**Modelarea orientata pe obiecte a unei aplicatii pentru analiza unui sistem de**

**reglare automata a pH-ului**

**apei potabile**

Proiect realizat de Duta Mihaela Stefania Nicoleta

Profesor coordonator : Sef lucr. dr. mat. Madalina Carbureanu

Grupa 10213

Anul II

Specializarea LCALZ

Facultatea de Inginerie Mecanica si Electrica

Universitatea Petrol-Gaze din Ploiesti

Cuprins

[Generalitati 3](#_Toc125624570)

[Modelul OMT 6](#_Toc125624571)

[Codul sursa al programului 8](#_Toc125624572)

[Rularea programului 17](#_Toc125624573)

[Concluzii 20](#_Toc125624574)

[Bibliografie 21](#_Toc125624575)

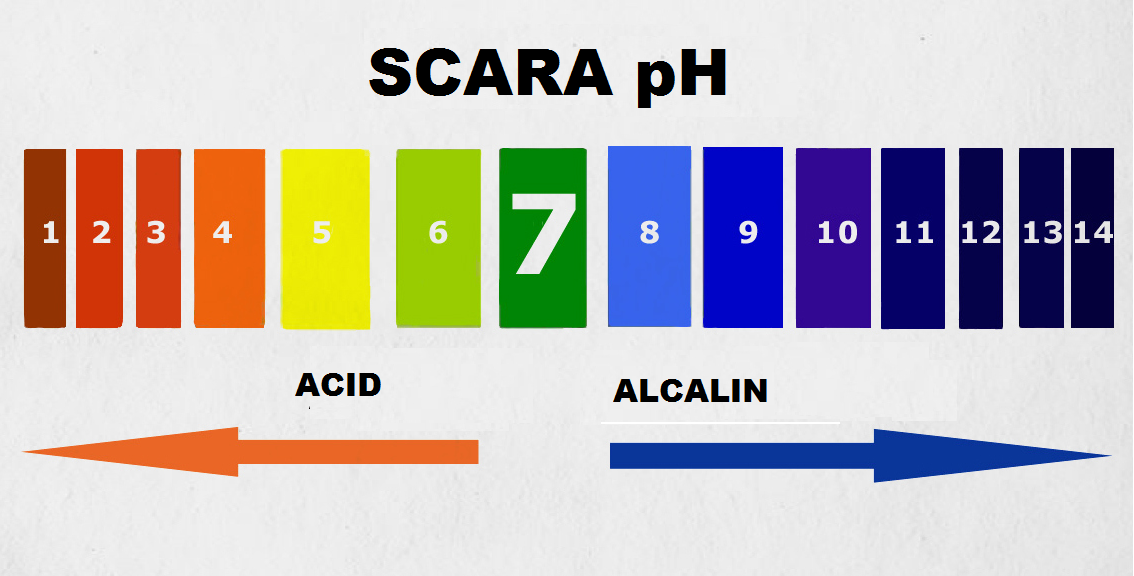
Capitolul 1

# Generalitati

Semnificatia si masurarea pH-ului ocupa un loc important in numeroase domenii practice si stiintifice: in analiza chimica, in controlul si reglarea proceselor tehnologice, in studierea echilibrelor chimice, etc. Fiecare este familiarizat cu termenul de valoare a pH-ului de la ploile acide sau de la balanta pH-ului din cosmetice.

In chimie, pH-ul este o masurare a concentratiei ionilor de hidrogen intr-o solutie pe baza de apa. Un pH mai scazut inseamna ca exista mai multi ioni de hidrogen in lichid, in timp ce un pH mai mare indica mai putini ioni de hidrogen in lichid.

In termeni simpli, pH-ul este o scara de la 1 la 14 care masoara aciditatea sau alcalinitatea unui lichid. La mijlocul scalei se afla apa distilata pura, cu un pH neutru de 7. Orice cu un pH sub 7 este un acid, iar orice cu un pH peste 7 este o baza.



Pentru experti, valoarea pH-ului reprezinta una dintre cele mai importante valori ce caracterizeaza calitatea unui fluid. De ce? Deoarece comportarea chimica a fluidului este determinata de natura sa acida neutra sau alcalina. Cateva exemple:

* raurile sau lacurile care sunt prea acide nu sunt propice vietii animale sau vegetale;
* apa potabila care este prea acida este nesanatoasa si totodata corodeaza tevile de transport ale apei;
* valoarea pH-ului este un factor decisiv pentru o mare varietate de procese din industria chimica,de la obtinerea plasticului pana la desulfurizarea gazului;

Aceasta lucrare este bazata pe analiza unui sistem de reglare automata, care poate fi un instrument important pentru asigurarea calitatii apei potabile, fiind motivul principal pentru care am ales aceasta tema. Consider ca apa este vitala si trebuie sa folosim cele mai bune mecanisme pentru a ne bucura de aceasta in cea mai pura forma.

Apa potabila trebuie sa aiba o valoare a pH-ului de 6,5-8,5. Apa cu pH ridicat poate avea gust de bicarbonat de sodiu, o senzatie alunecoasa si poate lasa depuneri pe corpuri de iluminat.Cu toate acestea, apa cu pH scazut poate avea un gust amar sau metalic.

