornar o volume do cilindro
 *
 * @param instance a instância do cilindro
 * @return o volume do cilindro
 */
double volume (Cylinder*);
```

```
/**
 * Arquivo Cylinder.c
 */

#include <math.h>
#include <stdlib.h>
#include "Cylinder.h"

/**
 * Encapsulamento dos atributos do cilindro
 */
struct cylinder
{
    /**
     * A altura do cilindro
     */
    double height;

    /**
     * O raio do cilindro
     */
    double radius;
};

/**
 * Construtor do cilindro
 *
 * @param height a altura do cilindro
 * @param radius o raio do cilindro
 * @return a instância do cilindro
 */
Cylinder* create (double height, double radius)
{
    Cylinder* instance = (Cylinder*) malloc(sizeof(Cylinder));

    instance->height = height;

    instance->radius = radius;

    return instance;
}

/**
 * Destrutor do cilindro
 *
 * @param instance a instância do cilindro
 */
void destroy (Cylinder* instance)
{
    free(instance);
}
```

```
/**
 * Retornar a altura do cilindro
 *
 * @param instance a instância do cilindro
 * @return a altura do cilindro
 */
double getHeight (Cylinder* instance)
{
    return instance->height;
}

/**
 * Retornar o raio do cilindro
 *
 * @param instance a instância do cilindro
 * @return o raio do cilindro
 */
double getRadius (Cylinder* instance)
{
    return instance->radius;
}

/**
 * Retornar a área da base do cilindro
 *
 * @param instance a instância do cilindro
 * @return a área da base do cilindro
 */
double baseArea (Cylinder* instance)
{
    return M_PI * pow(instance->radius, 2);
}

/**
 * Retornar a área lateral do cilindro
 *
 * @param instance a instância do cilindro
 * @return a área lateral do cilindro
 */
double lateralArea (Cylinder* instance)
{
    return 2.0 * M_PI * instance->height * instance->radius;
}

/**
 * Retornar a área total do cilindro
 *
 * @param instance a instância do cilindro
 * @return a área total do cilindro
 */
double surfaceArea (Cylinder* instance)
{
    return 2.0 * M_PI * (instance->height + instance->radius) * instance->radius;
}
```

```
/**
 * Retornar o volume do cilindro
 *
 * @param instance a instância do cilindro
 * @return o volume do cilindro
 */
double volume (Cylinder* instance)
{
    return M_PI * instance->height * pow(instance->radius, 2);
}
```

```
/**
 * Arquivo main.c
 */

#include <stdio.h>
#include "Cylinder.h"

int main()
{
    Cylinder* instance = create(6, 4);

    printf("Altura do cilindro: %f\n", getHeight(instance));

    printf("Raio do cilindro: %f\n", getRadius(instance));

    printf("Área da base do cilindro: %f\n", baseArea(instance));

    printf("Área lateral do cilindro: %f\n", lateralArea(instance));

    printf("Área total do cilindro: %f\n", surfaceArea(instance));

    printf("Volume do cilindro: %f\n", volume(instance));

    destroy(instance);

    return 0;
}
```