**Actividad 1.3: Clases Conceptos Básicos**

CECS2200 Computer Programming Fundamentals

Nombre: Groupo 6 Sec: 22

**Instrucciones para el estudiante:**

1. Definición de la clase, use la estructura de arreglos en la definición de los atributos.

2. Tabla Descriptiva

3. Diagrama de Clases UML (“Unified Modeling Language”)

4. Separar el archivo del código original en tres archivos dentro del mismo proyecto.

5. Salida de programa con los datos originales del problema.

6. Envíe su solución en formato PDF.

|  |  |
| --- | --- |
| **Variable** | **Descripción** |
| lottery[] | Variable tipo arreglo de longitud definida que almacena números aleatorios. |
| user[] | Variable tipo arreglo de longitud definida que almacena números definidos por el usuario. |
| MAX\_CAPACITY | Variable de tipo entero constante que almacena el valor de la longitud de arreglo. |
| **Función Miembro** | **Descripción** |
| Lottery():Lottery | Constructor por defecto es iniciado cuando no hay parámetros en la definición de objeto de la clase Lottery. |
| Lottery(aSize: const int) : Lottery | Constructor con parámetros iniciales de tipo int. |
| ~Lottery() : Lottery | Función Destructora. |
| setSize(aSize : int) : void | Función que determina el valor de la variable size. |
| setRandomNumbers() : void | Función que determina valores aleatorios para el arreglo de tipo entero lottery. |
| setUserNumbers() : void | Función que determina los valores de la variable de tipo arreglo user, entrados por el usuario. |
| getSize() const : void | Función que retorna valor size. |
| displayRandomArray() : void | Función que imprime contenido del arreglo lottery. |
| displayUserArray() : void | Función que imprime contenido del arreglo user. |
| compareArrays() : void | Función que compara los arreglos de tipo entero lottery y user y determina si los valores son iguales. |
| matchingDigits() : int | Función que busca los valores indenticos dentro de los arreglos lottery y user. |

## Class:

|  |
| --- |
| **Lottery** |
| -lottery[MAX\_CAPACITY: const int] : int  -user[MAX\_CAPACITY: const int] : int  -MAX\_CAPACITY: const int |
| +Lottery() :  +Lottery() :  +~Lottery() :  +setSize(aSize : int) : void  +setRandomNumbers() : void  +setUserNumbers() : void  +getSize() : void  +displayRandomArray() : void  +displayUserArray() : void  +compareArrays() : void  +matchingDigits() : int |

## Lottery Application:

**LotteryApplication.h**

#pragma once

#include <iostream>

using namespace::std;

const int MAX\_CAPACITY = 5;

class Lottery {

private:

int lottery[MAX\_CAPACITY];

int user[MAX\_CAPACITY];

int size;

public:

Lottery();

Lottery(const int aSize);

~Lottery();

void setSize(int aSize);

void setRandomNumbers();

void setUserNumbers();

int getSize() const;

void displayRandomArray() const;

void displayUserArray() const;

void compareArrays();

int matchingDigits();

};

**Main.cpp**

#include <iostream>

#include <stdlib.h>

#include "LotteryApplication.h"

using namespace::std;

int main() {

const int aSize = 5;

Lottery myNumbers(aSize);

myNumbers.displayRandomArray();

myNumbers.displayUserArray();

myNumbers.compareArrays();

cout << endl;

return 0;

}

**LotteryApplication.cpp**

#include <iostream>

#include <cstdlib>

#include "LotteryApplication.h"

using namespace::std;

//Constructor and Destructor

Lottery::Lottery() {

setSize(5);

//set lottery and user to 0

for (int i = 0; i < getSize(); i++) {

lottery[i] = 0;

user[i] = 0;

}//end for

}

Lottery::Lottery(const int aSize) {

setSize(aSize);

setRandomNumbers();

setUserNumbers();

}

Lottery::~Lottery() {

cout << "\nDestructor has been invoked" << endl;

}

// Mutators\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Lottery::setRandomNumbers() {

const int MinValue = 0;

const int MaxValue = 10;

//get the system time and seed the random generator

unsigned seed = time(0);

srand(seed);

for (int i = 0; i < getSize(); i++) {

lottery[i] = (rand() % (MaxValue - MinValue + 1)) + MinValue;

}//end for

}

void Lottery::setUserNumbers() {

cout << "Enter " << getSize() << " digits between 0 and 9." << endl;

for (int i = 0; i < getSize(); i++) {

cout << "Enter digit " << i + 1 << ":\t" << endl;

cin >> user[i];

}//end for

}

void Lottery::setSize(int aSize) {

while (aSize > 5) {

cout << "Invalid size please try again with digits between 1 and 5." << endl;

cin >> aSize;

}

size = aSize;

}

// Accessors\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int Lottery::getSize() const {

return size;

}

//display\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Lottery::displayRandomArray() const {

cout << "The Lottery Numbers are:\n";

for (int i = 0; i < getSize(); i++) {

cout << lottery[i] << " ";

}//end for

cout << endl;

}

void Lottery::displayUserArray() const {

cout << "The user's numbers are:\n";

for (int i = 0; i < getSize(); i++) {

cout << user[i] << " ";

}//end for

cout << endl;

}

//compare arrays\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void Lottery::compareArrays() {

if (matchingDigits() == 5) {

cout << "The number of matching digits is: " << matchingDigits() << endl;

cout << "You are the Grand Prize Winner!!" << endl;

}//end if

else

cout << "\nThe number of matching digits is: " << matchingDigits() << endl;

}

int Lottery::matchingDigits() {

int matchDigits = 0;

for (int i = 0; i < getSize(); i++) {

for (int j = 0; j < getSize(); j++) {

if (user[i] == lottery[j])

matchDigits++;

}//end if

}//end for

return matchDigits;

}

## Salida:

Text

Description automatically generated

Text

Description automatically generated

A computer screen capture

Description automatically generated with medium confidence