Cand nivelul pH-ului ajunge mai mic de 7,0, adica nu avem un pH neutru al apei, poate duce la calitatea coroziva a apei. Aceasta inseamna ca va contine zinc, fier, cupru si plumb. Ca rezultat, apa va avea un gust metalic sau amar.

Acest sistem de reglare automata al pH-ului apei potabile reprezinta un sistem de reglare automata dupa abatere. În cazul sistemelor de reglare după abatere comanda se determină pe baza abaterii, abatere rezultată ca urmare a comparației între referință şi reacție. In cadrul programului creat, valoarea de referinta o reprezinta valoarea pH-ului masurat la intrare in proces(pHsp), iar valoarea de reactie este valoarea pH-ului masurata la iesire din proces(pHsp).

La baza sistemului, sta cel mai răspândit algoritm de reglare după abatere si anume algoritmul Proporțional -Integrator- Derivator (PID), iar comanda se calculează conform relației de mai jos: ,unde

- este valoare mărimii de comandă în absența abaterii;

u- valoare curentă a mărimii de comandă;

e – eroarea(abaterea)

- factorul de proporționalitate;

– constanta de integrare;

-constanta de derivare.

Constantele , , se mai numesc și parametrii de acordare. La urmatorul nivel al sistemului, se realizeaza reglarea in functie de valoarea pH-ului, calculata prin intermediul relatiilor bazate pe tipul de caracter: acid sau bazic.

Pentru acizi tari, am folosit relatia: pH = -

Pentru baze tari, am folosit relatia: pH= 14+

Acest sistem poate fi util in industrii cum ar fi cele de tratare a apei, industria berii, industria alimentara, etc.

Este important ca sistemul de reglare automata sa fie construit si intretinut cu atentie, pentru a asigura functionarea corecta si a preveni defectiunile. Sistemele de alarma si monitorizarea calitatii apei pot fi utilizate pentru a detecta si preveni problemele inainte ca acestea sa afecteze calitatea apei.

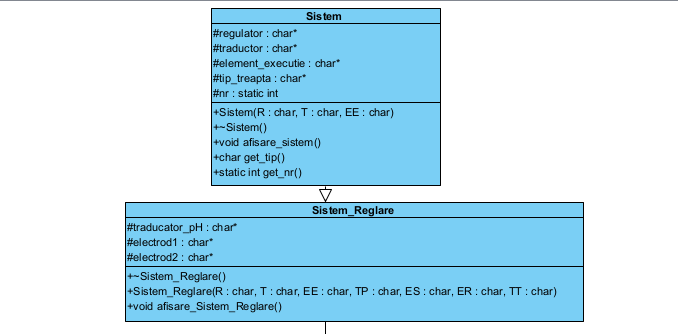
In plus, o sterilizare si filtrare adecvata a apei pot fi necesare pentru a elimina microbi si impuritati din apa, si pentru a preveni problemele de sanatate cauzate de microorganisme periculoase.

