|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

**Отчет**

|  |  |
| --- | --- |
| **По лабораторной работе №** | 7 |

**Название:**

Простые объекты

**Дисциплина:** Объектно-ориентированное программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-22Б |  |  | И.А. Люляев |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  |  |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2020

Лабораторная работа №7.

**Лабораторная работа 7. Простые объекты**

Описать класс, включающий заданные поля и методы, двумя способами: без конструктора и с конструктором. Протестировать все методы класса. Поля класса должны быть скрытыми (private) или защищенными (protected). Методы не должны содержать операций ввода/вывода, за исключением процедуры, единственной задачей которой является вывод информации об объекте на экран.

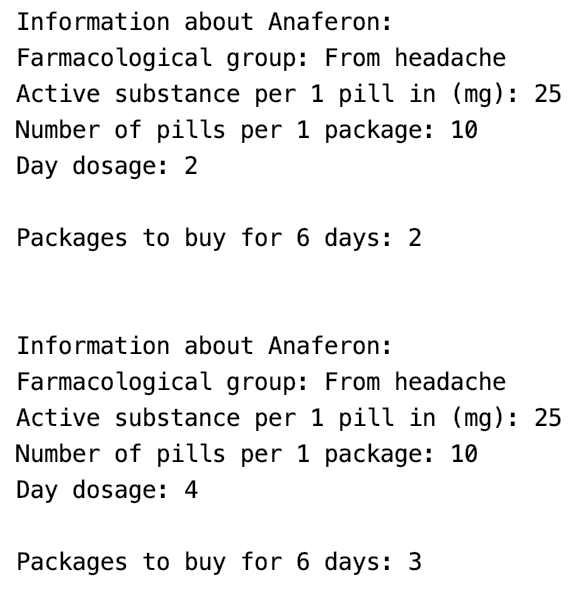
Объект – лекарственный препарат. Поля: название, фармакологическая группа, масса действующего вещества в одной таблетке в миллиграммах, количество таблеток в упаковке, дозировка (мг в сутки). Методы: процедура инициализации, процедура вывода информации о препарате на экран, процедура изменения дозировки, функция вычисления количества упаковок препарата, необходимых для его приема в течение заданного числа дней.

В отчете привести диаграмму разработанных классов и объектную декомпозицию.

Код программы:

#include **<iostream>**#include **<cmath>  
  
class** Drug {  
**public**:  
 Drug(std::string name, std::string farmacologicalGroup, **int** activeSubstance, **int** numberOfPills, **int** dayDosage) {  
 **this**->name = name;  
 **this**->farmacologicalGroup = farmacologicalGroup;  
 **this**->activeSubstance = activeSubstance;  
 **this**->numberOfPills = numberOfPills;  
 **this**->dayDosage = dayDosage;  
 }  
  
 **void** init(std::string name, std::string farmacologicalGroup, **int** activeSubstance, **int** numberOfPills, **int** dayDosage) {  
 **this**->name = name;  
 **this**->farmacologicalGroup = farmacologicalGroup;  
 **this**->activeSubstance = activeSubstance;  
 **this**->numberOfPills = numberOfPills;  
 **this**->dayDosage = dayDosage;  
 }  
  
 **void** printInfo() {  
 std::cout << **"Information about "** << name << **":"** << std::endl  
 << **"Farmacological group: "** << farmacologicalGroup << std::endl  
 << **"Active substance per 1 pill in (mg): "** << activeSubstance << std::endl  
 << **"Number of pills per 1 package: "** << numberOfPills << std::endl  
 << **"Day dosage: "** << dayDosage << std::endl;  
 }  
  
 **void** changeDayDosage(**int** newDayDosage) {  
 dayDosage = newDayDosage;  
 }  
  
 **int** numberOfPackages(**int** days) {  
 **double** a;  
 a = ceil((**double**)(dayDosage\*days) / numberOfPills);  
 **return** a;  
 }  
  
**private**:  
 std::string name;  
 std::string farmacologicalGroup;  
 **int** activeSubstance; *// per 1 pill (mg)* **int** numberOfPills; *// per 1 package* **int** dayDosage;  
};  
  
  
**int** main() {  
 Drug anaferon1(**"Anaferon"**, **"From headache"**, 25, 10, 2);  
 *// anaferon1.init("Anaferon", "From headache", 25, 10, 2);* anaferon1.printInfo();  
 std::cout << std::endl << **"Packages to buy for 6 days: "**<< anaferon1.numberOfPackages(6)  
 << std::endl << std::endl << std::endl;  
  
 anaferon1.changeDayDosage(4);  
  
 anaferon1.printInfo();  
 std::cout << std::endl << **"Packages to buy for 6 days: "**<< anaferon1.numberOfPackages(6) << std::endl;  
  
  
 **return** 0;  
}

Результат:



Вывод: мы научились создавать классы в С++, прописывать в них поля и методы, обращаться к методы созданного объекта. При заданных исходных условиях программа работает успешно.

