

4IT Erasmus+Project 2024-1-RO01-KA121-VET-000214264
Beneficiary Colegiul National Gheorghe Vranceanu, Bacau, Romania

Sending organization: Colegiul National ”Ferdinand” Bacau, Romania

MOBILITY PORTFOLIO
Paphos, Cyprus

Mobility period:
PAPHOS, CYPRUS, 13.04.2025-27.04.2025 – practice period 14.04.2025-25.04.2025

Receiving

Practice tutor: Lazaros Lazarou

Participant: Sociu Raul-Gabriel

DISCLAIMER: The European Commission’s support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

My personal website-About me
Sociu Raul-Gabriel



PASIUNI SI ASPIRATI

MUZICA



Pentru mine muzica este o modalitate de a evada din lumea reala si a trai cea ce sunetele, artisti si frecventele transmit.

Pasiunea pentru muzica a inceput de mic dar in ultimi doi ani a devenit un lucru extrem de importnat pentreu mine. Acest lucru sa intamplat in momentul in care am realizat cat de multe mesaje, sentimente si emoti sunt transmise de catre compozitor. Acest lucru ma motivat sa invat sa cant la un instrument si sa transmit si eu, la randul meu, lucrurile pe care cei mai buni artisti le pot capta.

Pe langa acest lucru muzica ma determinat sa invat cum sa imbin melodi in feluri inedite, acest lucru ma determinat sa imi cumpar primul mixer pentru muzica si sa invat sa reproduc lucrurile pe care DJ-i de talie mondiala le fac. Dupa multa perseverenta si timp petrecut exersand am ajuns sa pot transmite oamenilor o energie urisa si sa ii fac sa se distreze pe ritmurile alese de mine.



SCOUTING

Scouting sau in romana cercetasia este mai mult un stil de viata decat orice alt ceva. Acesta nu presupune sa te urci pe munti sau sa vinzi prajituri la casele oamenilor cum multi cred. Acesta presupune sa faci tot posibilul sa fi cea mai buna versiune a ta din orice punct de vedere , lumea poate face asta prin multe modalitati , merg la voluntariat, isi ajuta apropele si multe altele dar cel mai importnat lucru este sa nu ramai in zona de comfort sau sa stagnezi din orice punct de vedere.

Acest lucru ma determinat si inca o face sa invat lucruri noi in fiecare zi, sa cunosc persoane noi, din diferite zone sau clase sociale, si sa imi deschid orizonturile cat de mult posibil. Datorita acestor lucruri am reusit sa devin o persoana echilibrata, deschisa la orice oportunitate noua si gata de orice provocare pe care viata mi-o poate arunca la momentul actual.

VOLUNTARIAT

Daca mai sus am vorbit despre cum este bine sa iesi din zona de comfort, aici chiar am facut-o.

Prin experientele capatate in timpul diferitelor acte de voluntariat am invatat cum sa ma adaptez in diferite situati si sfere sociale. Datorita voluntariatului am reusit sa apar pe televizor si sa fac parte din proiectul educatie la inaltime, proiect sustinut de ministerul invatamantului. Aceste experiente m-au invatat cum sa imi gestionez timpul, cum sa ma descurc cu diferite responsabilitati si cum sa invat sa fac lucrurile din mers.

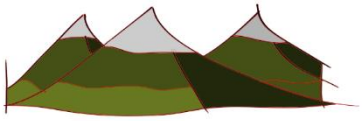
GALERIE DE IMAGINI SI DESENE CREATE IN APLICATIA INKSCAPE

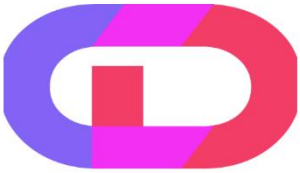
In realizarea urmatoarelor imagini am folosit aplicatia inkscape si toate uneltele de care dispune.

In urmatoarele imagini se poate observa folosirea gradientului, suprapunei,inserarea formelor geometrice, centrarea si altele.









OBIECTIVE

VIITORUL APROPIAT



in viitorul apropiat vreau sa reusesc sa ma implic mai mult in comunitate si sa reusesc sa determin oameni sa devina mai buni, sa realizeze ca este important sa invete lucruri noi constant.