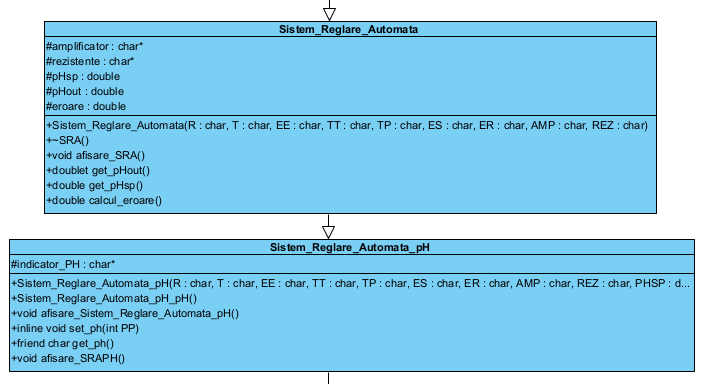


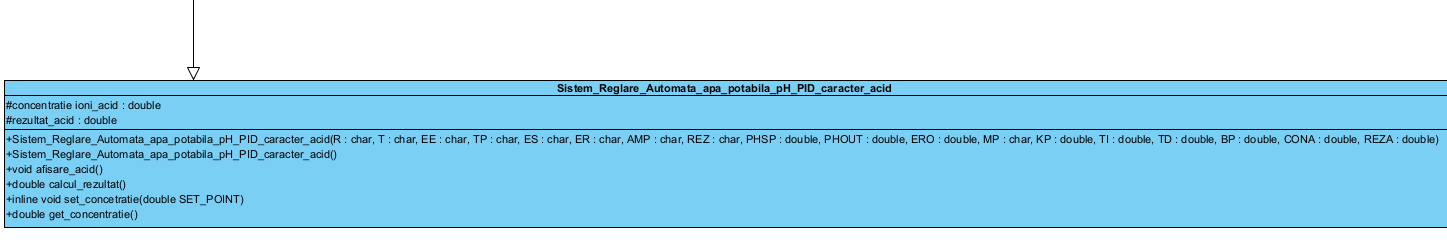
Capitolul 2

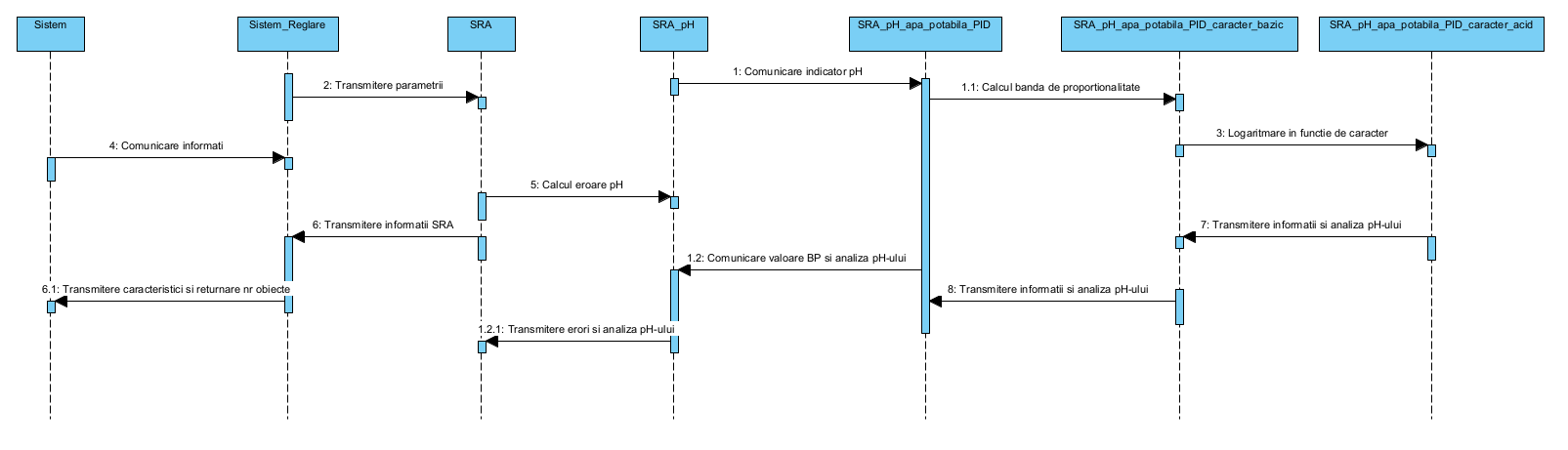
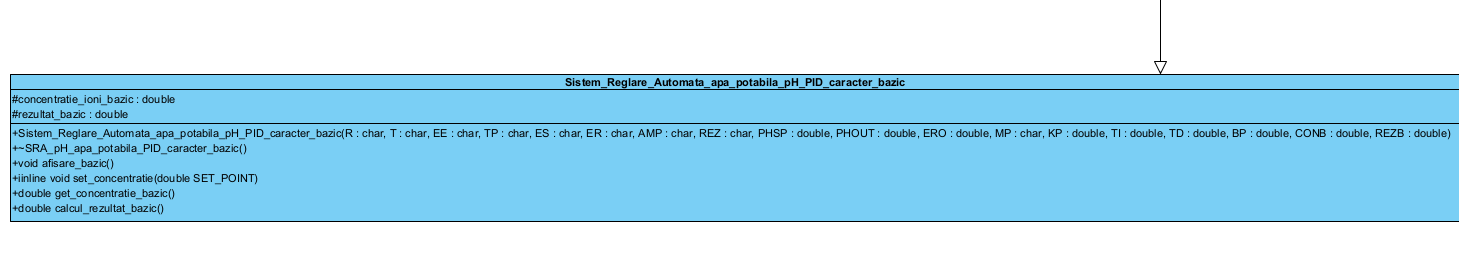
# Modelul OMT

In continuare, voi prezenta clasele si diagrama obiectelor.









Capitolul 3

# Codul sursa al programului

In continuarea proiectului, voi prezenta codul sursa al programului.

|  |
| --- |
| #include <iostream>  #include <string.h>  #include <conio.h>  #include <math.h>  using namespace std;  //interfata clasei Sistem  **class** **Sistem**  {  **protected**: **char** regulator[**100**], traductor[**100**], element\_executie[**100**],tip\_treapta[**100**];  **static** **int** nr;  **public**: //declarare constructor cu parametrii  Sistem(**char** **const**\*, **char** **const**\*, **char** **const**\*,**char** **const**\*);  ~Sistem();//declarare destructor  //declarare functie membru pentru afisare date membru  **void** **afisare\_sistem**();  **char**\*get\_tip();  **static** **int** **get\_nr**();  };  //initializare membru static  **int** **Sistem:**:nr=**0**;  //declarare functie ce returneaza tipul treptei  **char**\***Sistem:**:get\_tip()  {  **return** tip\_treapta;  }  //definire constructor cu parametrii  **Sistem:**:Sistem(**char** **const**\* R, **char** **const**\* T, **char** **const**\* EE,**char** **const**\*TT)  {  strcpy\_s(regulator, R);  strcpy\_s(traductor, T);  strcpy\_s(element\_executie, EE);  strcpy\_s(tip\_treapta,TT);  nr++;  }  //definire destructor  **Sistem:**:~Sistem()  {  cout << "\n Obiect distrus" << endl;  }  //definire functie de afisare  **void** **Sistem:**:afisare\_sistem()  {  cout<<"\nSistem"<<endl;  cout <<"\nRegulator:" << regulator;  cout <<"\nTraductor:" << traductor;  cout <<"\nElement executie:" << element\_executie;  }  //definire functie membru statica  **int** **Sistem:**:get\_nr()  {  **return** nr;  } |

