

Ayudantía 4

Repaso Solemne

Benjamín Aceituno

Programación Avanzada S09



Pregunta 1

Solemne 2022

Implemente una clase llamada **PreguntaRespuesta** que contenga los atributos `respuestaAlumno`, `respuestaCorrecta`, `puntajePregunta`. Esta clase debe incluir métodos que permitan obtener el puntaje que recibe el alumno según si su respuesta es correcta o no, además de imprimir la información de la pregunta.

Además, implemente una clase **Evaluacion** que contenga un arreglo de preguntas del tipo `PreguntaRespuesta`. La clase debe permitir agregar preguntas y verificar que el puntaje total de las preguntas sumen exactamente 6 puntos, de lo contrario, no calculará la nota obtenida y retornará -1. La nota final del alumno se calcula sumando los puntajes obtenidos en las preguntas correctas y se le añade 1 punto extra.

Solución

```
#include <iostream>
using namespace std;
class PreguntaRespuesta
{
private:
    int respuestaAlumno;
    int respuestaCorrecta;
    float puntajePregunta; //el que otorga si la respuesta es correcta
public:
    PreguntaRespuesta(int respuestaAlumno, int respuestaCorrecta, float puntajePregunta)
    {
        this->respuestaAlumno = respuestaAlumno;
        this->respuestaCorrecta = respuestaCorrecta;
        this->puntajePregunta = puntajePregunta;
    }
    float getPuntajePregunta()
    {
        return puntajePregunta;
    }
    float getPuntaje()
    {
        if (respuestaAlumno == respuestaCorrecta)
        {
            return puntajePregunta;
        }
        else
        {
            return 0;
        }
    }
    void imprimir()
    {
        cout << "Respuesta Alumno: " << respuestaAlumno << endl;
        cout << "Respuesta Correcta: " << respuestaCorrecta << endl;
        cout << "Puntaje: " << getPuntaje() << endl;
    }
};
```

Solución

```
class Evaluacion
{
private:
    private:
        PreguntaRespuesta *lista[6];
public:
    Evaluacion()
    {
        for(int i = 0; i < 6; i++)
        {
            lista[i] = NULL;
        }
    }
    void agregarPregunta(PreguntaRespuesta *resp)
    {
        bool encontro=false;
        for(int i = 0; i < 6; i++)
        {
            if (lista[i] == NULL)
            {
                lista[i] = resp;
                encontro = true;
                cout << "Respuesta asignada"<<endl;
                break;
            }
        }
        if (encontro == false)
        {
            cout << "No se puede agregar respuesta por falta de espacio"<<endl;
        }
    }
}
```

Solución

```
bool validaPuntajes()
{
    float suma = 0;
    for (int i = 0; i < 6; i++)
    {
        if (lista[i] != NULL)
        {
            suma = suma + lista[i]->getPuntajePregunta();
        }
    }
    if (suma == 6)
    {
        return true;
    }
    else
    {
        return false;
    }
}

float notaObtenida()
{
    float suma = 0;
    if (validaPuntajes() == false)
    {
        return -1;
    }
    else
    {
        for (int i = 0; i < 6; i++)
        {
            if (lista[i] != NULL)
            {
                suma = suma + lista[i]->getPuntaje();
            }
        }
        return suma+1;
    }
}

};
```

Solución

```
int main()
{
    PreguntaRespuesta *pr1 = new PreguntaRespuesta(1,2,1);
    PreguntaRespuesta *pr2 = new PreguntaRespuesta(3,3,1.5);
    PreguntaRespuesta *pr3 = new PreguntaRespuesta(1,1,1);
    PreguntaRespuesta *pr4 = new PreguntaRespuesta(2,2,2.5);
    Evaluacion *e = new Evaluacion();
    e->agregarPregunta(pr1);
    pr1->imprimir();
    e->agregarPregunta(pr2);
    pr2->imprimir();
    e->agregarPregunta(pr3);
    pr3->imprimir();
    e->agregarPregunta(pr4);
    pr4->imprimir();
    cout << "Su nota es: " << e->notaObtenida() << endl;
}
```

Pregunta 2

Solemne 2022

Desarrolle una clase llamada **Investigacion** que permita acumular artículos publicados en tres revistas científicas: Nature, Science y Astrophys. La clase debe contener métodos para acumular artículos en una revista específica y para eliminar artículos de una revista en caso de haber cometido un error en el registro. Además, debe implementar un método que imprima las estadísticas de los artículos acumulados, mostrando la cantidad de artículos por revista y el porcentaje de publicaciones en cada una con respecto al total acumulado.

El programa debe simular la acumulación de artículos en estas revistas, imprimir las estadísticas antes y después de registrar los artículos.

Solución

```
#include <iostream>
using namespace std;
class Investigacion
{
private: // INT REVISTAS
    int Nature;
    int Science;
    int Astrophys;
public:
    Investigacion()
    { // REVISTAS INICIALIZADAS EN 0
        Nature = 0;
        Science = 0;
        Astrophys = 0;
    }
    void acumularEnRevista(string nombreRevista) // metodo acumularRevista que recibe una revista y la aumenta en uno
    {
        if (nombreRevista == "Nature")
        {
            Nature++;
        }
        if (nombreRevista == "Science")
        {
            Science++;
        }
        if (nombreRevista == "Astrophys")
        {
            Astrophys++;
        }
    }
    void eliminarArticuloRevista(string nombreRevista) // eliminar revista, recibe una revista y la elimina
    {
        if (nombreRevista == "Nature" && Nature > 0) // tiene que tener por lo menos una revista
        {
            Nature--;
        }
        if (nombreRevista == "Science" && Science > 0)
        {
            Science--;
        }
        if (nombreRevista == "Astrophys" && Astrophys > 0)
        {
            Astrophys--;
        }
    }
}
```


Solución

```
void imprimirEstadisticas()
{
    cout << "ESTADISTICAS INVESTIGACION"<<endl; // Imprime todas las estadisticas
    cout << "Nature " << Nature << " articulos" << endl;
    cout << "Science " << Science << " articulos" << endl;
    cout << "Astrophys " << Astrophys << " articulos" << endl;
    float suma = Nature + Science + Astrophys; // Suma de todas las revistas
    if ( suma > 0)
    {
        cout << "Nature " << Nature/suma *100.0 << " %" << endl; // Porcentaje de nature
        cout << "Science " << Science/suma *100.0 << " %" << endl; // Porcentaje de science
        cout << "Astrophys " << Astrophys/suma *100.0 << " %" << endl; // Porcentaje de astrophys

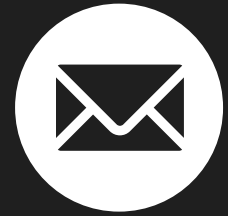
    }
    cout << "-----"<<endl<<endl;
}

int main()
{
    Investigacion *inv = new Investigacion();
    inv->imprimirEstadisticas();
    inv->acumularEnRevista("Nature");
    inv->acumularEnRevista("Nature");
    inv->acumularEnRevista("Science");
    inv->acumularEnRevista("Nature");
    inv->acumularEnRevista("Nature");
    inv->acumularEnRevista("Astrophys");
    inv->acumularEnRevista("Astrophys");
    inv->imprimirEstadisticas();
}
```

Contacto



+569 86031881



benjamin.aceituno@mail.udp.cl



benja.mp4



[https://github.com/benjamp4/
prog.av-2024-2](https://github.com/benjamp4/prog.av-2024-2)



QR para el grupo de Whatsapp

