ОДЕССКИЙ НАЦИОНАЛЬНЫЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ИНСТИТУТ КОМПЬЮТЕРНЫХ СИСТЕМ

Кафедра информационных систем

**Лабораторная работа №1**

по дисциплине: «Технологии создания программного продукта»   
на тему: «Знакомство с базовым синтаксисом языка С++»

**Выполнила**:

ст. гр. НАИ-166

Граматик А.А.

**Проверили**:

Бабич Н.И.

Трояновская Ю.Л.

Одесса 2018

Цели лабораторной работы:

* установить необходимое ПО для программирования на С++;
* ознакомиться с базовым синтаксисом языка и основными операторами;
* изучить примеры соглашения по коду для С++.

Задания на лабораторную работу:

Создать классы, их объекты и реализовать взаимодействие. Определить общий объем дома, общее количество литров краски, количество паркета, количество стекла. Необходимо реализовать следующие классы.

Класс House:

* количество этажей;
* количество квартир;
* квартиры: Flat.

Класс Flat:

* количество комнат;
* комнаты: Room.

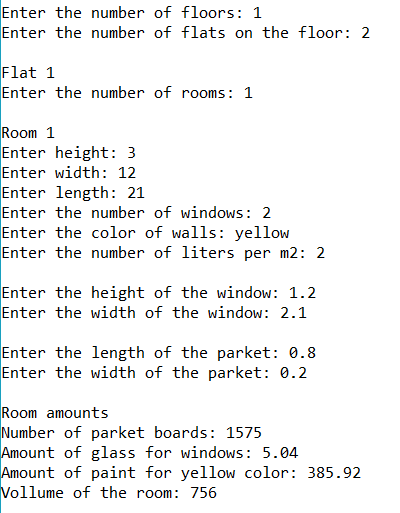
Класс Room:

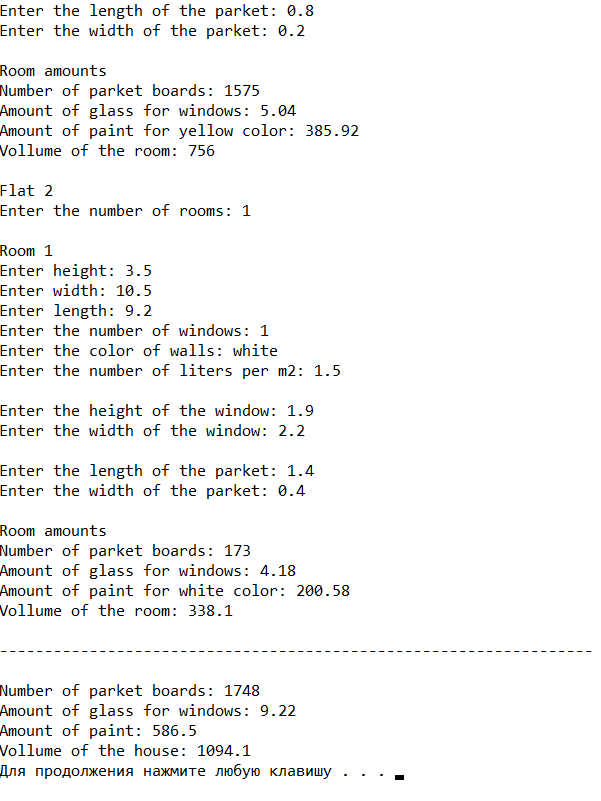
* l, h, w;
* цвет;
* количество литров на м2;
* паркет (l, w) – отдельный класс;
* количество окон (w, h) – отдельный класс.

Код программы:

|  |
| --- |
| #include "pch.h"  #include "math.h"  #include <iostream>  #include <string>  using namespace std;  class Parket {  private:  float length, width;  public:  void getParams() {  cout << "Enter the length of the parket: ";  cin >> length;  cout << "Enter the width of the parket: ";  cin >> width;  }  float getParketSurface() {  return length \* width;  }  };  class Window {  private:  float height, width;  public:  void getParams() {  cout << "Enter the height of the window: ";  cin >> height;  cout << "Enter the width of the window: ";  cin >> width;  }  float getWindowSurface() {  return height \* width;  }  };  class Room {  private:  float length, width, height;  int numWindows;  string color;  float LperM2;  Window window;  Parket parket;  public:  float getFloorSurface() {  return length \* width;  }  float getWallSurface() {  return height \* (length + width) \* 2;  }  float getVollume() {  return height \* length \* width;  }  float getPaintAmount() {  return (getWallSurface() - getGlassAmount()) \* LperM2;  }  float getGlassAmount() {  return window.getWindowSurface() \* numWindows;  }  int getParketAmount() {  return ceil(getFloorSurface() / parket.getParketSurface());  }  void getParams() {  cout << endl << "Enter height: ";  cin >> height;  cout << "Enter width: ";  cin >> width;  cout << "Enter length: ";  cin >> length;  cout << "Enter the number of windows: ";  cin >> numWindows;  cout << "Enter the color of walls: ";  cin >> color;  cout << "Enter the number of liters per m2: ";  cin >> LperM2;  cout << endl;  window.getParams();  cout << endl;  parket.getParams();  }  void printRoomAmounts() {  cout << endl << "Room amounts" << endl;  cout << "Number of parket boards: " << getParketAmount() << endl;  cout << "Amount of glass for windows: " << getGlassAmount() << endl;  cout << "Amount of paint for " << color << " color: " << getPaintAmount() << endl;  cout << "Vollume of the room: " << getVollume() << endl;  }  };  class Flat {  private:  int numRooms;  Room\* rooms;  float flatVoll = 0;  float paintAmount = 0;  int parketAmount = 0;  float glassAmount = 0;  public:  void getParams() {  cout << endl << "Enter the number of rooms: ";  cin >> numRooms;  rooms = new Room[numRooms];  for(int i = 0; i < numRooms; ++i) {  cout << endl << "Room " << i + 1;  rooms[i].getParams();  rooms[i].printRoomAmounts();  }  }  float getFlatVoll() {  for (int i = 0; i < numRooms; ++i) {  flatVoll += rooms[i].getVollume();  }  return flatVoll;  }  float getPaintAmount() {  for (int i = 0; i < numRooms; ++i) {  paintAmount += rooms[i].getPaintAmount();  }  return paintAmount;  }  float getGlassAmount() {  for (int i = 0; i < numRooms; ++i) {  glassAmount += rooms[i].getGlassAmount();  }  return glassAmount;  }  int getParketAmount() {  for (int i = 0; i < numRooms; ++i) {  parketAmount += rooms[i].getParketAmount();  }  return parketAmount;  }  void printFlatAmounts() {  cout << endl << "Number of parket boards: " << getParketAmount() << endl;  cout << "Amount of glass for windows: " << getGlassAmount() << endl;  cout << "Amount of paint: " << getPaintAmount() << endl;  cout << "Vollume of the flat: " << getFlatVoll() << endl;  }  void destructor() {  delete[] rooms;  }  };  class House {  private:  int numFloors;  int numFlats;  Flat\* flats;  float houseVoll = 0;  float paintAmount = 0;  int parketAmount = 0;  float glassAmount = 0;  public:  void getParams() {  cout << "Enter the number of floors: ";  cin >> numFloors;  cout << "Enter the number of flats on the floor: ";  cin >> numFlats;  flats = new Flat[numFloors \* numFlats];  for (int i = 0; i < numFloors \* numFlats; ++i) {  cout << endl << "Flat " << i + 1;  flats[i].getParams();  }  }  float getHouseVoll() {  for (int i = 0; i < numFloors \* numFlats; ++i) {  houseVoll += flats[i].getFlatVoll();  }  return houseVoll;  }  float getPaintAmount() {  for (int i = 0; i < numFloors \* numFlats; ++i) {  paintAmount += flats[i].getPaintAmount();  }  return paintAmount;  }  float getGlassAmount() {  for (int i = 0; i < numFloors \* numFlats; ++i) {  glassAmount += flats[i].getGlassAmount();  }  return glassAmount;  }  int getParketAmount() {  for (int i = 0; i < numFloors \* numFlats; ++i) {  parketAmount += flats[i].getParketAmount();  }  return parketAmount;  }  void printHouseAmounts() {  cout << endl << "------------------------------------------------------------------" << endl;  cout << endl << "Number of parket boards: " << getParketAmount() << endl;  cout << "Amount of glass for windows: " << getGlassAmount() << endl;  cout << "Amount of paint: " << getPaintAmount() << endl;  cout << "Vollume of the house: " << getHouseVoll() << endl;  }  void destructor() {  delete[] flats;  }  };  int main(){    House house;  house.getParams();  house.printHouseAmounts();  system("pause");  return 0;  } |

Примеры работы программы:





Выводы

В ходе данной лабораторной работы были изучены базовый синтаксис и основные операторы языка программирования С++. Был также изучен объектно-ориентированный подход к программированию на языке. Были успешно выполнены все задания, а именно реализация взаимодействия между объектами классов. Был приведен рабочий исходный код, а также примеры работы каждого из методов с различными данными.