|  |
| --- |
| //interfata clasei Sistem\_Reglare  **class** **Sistem\_Reglare** : **public** Sistem  {  **protected**: **char** traductor\_pH[**100**], electrod1[**100**], electrod2[**100**];  //zona declarare metode  **public**: //declarare constructor cu parametrii  Sistem\_Reglare(**char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*,**char** **const**\*);  //declarare destructor  ~Sistem\_Reglare();  //declarare functie membru de afisare  **void** **afisare\_Sistem\_Reglare**();  };  //definire constructor cu parametrii  **Sistem\_Reglare:**:Sistem\_Reglare(**char** **const**\* R, **char** **const**\* T, **char** **const**\* EE, **char** **const**\*TT, **char** **const**\* TP, **char** **const**\* ES, **char** **const**\* ER) :Sistem(R, T, EE,TT)  {  strcpy\_s(traductor\_pH, TP);  strcpy\_s(electrod1, ES);  strcpy\_s(electrod2, ER);  }  //definire destructor  **Sistem\_Reglare:**:~Sistem\_Reglare()  {  cout << "\nObiect distrus";  }  **void** **Sistem\_Reglare:**:afisare\_Sistem\_Reglare()  {  cout << "\-Sistem Reglare";  **Sistem:**:afisare\_sistem();  cout << "\nTraductor pH:" << traductor\_pH;  cout << "\nElectrodul nr 1:" << electrod1;  cout << "\nElectrodul nr 2:" << electrod2;}  //interfata clasei Sistem\_Reglare\_Automata  **class** **Sistem\_Reglare\_Automata** : **public** Sistem\_Reglare  {  **protected**: **char** amplificator[**100**], rezistente[**100**];  **double** pHsp, pHout, eroare;  //metode  **public**: Sistem\_Reglare\_Automata(**char** **const**\*,**char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **double**, **double**, **double**);  ~Sistem\_Reglare\_Automata();  **void** **afisare\_SRA**();  inline **void** **set\_phsp**(**double** SET\_POINT) { pHsp = SET\_POINT; }  **double** **get\_pHout**();  **double** **get\_pHsp**();  **double** **calcul\_eroare**();  };  //definire functie ce returneaza valoarea pH-ului la iesirea din proces pHout  **double** **Sistem\_Reglare\_Automata:**:get\_pHout()  {  **return** pHout;  }  //definire functie ce returneaza valoarea de referinta a pH-ului  **double** **Sistem\_Reglare\_Automata:**:get\_pHsp()  {  **return** pHsp;  }  //definire constructor cu parametrii  **Sistem\_Reglare\_Automata:**:Sistem\_Reglare\_Automata(**char** **const**\* R, **char** **const**\* T, **char** **const**\* EE, **char** **const**\*TT, **char** **const**\* TP, **char** **const**\* ES, **char** **const**\* ER, **char** **const**\* AM, **char** **const**\* REZ, **double** PHSP, **double** PHOUT, **double** ERR):Sistem\_Reglare(R,T,EE,TT,TP,ES,ER)  {  strcpy\_s(amplificator, AM);  strcpy\_s(rezistente, REZ);  pHsp = PHSP;  pHout = PHOUT;  eroare=ERR;  }  //definire destructor-dealocare dinamica  **Sistem\_Reglare\_Automata:**:~Sistem\_Reglare\_Automata()  {  delete amplificator;  delete rezistente;  }  **void** **Sistem\_Reglare\_Automata:**:afisare\_SRA()  {  //redefinirea functiei de afisare din clasa Sistem\_Reglare  cout<<"\--Sistem de Reglare Automata"<<endl;  **Sistem\_Reglare:**:afisare\_Sistem\_Reglare();  cout<<"\nAmplificator:"<<amplificator;  cout<<"\nRezistente:"<<rezistente;  }  //definire functie pentru calculul erorii  **double** **Sistem\_Reglare\_Automata:**:calcul\_eroare()  {  **return** **get\_pHsp**()-get\_pHout();  } |

|  |
| --- |
| //interfata clasei Sistem\_Reglare\_Automata\_pH  **class** **Sistem\_Reglare\_Automata\_pH**:**public** Sistem\_Reglare\_Automata  {  **protected**: **char**\*indicator\_ph;  **public**: Sistem\_Reglare\_Automata\_pH(**char** **const**\*,**char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **double**, **double**, **double**,**char** **const**\*);  ~Sistem\_Reglare\_Automata\_pH();  inline **void** **set\_indicator**(**char**\*SET\_POINT)  { indicator\_ph = SET\_POINT; }  friend **char**\*get\_indicator(Sistem\_Reglare\_Automata\_pH&);  **void** **afisare\_SRAPH**();  };  //definire constructor cu parametrii  **Sistem\_Reglare\_Automata\_pH:**:Sistem\_Reglare\_Automata\_pH(**char** **const**\* R, **char** **const**\* T, **char** **const**\* EE, **char** **const**\*TT, **char** **const**\* TP, **char** **const**\* ES, **char** **const**\* ER, **char** **const**\* AM, **char** **const**\* REZ, **double** PHSP, **double** PHOUT, **double** ERR,**char** **const**\*IND):Sistem\_Reglare\_Automata(R,T,EE,TT,TP,ES,ER,AM,REZ,PHSP,PHOUT,ERR)  {  indicator\_ph=**new** **char**[strlen(IND)+**1**];  strcpy(indicator\_ph,IND);  }  //definire destructor-dealocare dinamica  **Sistem\_Reglare\_Automata\_pH:**:~Sistem\_Reglare\_Automata\_pH()  {  delete indicator\_ph;  }  **char**\*get\_indicator(Sistem\_Reglare\_Automata\_pH&X)  {  **return** X.indicator\_ph;  }  **void** **Sistem\_Reglare\_Automata\_pH:**:afisare\_SRAPH()  {  cout<<"\n---Sistem de Reglare Automata pH"<<endl;  **Sistem\_Reglare\_Automata:**:afisare\_SRA();  cout<<"\nCuloarea indicatorului este:"<<indicator\_ph;  } |