Pentru a reusi asta am inceput sa ma pregatesc si sa ma implic in organizarea mai multor proiecte ca de exemplu:Festivalul lumini 2024, doua proiecte sociale si recrutari pentru Insula cercetasilor si activitati pentru copii din mediul rural. Experienta capata la organizarea acestor activitati ma va ajuta la urmatoarele proiecte.

Am ajutat multe persoane sa realizeze ca au nevoie de o schimbare in viata. Acest lucru a fost facut prin mai multe sesiuni in care am vorbit cu diferiti tineri si am reusit prin exemple concrete sa ii fac sa realizeze ca pt a evolua trebuie sa schimbe ceva in viata lor. Acest lucru a determinat un numar de persoane sa isi schimbe comportamentul si sa se implice in activitati care reusesc sa faca comunitatea locala un loc mai bun.



FACULTATE

Dupa ce termin clasa a 12-a vreau sa intru la o facultate pe profil de informatica pentru a reausi sa fac rost de o diploma ce ma va ajuta pe viitor. Din cauza aceztui fapt, participarea la acest proiect ma va ajuta enorm pe viitor, nu doar pentru intrarea la facultate, ci si pentru gasirea unui loc de serviciu respectat.

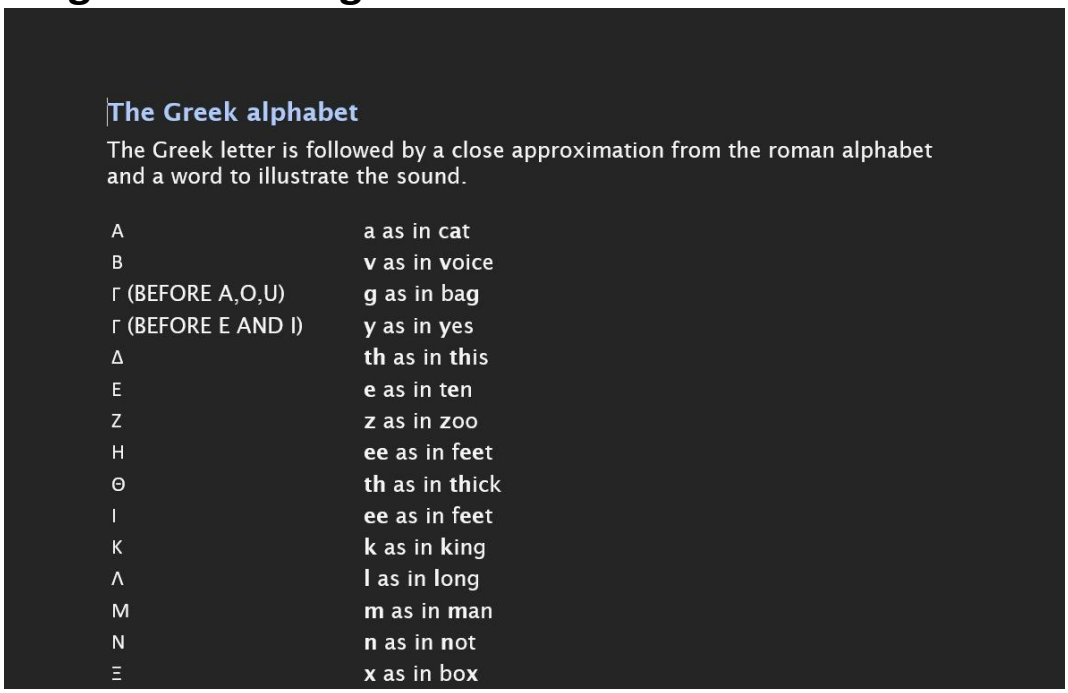
Training before mobility

-cultural training

Cultural diversity and combating discrimination



-linguistic training

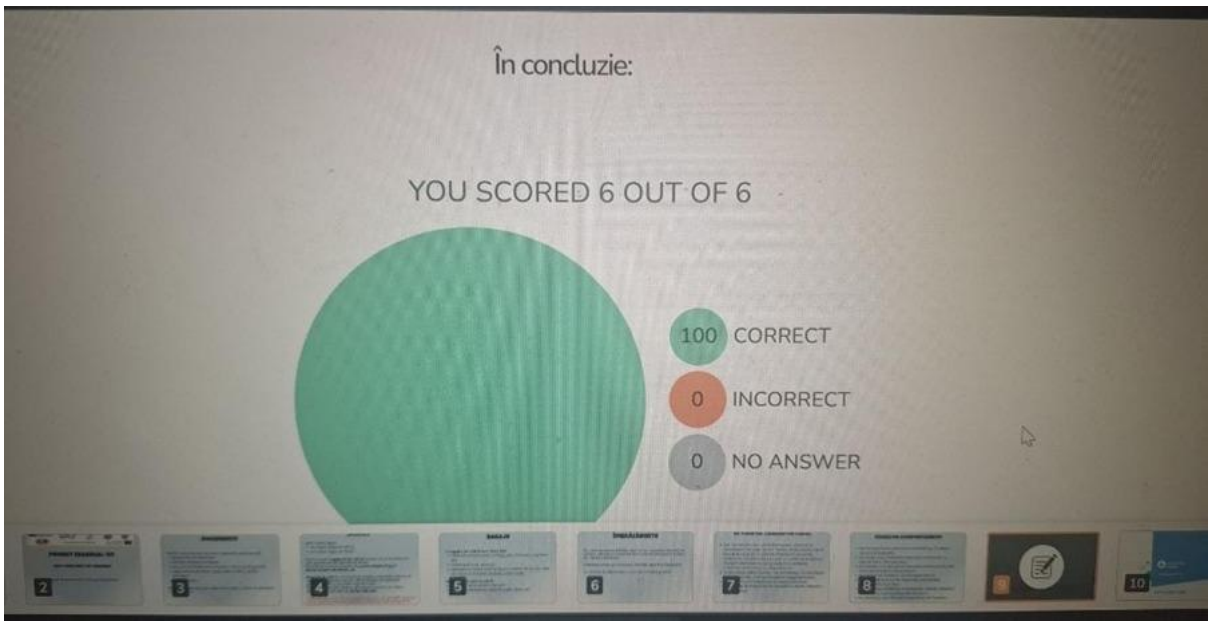


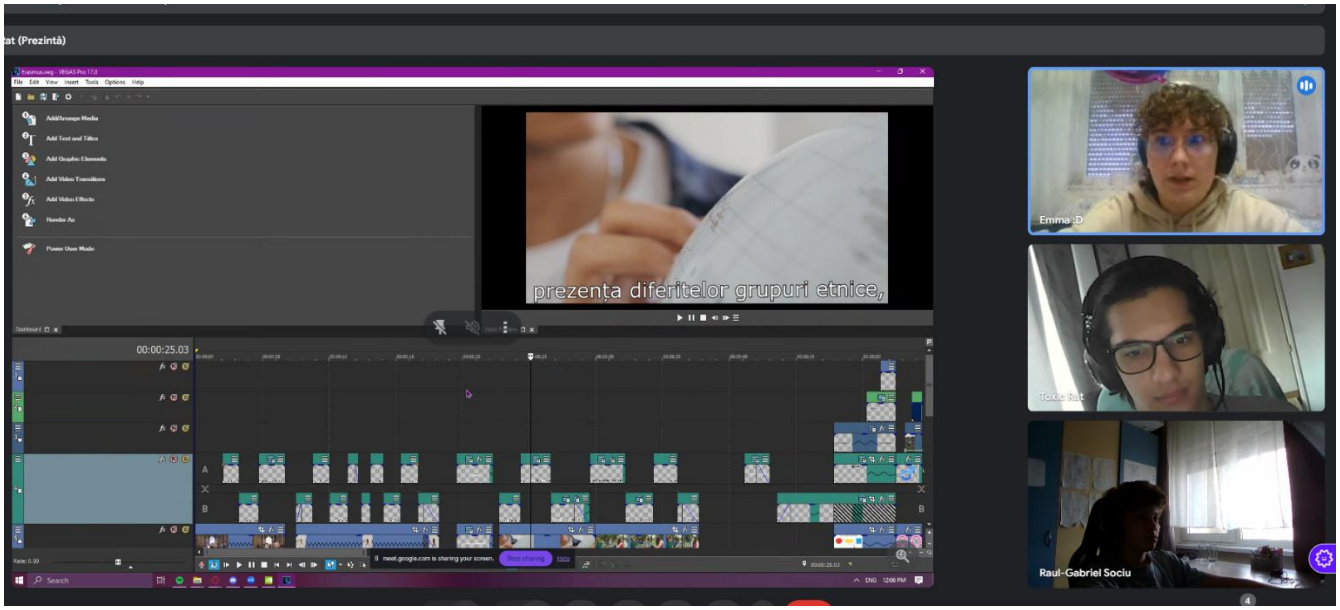
The Greek alphabet

The Greek letter is followed by a close approximation from the roman alphabet and a word to illustrate the sound.

