

## Lab – Genéricos com Void \* - Desflorestamento

**Objetivo:** Desenvolver TADs utilizando tipos opacos, isto é, com a implementação da estrutura apenas no arquivo código .c. Para este exercício, iremos utilizar o EA Desflorestamento como base. A sua missão é recodificar para que todos os TADs sejam opacos. Neste exercício, também será necessário alocar e liberar dinamicamente uma lista de terrenos.

**ATENÇÃO:** Para a devida entrega desse exercício, pedimos que você disponibilize o link do seu código do replit, no ava. (Caso queiram correções com comentários em seus códigos do replit adicionem o seguinte email como convidado: [alessandro.baldi@edu.ufes.br](mailto:alessandro.baldi@edu.ufes.br))

O Projeto de Monitoramento do Desflorestamento na Amazônia Legal (Prodes/Inpe) fiscaliza o desmatamento da floresta amazônica por meio de satélites e fiscais do Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis (IBAMA). A fiscalização identifica o responsável pelo desmatamento e aplica multas em relação ao número de hectares desmatados.

Implemente um programa que calcule a média e desvio padrão de desmatamento nos terrenos com base nos hectares. Existem 3 tipos de terrenos que podem ser calculados: terrenos retangulares, circulares e triangulares.

Crie e/ou altere os TADs que armazenem as informações de cada tipo de terreno, considerando que agora todos os TADs devem ser opacos.

**Círculos:** os terrenos em forma circular devem armazenar o raio do terreno. Lembre-se que a área do círculo =  $PI * raio^2$  e que 1 hectare = 10000 metros quadrados.

**Retangulares:** terrenos em forma retangular devem armazenar o comprimento e a largura do terreno.

**Triangulares:** os terrenos em forma triangular, para ser facilitado o tratamento de dados, tem como armazenamento o comprimento e a largura do terreno.

Considere um **TAD Terreno** que possui `void * terreno` e um `char` tipo indicando qual tipo está armazenando naquele Terreno.

```
struct terreno{  
    void * terreno;  
    enum TERRENO tipo;  
};
```

A seguir, segue um exemplo do .h do tipo Terreno:

```
enum TERRENO{
    CIRCULO=0,
    RETANGULO=1,
    TRIANGULO=2,
};

//Contem um vetor para cada tipo de forma, alem de armazenar o espaço
utilizado por cada vetor
typedef struct terreno Terreno;

//Renomeando um ponteiro de Terreno
typedef Terreno * Terreno_pt;

/**
 * Inicializa o struct terreno, atribuindo os parametros aos membros
internos do Terreno_pt
 */
Terreno_pt InicializaTerreno(void * terreno, enum TERRENO tipo);

/**
 * Calcula a área do terreno.
 */
double Area(Terreno_pt t);

/**
 * Libera todas as posições da memória
 */
void liberaTerreno(Terreno_pt t);
```

As informações de todos os terrenos devem ser armazenadas em um struct que contém um vetor de **Terreno**. A quantidade de elementos nesse vetor será definida na primeira linha de entrada.

```
struct terrenos{  
    terreno * terrenos;  
    int qtde;  
};
```

A seguir, segue um exemplo do .h do tipo **Terrenos**:

```
#include "Terreno.h"  
  
typedef struct terrenos Terrenos;  
  
typedef Terrenos * Terrenos_pt;  
  
Terrenos_pt InicializaTerrenos(int qtde);  
  
/**  
 * Adiciona no Terreno uma forma f descrita pelo enum TERRENO  
 */  
void adicionarArea(Terrenos_pt t, Terreno * f);  
  
/**  
 * Calcula a área dos terrenos.  
 */  
double AreaTotal(Terrenos_pt t);  
  
/**  
 * Calcula a média total de desmatamento  
 */  
double Media(Terrenos_pt terrenos);  
  
/**  
 * Calcula o desvio padrão de desmatamento  
 */  
double DesvioPadrao(Terrenos_pt terrenos);  
  
/**  
 * Libera todas as posições da memória  
 */  
void liberaTerrenos(Terrenos_pt t);
```

A saída do programa será a média e o desvio padrão de desmatamento.

A média é calculada como: A soma de todas as áreas dividido pelo número de áreas, isto é,

$$Ma = \sum \frac{area}{n_{areas}}$$

$$DP = \sqrt{\frac{\sum_{i=1}^n (x_i - M_A)^2}{n}}$$

O Desvio padrão é calculado onde  $x_i$  é área desmatada,  $MA$  é a média e  $n$  a quantidade de áreas.

**IMPORTANTE:** Cada TAD deverá ser alocado e liberado da memória corretamente. Verifique seu código utilizando o valgrind para garantir que não há vazamento de memória.

Entradas:

Na primeira linha é dado um número **N** referente ao número de terrenos a serem lidos (**1 < N < 100**). As próximas linhas referem-se ao tipo do terreno (C, R ou T), e seus tipos de dados referentes. (lembre-se de utilizar os conceitos de TADs **opacas**, pois será avaliado neste quesito).

C raio

R comprimento largura

T base altura

Saída:

A saída do programa deve informar a média e desvio padrão do terreno desmatado.

Entrada	Saída
5 C 100 R 150 200 R 500 90 T 1000 80 C 2000	MEDIA DESMATAMENTO: 254.248 DESVIO PADRAO: 501.176