|  |
| --- |
| //interfata clasei Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID  **class** **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID**:**public** Sistem\_Reglare\_Automata\_pH  {  **protected**: **char**\*micro\_procesor;  **double** kp,td,ti,bp;  **public**: Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID(**char** **const**\*,**char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **double**, **double**, **double**,**char** **const**\*,**char** **const**\*,**double**,**double**,**double**,**double**);  ~Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID();  **char**\*get\_proces();  inline **void** **set\_KP**(**double** SET\_POINT)  { kp = SET\_POINT; }  **double** **get\_KP**();  **double** **get\_BP**();  **void** **afisre\_SRAPID**();  };  //definire constructor cu parametrii  **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID:**:Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID(**char** **const**\* R, **char** **const**\* T, **char** **const**\* EE, **char** **const**\*TT, **char** **const**\* TP, **char** **const**\* ES, **char** **const**\* ER, **char** **const**\* AM, **char** **const**\* REZ, **double** PHSP, **double** PHOUT, **double** ERR,**char** **const**\*IND,**char** **const**\*MP,**double** KP,**double** TD,**double** TI,**double** BP):Sistem\_Reglare\_Automata\_pH(R,T,EE,TT,TP,ES,ER,AM,REZ,PHSP,PHOUT,ERR,IND)  {  micro\_procesor=**new** **char**[strlen(MP)+**1**];  strcpy(micro\_procesor,MP);  kp=KP;  td=TD;  ti=TI;  bp=BP;  }  **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID:**:~Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID()  {  delete micro\_procesor;  }  //definire functie ce returneaza tipul micro procesorului  **char**\***Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID:**:get\_proces()  {  **return** micro\_procesor;  }  //definire functie ce returneaza constanta de proportionalitate  **double** **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID:**:get\_KP()  {  **return** kp;  }  //definire functie ce returneaza banda de proportionalitate  **double** **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID:**:get\_BP()  {  **return** (**100**/get\_KP());  }  **void** **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID:**:afisre\_SRAPID()  {  cout<<"\n----Sistem de Reglare Automata Apa Potabila pH PID";  //Sistem\_Reglare\_Automata\_pH::afisare\_SRAPH();  cout<<"\nMicro-procesor:"<<micro\_procesor;  cout<<"\nKP:"<<kp;  cout<<"\nTD:"<<td;  cout<<"\nTI:"<<ti;  cout<<"\nBP:"<<bp;  } |

|  |
| --- |
| //interfata clasei Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_acid  **class** **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_acid**:**public** Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID  {  **protected**: **double** concentratie\_ioni\_acid,rezultat\_acid;  **public**: Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_acid(**char** **const**\*,**char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **double**, **double**, **double**,**char** **const**\*,**char** **const**\*,**double**,**double**,**double**,**double**,**double**,**double**);  ~Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_acid();  inline **void** **set\_concentratie**(**double** SET\_POINT) { concentratie\_ioni\_acid = SET\_POINT; }  **double** **get\_concentratie**();  **double** **calcul\_rezultat**();  **void** **afisare\_acid**();  };  **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_acid:**:Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_acid(**char** **const**\* R, **char** **const**\* T, **char** **const**\* EE, **char** **const**\*TT, **char** **const**\* TP, **char** **const**\* ES, **char** **const**\* ER, **char** **const**\* AM, **char** **const**\* REZ, **double** PHSP, **double** PHOUT, **double** ERR,**char** **const**\*IND,**char** **const**\*MP,**double** KP,**double** TD,**double** TI,**double** BP,**double** CON,**double** RREZ):Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID(R,T,EE,TT,TP,ES,ER,AM,REZ,PHSP,PHOUT,ERR,IND,MP,KP,TD,TI,BP)  {  concentratie\_ioni\_acid=CON;  rezultat\_acid= -log(concentratie\_ioni\_acid);  }  **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_acid:**:~Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_acid()  {  cout<<"\nDestructor obiect!";  }  //definire functie ce returneaza valoarea concentratiei de ioni  **double** **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_acid:**:get\_concentratie()  {  **return** concentratie\_ioni\_acid;  }  //definire functie ce calculeaza rezultatul neutralizarii concentratiei de ioni  **double** **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_acid:**:calcul\_rezultat()  {  **return** -log(get\_concentratie());  }  **void** **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_acid:**:afisare\_acid()  {  cout<<"\n-----Sistem Reglare Automata Apa Potabila pH PID Caracter Acid"<<endl;  **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID:**:afisre\_SRAPID();  cout<<"\nConcentratia de ioni-acid este:"<<concentratie\_ioni\_acid;  cout<<"\nRezultatul-acid este:"<<rezultat\_acid;  } |