Α	**a** as in **cat**
Β	**v** as in **v**oice
Γ (BEFORE A,O,U)	**g** as in ba**g**
Γ (BEFORE E AND I)	**y** as in **y**es
Δ	**th** as in **th**is
Ε	**e** as in t**e**n
Ζ	**z** as in **z**oo
Η	**ee** as in f**ee**t
Θ	**th** as in **th**ick
Ι	**ee** as in f**ee**t
Κ	**k** as in **k**ing
Λ	**l** as in **l**ong
Μ	**m** as in **m**an
Ν	**n** as in **n**ot
Ξ	**x** as in bo**x**

-pedagogical training





Mobility projects

Car Database

Write a program that uses a class to represent a car. The program should be able to collect information about multiple cars, save them in an array (list) of objects, and offer options to display and search through this data.

Requirements:

1. Define a class (e.g., Car) that contains the following attributes:
    - Car Brand
    - Car Model
    - Plate Number (e.g., "ABC 999")
    - Name of Owner
2. At the beginning of the program execution, an interactive menu should be displayed, allowing the user to choose one of the following options:
    - **Insert data** – Add a new car's information
    - **Display data** – Display all cars saved in the array
    - **Search** – Search based on one of the following criteria:
        - Car Brand
        - Car Model
        - Plate Number
        - Name of Owner
3. The program should allow repeated entry of data for multiple cars.
4. All entered data will be saved in a list of objects of the defined class type.

```
#include <iostream>
#include <string>
#include <algorithm>
using namespace std;

class masina{
   private:
      string brand;
      string model;
      string plate;
      string owner;
   public:
      masina(){//constructor: We use constructors to initialize the object's variables
         brand="EMPTY";
         model="EMPTY";
         plate="EMPTY";
         owner="EMPTY";
      }
   void setBrand(string n){  brand=n; }
   void setModel(string l){ model=l; }
   void setPlate(string e1){ plate=e1;}
   void setOwner(string e2){ owner=e2;}
   string getBrand(){ return brand;}
   string getModel(){ return model;}
   string getPlate(){ return plate;}
   string getOwner(){ return owner;}
};
   masina* mycar;
   int nrMasini = 0;
   string caut;
   int MAXCARS=nrMasini;
   int nextpos=0;
void cauta1()
{
   int i,ok=0;
   for (i=0;i<nextpos;i++)
   {
      if (mycar[i].getBrand()==caut)
         {
      cout<<"car nr "<<i<<endl<<"the car brand is :"<<mycar[i].getBrand()<<endl
         <<" the car model is: "<<mycar[i].getModel()<<endl
         <<"The license plate number is:"<<mycar[i].getPlate()<<endl
         <<"The owners name is: "<<mycar[i].getOwner()<<endl<<endl;
         ok=1;
         }
   }
   if (ok==0) cout<<"Brand not found";
```

```
}
void cauta2()
{
   int i,ok=0;
   for (i=0;i<nextpos;i++)
   {
      if (mycar[i].getModel()==caut)
         {
      cout<<"car nr "<<i<<endl<<"the car brand is :"<<mycar[i].getBrand()<<endl
         <<" the car model is: "<<mycar[i].getModel()<<endl
         <<"The license plate number is:"<<mycar[i].getPlate()<<endl
         <<"The owners name is: "<<mycar[i].getOwner()<<endl<<endl;
         ok=1;
         }
   }
   if (ok==0) cout<<"Model not found";

}
void cauta3()
{
   int i,ok=0;
   for (i=0;i<nextpos;i++)
   {
      if (mycar[i].getOwner()==caut)
         {
      cout<<"car nr "<<i<<endl<<"the car brand is :"<<mycar[i].getBrand()<<endl
         <<" the car model is: "<<mycar[i].getModel()<<endl
         <<"The license plate number is:"<<mycar[i].getPlate()<<endl
         <<"The owners name is: "<<mycar[i].getOwner()<<endl<<endl;
         ok=1;
         }
   }
   if (ok==0) cout<<"Owner not found";

}
void cauta4()
{
   int i,ok=0;
   for (i=0;i<nextpos;i++)
   {
      if (mycar[i].getPlate()==caut)
         {
      cout<<"car nr "<<i<<endl<<"the car brand is :"<<mycar[i].getBrand()<<endl
