Міністерство освіти і науки України

Національний лісотехнічний університет України

Кафедра інформаційних технологій

**Звіт до лабораторної роботи №3**

з навчальної дисципліни

**«ООП»**

на тему:

«опрацювання виняткових ситуацій в C++, перевантаження операторів для класу. Робота з конструкторами та перевантаження методів класу»

**Виконав:**

Студент групи КН-21/2

Манжула Д. В.

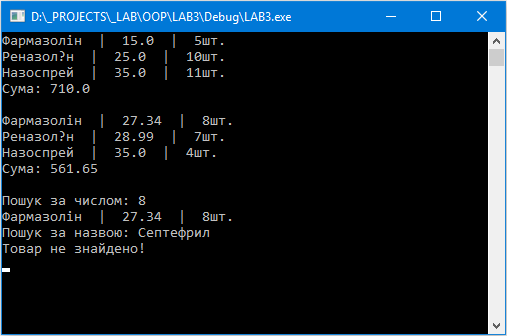
**Перевірив:**

Семенишин Н.О.

Львів – 2017

**Завдання**

Створити клас ЛІКИ, які містять: назву, кількість одиниць товару, вартість. Визначити вартість ліків за рецептом. Перевантажити операції (+=, -=) для зміни кількості одиниць товару, (+, -) для зміни вартості товару. Первантажити методи вартості ліків за рецептом та пошуку ліків за різними критеріями. Передбачити обробку виняткових ситуацій при встановленні та читанні числових полів класу ЛІКИ.



|  |  |
| --- | --- |
| Файл Main.cpp | |
| #include <iostream>  #include "DrugStore.h"  #include "Drug.h"  using namespace std;  void main(){  setlocale(LC\_ALL, "Ukrainian");  DrugStore store;  store.addItem("Farmazolin", 1500, 5);  store.addItem("Renazolin", 2500, 10);  store.addItem("Nazosprei", 3500, 11);  store.print();  try{  store.list[0] += 3;  store.list[0] + 1234;  store.list[1] -= 3;  store.list[1] + 399;  store.list[2] -= 7;  store.list[7] - 1;  }catch(runtime\_error &e){}  store.print();  int cnt;  cout << "Пошук за числом: "; cin >> cnt;  try{  store.search(cnt).render();  }catch(Drug){  cout << "Товар не знайдено!" << endl;  }  char name[64];  cout << "Пошук за назвою: "; cin >> name;  try{  store.search(name).render();  }catch(Drug){  cout << "Товар не знайдено!" << endl;  }  system("pause >> NUL");  } | |
| Файл DrugStore.h | Файл Drug.h |
| #pragma once  #include "Drug.h"  #include <iostream>  using namespace std;  class DrugStore{  public:  DrugStore(void);  ~DrugStore(void);  int getSum(void);  void addItem(char\*);  void addItem(char\*, int);  void addItem(char\*, int, int);  void print(void);  Drug search(int);  Drug search(int, int);  Drug search(char \*);  Drug search(char \*, int);  int current;  Drug list[1024];  }; | #pragma once  class Drug{  public:  Drug(void);  Drug(char\*);  Drug(char\*, int);  Drug(char\*, int, int);  ~Drug(void);  char\* getName(void);  int getCount(void);  int getPrice(void);  void render(void);  Drug operator+=(int);  Drug operator-=(int);  Drug operator+(int);  Drug operator-(int);  private:  char\* name;  int count;  int price;  }; |
| Файл DrogStore.cpp | |
| #include "DrugStore.h"  DrugStore::DrugStore(void){  this->current = 0;  }  DrugStore::~DrugStore(void){}  int DrugStore::getSum(){  int sum = 0;  for(int i=0; i<current; i++){  sum += list[i].getPrice()\*list[i].getCount();  }  return sum;  }  void DrugStore::addItem(char\* n){  Drug temp(n);  list[current++] = temp;  }  void DrugStore::addItem(char\* n, int p){  Drug temp(n, p);  list[current++] = temp;  }  void DrugStore::addItem(char\* n, int p, int c){  Drug temp(n, p, c);  list[current++] = temp;  }  void DrugStore::print(){  for(int i=0; i<current; i++){  list[i].render();  }  int sum = getSum();  int coins = sum % 100;  int uah = (sum-coins)/100;  cout << "Сума: " << uah << "." << coins << endl << endl;  }  Drug DrugStore::search(int val){return search(val, 0);}  Drug DrugStore::search(int val, int offset = 0){  for(int i = offset; i < current; i++){  if(list[i].getCount() == val || list[i].getPrice() == val){  return list[i];  }  }  throw list[0];  return list[0];  }  Drug DrugStore::search(char\* val){return search(val, 0);}  Drug DrugStore::search(char\* val, int offset = 0){  for(int i = offset; i < current; i++){  int cmp = strcmp(list[i].getName(), val);  if(cmp == 0){  return list[i];  }  }  throw list[0];  return list[0];  } | |
| Файл Drug.cpp | |
| #include "Drug.h"  #include "iostream"  using namespace std;  Drug::Drug(void){  this->name = "Undefined";  this->price = 0;  this->count = 0;  }  Drug::Drug(char\* n){  this->name = n;  this->price = 0;  this->count = 0;  }  Drug::Drug(char\* n, int p){  this->name = n;  this->price = p;  this->count = 0;  }  Drug::Drug(char\* n, int p, int c){  this->name = n;  this->price = p;  this->count = c;  }  Drug::~Drug(void){}  char\* Drug::getName(){  return name;  }  int Drug::getCount(){  return count;  }  int Drug::getPrice(){  return price;  }  void Drug::render(void){  int coins = price % 100;  int uah = (price-coins)/100;  cout << name << " | " << uah << "." << coins << " | " << count << "шт." << endl;  }  Drug Drug::operator+=(int n){  try{  this->count += n;  }catch(int e){}  return \*this;  }  Drug Drug::operator-=(int n){  try{  this->count -= n;  }catch(int e){}  return \*this;  }  Drug Drug::operator+(int n){  try{  this->price += n;  }catch(int e){}  return \*this;  }  Drug Drug::operator-(int n){  try{  this->price -= n;  }catch(int e){}  return \*this;  } | |