|  |
| --- |
| //interfata clasei Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_bazic  **class** **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_bazic**:**public** Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID  {  **protected**: **double** concentratie\_ioni\_bazic,rezultat\_bazic;  **public**: Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_bazic(**char** **const**\*,**char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **char** **const**\*, **double**, **double**, **double**,**char** **const**\*,**char** **const**\*,**double**,**double**,**double**,**double**,**double**,**double**);  ~Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_bazic();  inline **void** **set\_concentratie**(**double** SET\_POINT) { concentratie\_ioni\_bazic = SET\_POINT; }  **double** **get\_concentratie\_bazic**();  **double** **calcul\_rezultat\_bazic**();  **void** **afisare\_bazic**();  };  **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_bazic:**:Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_bazic(**char** **const**\* R, **char** **const**\* T, **char** **const**\* EE, **char** **const**\*TT, **char** **const**\* TP, **char** **const**\* ES, **char** **const**\* ER, **char** **const**\* AM, **char** **const**\* REZ, **double** PHSP, **double** PHOUT, **double** ERR,**char** **const**\*IND,**char** **const**\*MP,**double** KP,**double** TD,**double** TI,**double** BP,**double** CONB,**double** RREZB):Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID(R,T,EE,TT,TP,ES,ER,AM,REZ,PHSP,PHOUT,ERR,IND,MP,KP,TD,TI,BP)  {  concentratie\_ioni\_bazic=CONB;  rezultat\_bazic= log(concentratie\_ioni\_bazic);  }  **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_bazic:**:~Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_bazic()  {  cout<<"\nObiect distrus";  }  //definire functie ce returneaza valoarea concentratiei de ioni  **double** **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_bazic:**:get\_concentratie\_bazic()  {  **return** concentratie\_ioni\_bazic;  }  //definire functie ce calculeaza rezultatul neutralizarii concentratiei de ioni  **double** **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_bazic:**:calcul\_rezultat\_bazic()  {  **return** **14**+log(get\_concentratie\_bazic());  }  **void** **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_bazic:**:afisare\_bazic()  {  cout<<"\n------Sistem Reglare Automata Apa Potabila pH PID Caracter Bazic"<<endl;  **Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID:**:afisre\_SRAPID();  cout<<"\nConcentratia de ioni-bazic este:"<<concentratie\_ioni\_bazic;  cout<<"\nRezultatul-bazic este:"<<rezultat\_bazic;  } |

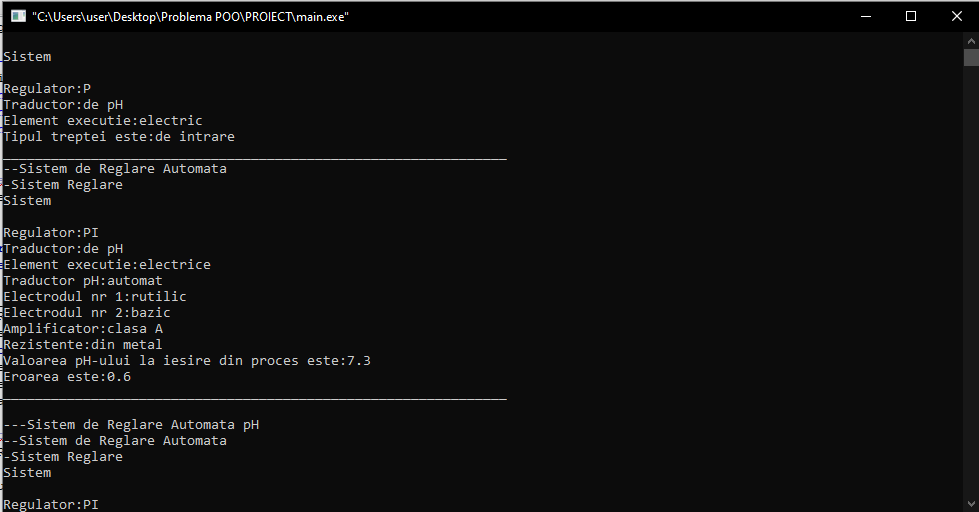
Programul principal:

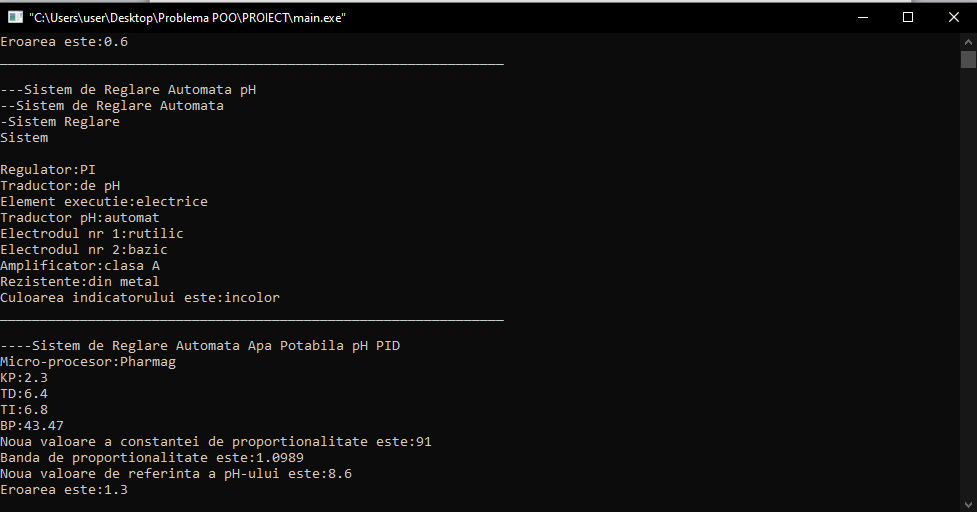
|  |
| --- |
| **int** **main**()  {  Sistem **ob1**("P","de pH","electric","de intrare");  ob1.afisare\_sistem();  cout<<"\nTipul treptei este:"<<ob1.get\_tip();  cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;  Sistem\_Reglare\_Automata **ob2**("PI","de pH","electrice","de intrare","automat","rutilic","bazic","clasa A","din metal",**7.2**,**7.3**,**0.1**);  ob2.afisare\_SRA();  ob2.set\_phsp(**7.9**);  cout<<"\nValoarea pH-ului la iesire din proces este:"<<ob2.get\_pHout();  cout<<"\nEroarea este:"<<ob2.calcul\_eroare();  cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;  Sistem\_Reglare\_Automata\_pH **ob3**("PI","de pH","electrice","de intrare","automat","rutilic","bazic","clasa A","din metal",**7.2**,**7.3**,**0.1**,"incolor");  ob3.afisare\_SRAPH();  cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;  Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID **ob4**("PID","de reglare","electrice","de intrare","automat","rutilic","bazic","clasa A","din metal",**7.2**,**7.3**,**0.1**,"incolor","Pharmag",**2.3**,**6.4**,**6.8**,**43.47**);  ob4.afisre\_SRAPID();  ob4.set\_KP(**91**);  cout<<"\nNoua valoare a constantei de proportionalitate este:"<<ob4.get\_KP();  cout<<"\nBanda de proportionalitate este:"<<ob4.get\_BP();  ob4.set\_phsp(**8.6**);  cout<<"\nNoua valoare de referinta a pH-ului este:"<<ob4.get\_pHsp();  cout<<"\nEroarea este:"<<ob4.calcul\_eroare();  cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;  Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_acid **ob5**("PID","de neutralizare","electrice","de intrare","automat","rutilic","bazic","clasa B","din metal",**8.1**,**7.5**,**0.6**,"rosu","portabil",**2.1**,**5.2**,**5.8**,**47.61**,**0.2**,-**0.69**);  ob5.afisare\_acid();  ob5.set\_concentratie(**0.004**);  cout<<"\nNoua concentratie este:"<<ob5.get\_concentratie();  cout<<"\nValoarea ph-ului este:"<<ob5.calcul\_rezultat();  cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;  Sistem\_Reglare\_Automata\_apa\_potabila\_pH\_PID\_caracter\_bazic **ob6**("PID","de neutralizare","electrice","de intrare","automat","rutilic","bazic","clasa B","din metal",**7.6**,**7.2**,**0.4**,"verde","portabil",**2.5**,**4.3**,**4.9**,**40**,**400**,**16.60**);  ob6.afisare\_bazic();  ob6.set\_concentratie(**0.002**);  cout<<"\nNoua concentratie este:"<<ob6.get\_concentratie\_bazic();  cout<<"\nValoarea pH-ului este:"<<ob6.calcul\_rezultat\_bazic();  cout<<"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;  cout<<"\nNumarul de obiecte create: "<<**Sistem:**:get\_nr();} |