         <<" the car model is: "<<mycar[i].getModel()<<endl
         <<"The license plate number is:"<<mycar[i].getPlate()<<endl
         <<"The owners name is: "<<mycar[i].getOwner()<<endl<<endl;
         ok=1;
         }
   }
   if (ok==0) cout<<"Plate not found";

}
void afis()
{
   int i;
   for (int i=0;i<nextpos;i++){

      cout<<"car nr "<<i<<endl
         <<"the car brand is :"<<mycar[i].getBrand()<<endl
         <<"the car model is: "<<mycar[i].getModel()<<endl
         <<"The license plate number is:"<<mycar[i].getPlate()<<endl
         <<"The owners name is: "<<mycar[i].getOwner()<<endl<<endl;
   }
}
//input data from the user via console. it has global valuables no imported valuables
void declarare() {
   int nrNoi;
   cout << "How many cars would you like to input? ";
   cin >> nrNoi;

   // Allocate a new array
   masina* newArray = new masina[nextpos + nrNoi];

   // Copy over existing cars
   for (int i = 0; i < nextpos; i++) {
      newArray[i] = mycar[i];
   }
   //new array is a tomporary data storage, it is going to be copied into mycar;
   // Input new cars
   string brand, model, plate, owner, platePart2;
```

```
    for (int i = 0; i < nrNoi; i++) {
        cout << "Brand for car nr " << (nextpos + i) << " is: ";
        cin >> brand;
        newArray[nextpos + i].setBrand(brand);

        cout << "Model " << (nextpos + i) << ": ";
        cin >> model;
        newArray[nextpos + i].setModel(model);

        cout << "Plate number " << (nextpos + i) << ": ";
        cin >> plate >> platePart2;
        plate = plate + ' ' + platePart2;
        newArray[nextpos + i].setPlate(plate);

        cout << "Owner name " << (nextpos + i) << ": ";
        cin >> owner;
        newArray[nextpos + i].setOwner(owner);
    }

    //Free old array memory
    delete[] mycar;

    //Update pointer and nextpos
    mycar = newArray;
    nextpos += nrNoi;
}


int main(){
    int ok=-1;

    do{
        cout << "what action do you want to do?" << endl
             << "type 1 if you want to input car data" << endl
             << "type 2 to show data" << endl
             << "type 3 to search cars based on model" << endl
             << "type 4 to search cars based on brand" << endl
             << "type 5 to search cars based on owner" << endl
             << "type 6 to search cars based on number plate" << endl
             << "type 0 to exit" << endl;
        cin>>ok;
        cout<<endl<<endl;
        //un fel de if dar pt oameni mai destepti
        switch(ok)
        {
        case 1:
            declarare();
            //it inputs the amount and details for the cars
            break;
        case 2:
            afis();
            //it displays the inputed data
            break;
        case 3:
            cout<<"what brand are you looking for?";
            cin>>caut;
            cauta1();
            break;
        case 4:
            cout<<"what model are you looking for?";
            cin>>caut;
            cauta2();
            break;
        case 5:
            cout<<"what owner are you looking for?";
            cin>>caut;
            cauta3();
        break;
        case 6:
            cout<<"what license plate are you looking for?";
            string nrimat1,nrimat2;
            cin>>nrimat1>>nrimat2;
            //caut primeste nr de inmatriculare format din doua stringuri
            caut=nrimat1+' '+nrimat2;
            cauta4();
            break;
        }
        }while(ok);
delete[] mycar;
    return 0;
}
```

Supermarket Program

Write a program for a supermarket that will collect and store information about various products being sold. The program should allow:

- entering product data,
- displaying all saved products,
- searching for products by brand and by type.

