Міністерство освіти і науки України

Національний лісотехнічний університет України

Кафедра інформаційних технологій

**Звіт до лабораторної роботи №2**

з навчальної дисципліни

**«ООП»**

на тему:

«РОБОТА З КЛАСАМИ, МАСИВИ ОБ’ЄКТІВ»

**Виконав:**

Студент групи КН-21/2

Манжула Д. В.

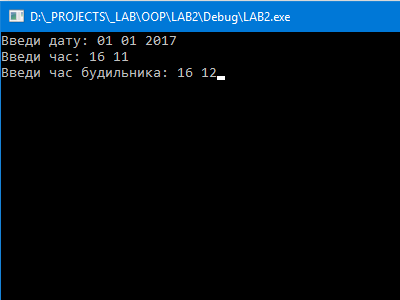
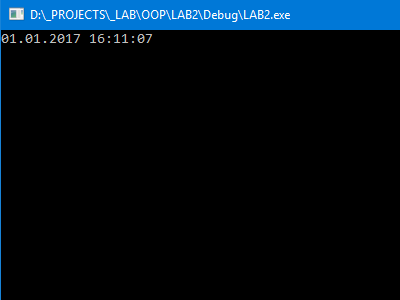
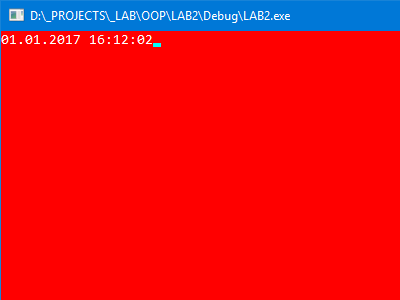
**Перевірив:**

Семенишин Н.О.

Львів – 2017

**Завдання**

Клас ЦИФРОВИЙ ГОДИННИК містить об’єкти класів КОНТРОЛЕР ЧАСУ і КОНТРОЛЕР ДАТИ, ЕКРАН для відображення часу. Визначити необхідні дані, конструктори, деструктори та методи роботи з елементами даних. Встановити поточні дату і час. Встановити час будильника. Зімітувати спрацьовування будильника.

|  |  |
| --- | --- |
| DateController.h | TimeController.h |
| #pragma once  class DateController  {  public:  DateController(void);  ~DateController(void);  void setDate(int y, int m, int d);  void addDay(void);  int getYear(void);  int getMonth(void);  int getDay(void);  private:  int y, m, d;  }; | #pragma once  class TimeController  {  public:  TimeController(void);  ~TimeController(void);  void setTime(int h, int m);  bool addSecond(void);  int getHours(void);  int getMinutes(void);  int getSeconds(void);  private:  int h, m, s;  }; |

|  |  |
| --- | --- |
| DigitalClock.h | Display.h |
| #pragma once  #include "TimeController.h"  #include "DateController.h"  #include "Display.h"  class DigitalClock  {  public:  DigitalClock(void);  ~DigitalClock(void);  void start(void);  void process(void);  void alarm(void);  void setTime(int h, int m);  void setDate(int y, int m, int d);  void setAlarm(int h, int m);  private:  TimeController time;  DateController date;  Display display;  int alarmH, alarmM;  bool stopped;  }; | #pragma once  #include "TimeController.h"  #include "DateController.h"  #include "Display.h"  class Display  {  public:  Display(void);  ~Display(void);  void render(DateController, TimeController);  }; |

|  |
| --- |
| DigitalClock.cpp |
| #include "DigitalClock.h"  #include <iostream>  #include <windows.h>  DigitalClock::DigitalClock(void){  this->stopped = true;  }  DigitalClock::~DigitalClock(void){}  void DigitalClock::start(void){  this->stopped = false;  while(!this->stopped){  this->process();  Sleep(1000);  }  }  void DigitalClock::process(void){  bool needAddDay = this->time.addSecond();  if(needAddDay){  this->date.addDay();  }  display.render(this->date, this->time);  if(  this->time.getHours() == this->alarmH &&  this->time.getMinutes() == this->alarmM  ){  this->alarm();  }  }  void DigitalClock::alarm(void){  system("color cf");  Beep(1400, 10);  }  void DigitalClock::setTime(int h, int m){  this->time.setTime(h, m);  }  void DigitalClock::setDate(int y, int m, int d){  this->date.setDate(y, m, d);  }  void DigitalClock::setAlarm(int h, int m){  this->alarmH = h;  this->alarmM = m;  } |

|  |
| --- |
| DateController.cpp |
| #include "DateController.h"  DateController::DateController(void){  this->y = 2017;  this->m = 9;  this->d = 27;  }  DateController::~DateController(void){}  void DateController::setDate(int y, int m, int d){  this->y = y;  this->m = m;  this->d = d;  }  void DateController::addDay(void){  this->d += 1;  if(this->d > 30){  this->d = 1;  this->m += 1;  if(this->m > 12){  this->m = 1;  this->y += 1;  }  }  }  int DateController::getYear(void){return this->y;}  int DateController::getMonth(void){return this->m;}  int DateController::getDay(void){return this->d;} |

|  |
| --- |
| TimeController.cpp |
| #include "TimeController.h"  TimeController::TimeController(void){  this->h = 0;  this->m = 0;  this->s = 0;  }  TimeController::~TimeController(void){}  void TimeController::setTime(int h, int m){  this->h = h;  this->m = m;  this->s = 0;  }  bool TimeController::addSecond(void){  this->s += 1;  if(this->s > 59){  this->s = 0;  this->m += 1;  if(this->m > 59){  this->m = 0;  this->h += 1;  if(this->h > 23){  this->h = 0;  return true;  }  }  }  return false;  }  int TimeController::getHours(void){return this->h;}  int TimeController::getMinutes(void){return this->m;}  int TimeController::getSeconds(void){return this->s;} |

|  |
| --- |
| Display.cpp |
| #include "Display.h"  #include <iostream>  using namespace std;  Display::Display(void){}  Display::~Display(void){}  void Display::render(DateController date, TimeController time){  int  y = date.getYear(),  m = date.getMonth(),  d = date.getDay(),  hh = time.getHours(),  mm = time.getMinutes(),  ss = time.getSeconds();    cout << "\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b";  if(d<10)cout<<"0";cout<<d;  cout << ".";  if(m<10)cout<<"0";cout<<m;  cout << ".";  cout<<y;  cout << " ";  if(hh<10)cout<<"0";cout<<hh;  cout << ":";  if(mm<10)cout<<"0";cout<<mm;  cout << ":";  if(ss<10)cout<<"0";cout<<ss;  } |

|  |
| --- |
| Main.cpp |
| #include <iostream>  #include "DigitalClock.h"  using namespace std;  void main(){  setlocale(LC\_ALL, "Ukrainian");  DigitalClock clock;  int y, m, d, hh, mm, ah, am;  cout << "Введи дату: d m yyyy\b\b\b\b\b\b\b\b";  cin >> d >> m >> y;  cout << "Введи час: hh mm\b\b\b\b\b";  cin >> hh >> mm;    cout << "Введи час будильника: hh mm\b\b\b\b\b";  cin >> ah >> am;  system("cls");  clock.setDate(y, m, d);  clock.setTime(hh, mm);  clock.setAlarm(ah, am);  clock.start();  } |