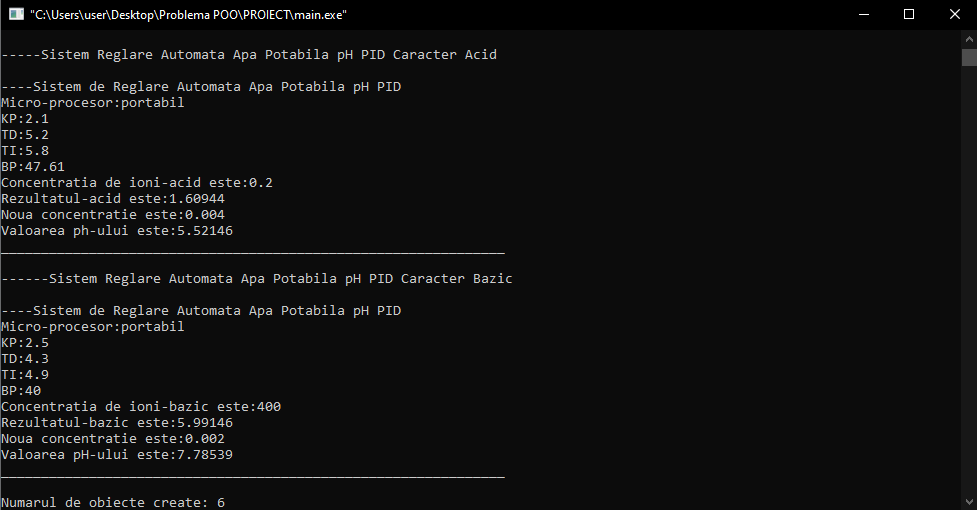
Capitolul 4

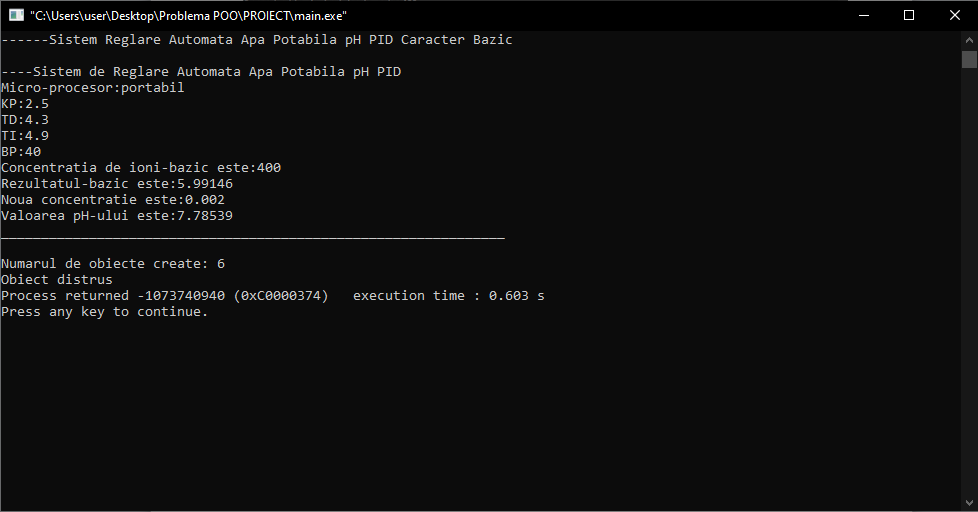
# Rularea programului

In continuarea proiectului, voi arata cum ruleaza programul.

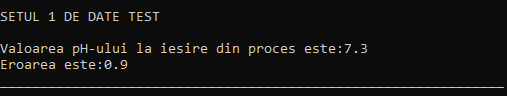




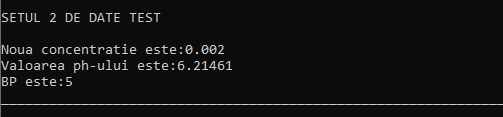




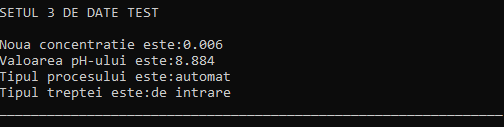
Pentru primul set de date de test, valoarea pH-ului de referinta este 8.2, iar valoarea pH-ului la iesire din process este 7.3. Eroarea va fi 0.9.



Pentru al doilea set de date de test, valoarea concentratiei este 0.002. Valoarea pH-ului trebuie sa fie 6.21. Kp ia valoarea 20, astfel BP trebuie sa aiba valoarea 5.



Pentru al 3 lea set de date de test, valoarea concentratiei este 0.004. Valoara pH-ului trebuie sa fie 8.88. Tipul treptei trebuie sa fie de intrare, iar procesul este unul automat.



In urma rularii celor 3 seturi de date de test, se observa ca programul returneaza valorile corecte, cele asteptate de utilizator, fara erori si fara probleme. Scrierea este usor de inteles si urmarit, astfel programul este usor accesibil tututor utilizatorilor.

Capitolul 5

# Concluzii

In urma aprofundarii materialelor necesare pentru realizarea aplicatiei, am consolidat baza materiei Introducere in automatica si Calculatoare, studiata in anul 1, materie necesara pentru studentii de la Calculatoare. Am invatat cum sa construiesc si sa gandesc o diagrama corecta a obiectelor si cum sa modelez un program de la 0, conform acesteia. Realizarea programului nu a fost usoara si a necesitat multa atentie pentru evitarea erorilor, pe care le-am corectat eficient si am reusit sa inteleg cat de greu este sa mentii un program optim pentru un sistem de reglare automata a pH-ului apei potabile. De asemenea, in urma dezvoltarii constante a programului, cunostiintele mele legate de cursul si laborator de Programare Orientata pe Obiecte s-au consolidat si imi va fi mai usor sa studiez pentru examen si sa inteleg urmatoarele cursuri de programare din viitori ani de facultate.

Capitolul 6

# Bibliografie

Am folosit urmatoarele materiale:

<https://drive.google.com/file/d/14JjXkw-zAxap2AsDpX2T1uC7cuBOtsbu/view>

<http://ac.upg-ploiesti.ro/cursuri/apich2/APCh_II_6.pdf>

<https://ime.upg-elearning.ro/pluginfile.php/16597/mod_resource/content/2/Lucrarea_4_IAC_10213.pdf>

<https://ime.upg-elearning.ro/pluginfile.php/16596/mod_resource/content/1/Lucrarea_3_IAC_10213.pdf>

<https://www.heizer.hu/ro/tratarea-apei/>

<https://ro.eferrit.com/cum-se-calculeaza-ph-ul-revizuire-rapida/>

<https://smartliving.ro/ce-este-ph-ul-si-cand-ne-pune-viata-in-pericol/>