Requirements:

1. Define a class (e.g., Product) that contains the following attributes:
    - Product Brand (e.g., "Nestlé", "Coca-Cola")
    - Product Type (e.g., "Beverage", "Dairy", "Canned Goods")
    - Price (for example, 12.50 lei)
    - Available Stock (number of units remaining in stock)
2. Information about each product will be saved in a list (array) of Product objects.
3. When the program starts, an interactive menu must appear, from which the user can choose one of the following options:

Additional Features:

- **Add New Product**: The user will enter values for all 4 fields (brand, type, price, stock).
- **Display Products**: All saved products will be displayed, each with its details.
- **Search Product**:
    - By Brand: The user will input a brand, and the program will display all corresponding products.
    - By Type: The user will input a product type, and the program will display all products of that type.

```
#include <iostream>
using namespace std;


class magazin{
   private:
      string brand;
      string type;
      long long price;
      long long stock;
   public:
      magazin(){//constructor: We use constructors to initialize the object's variables
         brand="EMPTY";
         type="EMPTY";
         price=-1;
         stock=-1;
      }
   void setBrand(string n){
      brand=n; }
   void setType(string l){
      type=l; }
   void setPrice(int e1){
      price = e1; }
   void setstock(int e2){
      stock = e2; }
   string getBrand(){
      return brand;}
   string getType(){
      return type;}
   int getPrice(){
      return price;}
   int getStock(){
      return stock;}

};
magazin* mystore;
int MAXPRODUCTS = 0;
string cauta;

void declarare()
{
   cout << "How many products do you want to enter? ";
   cin >> MAXPRODUCTS;
   mystore = new magazin[MAXPRODUCTS];
   string brand="", type="";
   int price=-1, stock=-1,i;

   for (int i=0;i<MAXPRODUCTS;i++){
      cout<<"Brand for item number "<<i<<"is: ";
```

```
      cin>>brand;
      mystore[i].setBrand(brand);
      cout<<"type for item number "<<i<<": ";
      cin>>type;
      mystore[i].setType(type);
      cout<<"Price for item number "<<i<<": ";
      cin>>price;
      mystore[i].setPrice(price)
      cout<<"stock for item number "<<i<<": ";
      cin>>stock;
      mystore[i].setstock(stock);
   }
}
void afis()
{
   for (int i=0;i<MAXPRODUCTS;i++){

      cout
      <<"Brand: "<<mystore[i].getBrand()<<endl
      <<"type:"<<mystore[i].getType()<<endl
      <<"price: "<<mystore[i].getPrice()<<endl
      <<"stock: "<<mystore[i].getStock()<<endl<<endl;
   }
}
void cautabrand()
{
   for(int i=0;i<MAXPRODUCTS;i++)
      {
         if (cauta==mystore[i].getBrand())
            cout
            <<"the brand is: "<<mystore[i].getBrand()<<endl
            <<"the type is: "<<mystore[i].getType()<<endl
            <<"the price is: "<<mystore[i].getPrice()<<endl
            <<"the stock is: "<<mystore[i].getStock()<<endl;
      }
}
void cautatype()
{
   for(int i=0;i<MAXPRODUCTS;i++)
      {
         if (cauta==mystore[i].getType())
            cout
            <<"the brand is: "<<mystore[i].getBrand()<<endl
            <<"the type is: "<<mystore[i].getType()<<endl
            <<"the price is: "<<mystore[i].getPrice()<<endl
            <<"the stock is: "<<mystore[i].getStock()<<endl;
      }
}
int main(){
   int ok=1;
 do
   {       cout << "what action do you want to do?Plese select from the menu." << endl
         << "type 1 if you want to input stock data" << endl
         << "type 2 to show data" << endl
         << "type 3 to search based on Brand" << endl
         << "type 4 to search cars based on Type" << endl
         << "type 0 to exit" << endl;
      cin>>ok;
      cout<<endl<<endl;
      if(ok==1)declarare();
      if(ok==2)afis();
      if(ok==3)
      {
         cout<<"what brand are you looking for?";
         cin>>cauta;
         cautabrand();
      }
      if(ok==4)
      {
         cout<<"what type are you looking for?";
         cin>>cauta;
         cautatype();
      }
      }while(ok);

   delete[] mystore;
   return 0;
```



