**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

**Факультет инфокоммуникационных Сетей и систем (иксс)**

**кафедра программной инженерии и вычислительной техники (пи и вт)**

Дисциплина: «**Разработка Java-приложений управления телекоммуникациями**»

Лабораторная работа №2.

Вариант 3.

Выполнил:

Студент группы ИКПИ-91

Балицкий В.М.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Принял:

Белая Т.И.

\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2021

1. **Цель работы**

Ознакомление с работой с массивами, а также с элементами объектно-ориентированного программирования.

1. **Описание классов**

|  |  |  |  |
| --- | --- | --- | --- |
| **Public class Java\_lab1** | | | |
| **Название метода** | **Входные данные** | **Выходные данные** | **Назначение** |
| public static void main | String[] args | **-** | Точка входа программы |
| static double scanNumber | Scanner sc | double j | Проверка на правильность ввода числа типа Double |
| static int intCheck | Scanner ik | int x | Проверка на правильность ввода числа типа Int |
| public static String readString | String varName | String str; | Проверка на правильность ввода числа типа String |
| static void firstlow | Scanner in1 | - | Выполнение заданий |
| static int firstmid | Scanner in1 | - |
| static double firsthight | Scanner in1 | - |
| static void firstlow2 | Scanner in1 | - |
| static void firstmid2 | - | - |
| static double firsthight2 | Scanner in1 | - |
| static int firstlow3 | Scanner in1 | - |
| static int firstmid3 | Scanner in1 | - |
| static int firstlow4 | Scanner in1 | - |
| static int firstmid4 | Scanner in1 | - |
| static double firstlow5 | Scanner in1 | - |
| static double firstmid5 | Scanner in1 | - |
| static void firsthight5 | - | - |
| static void firsthight6 | - | - |
| static void firsthight7 | - | - |
| static void firsthight8 | - | - |

1. **Цель работы.**

**Задание 1. Вариант 3.**

Составить схему алгоритма и программу решения задачи.



**Решение:**

**static void** firstlow(Scanner in1) {  
 **int**[] x =**new int**[11];  
 **double** z =0;  
 **int** k=0;  
 **for** (**int** i=0; i<11; i++){  
 System.*out*.println("Введите x" + (i+1) + " = ");  
 x[i] = *intCheck*(in1);  
 z += x[i];  
 }  
 z=z/11;  
 **for** (**int** i=0; i<11;i++) {  
 **if** (z<x[i]) {  
 k++;  
 }  
 }  
 System.*out*.println("Кол-во элементов, абсолютное значение которых больше среднего арифметического - " + k );  
}

**Задание 2. Вариант 3.**

Составить схему алгоритма и программу решения задачи



**Решение:**

**static int** firstmid(Scanner in1) {  
 **int**[] x = **new int**[17];  
 **int** z =0;  
 System.*out*.println("Введите двухзначное целое число:");  
 **for**(**int** i =0; i<17; i++) {  
 System.*out*.println("Введите x" + (i+1) + " = ");  
 x[i] = *intCheck*(in1);  
 **if** ((x[i]>99)||(x[i]<10)) {  
 System.*out*.println("Ошибка! Вы ввели не двухзначное целое число! Попробуйте снова:");  
 **return** *firstmid*(in1);  
 }  
 z += x[i]%10;  
 z += x[i]/10%10;  
 }  
 System.*out*.println("Сумма цифр массива - " + z);  
 **return** 0;  
}

**Задание 3. Вариант 3.**

Составить схему алгоритма и программу решения задачи



**Решение:**

**static double** firsthight(Scanner in1) {  
 System.*out*.println("Введите число в двоичной форме записи:");  
 String dv = in1.nextLine();  
 **boolean** x = *validBinary*(dv);  
 **if** (!x) {  
 **return** *firsthight*(in1);  
 }  
 System.*out*.println("Двоичная запись: " + dv + " -> Шестнадцатеричная запись - " + Integer.*toHexString*(Integer.*parseInt*(dv, 2)));  
 **return** 0;  
}

**Задание 4. Вариант 3.**

Составить схему алгоритма и программу решения задачи

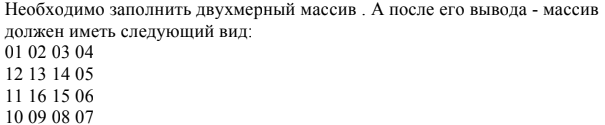


**Решение:**

**static void** firstlow2(Scanner in1){  
 **double**[][] x= **new double**[5][6];  
 **double** sum;  
 **for** (**int** i=0;i<5;i++){  
 sum = 0;  
 **for** (**int** j=0;j<6;j++) {  
 System.*out*.println("Введите x" +"["+(i+1)+"]"+"["+(j+1)+"] =");  
 x[i][j]=*scanNumber*(in1);  
 **if** (x[i][j]>0) {  
 sum += x[i][j];  
 }  
 }  
 sum = sum/6;  
 System.*out*.println("Среднее арифметическое положительных элементов столбца " + (i+1) + " = " + String.*format*("%.3f",sum));  
 }  
}

**Задание 5. Вариант 3.**

Написать блок-схему и программу согласно заданию.



**Решение:**

**static void** firstmid2(){  
 System.*out*.println(«Двухмерный массив:»);  
 String [][] x = {{«01»,»02»,»03»,»04»},  
 {«12»,»13»,»14»,»05»},  
 {«11»,»16»,»15»,»06»},  
 {«10»,»09»,»08»,»07»}};  
 **for** (**int** i=0;i<4;i++){  
 System.*out*.println();  
 **for** (**int** j =0; j<4; j++){  
 System.*out*.print(x[i][j]+»\t»);  
 }  
 }  
 System.*out*.println();  
 System.*out*.println();  
}

}  
}

**Задание 6. Вариант 3.**

Написать блок-схему и программу согласно заданию



**Решение:**

**static double** firsthight2(Scanner in1){  
 System.*out*.println("Введите кол-во строк двухмерного массива - ");  
 **int** n = *intCheck*(in1);  
 **if** (n<=0) {  
 System.*out*.println("Ошибка ввода! Кол-во строк не может быть ранов 0 или быть меньше 0! Введите число снова:");  
 **return** *firsthight2*(in1);  
 }  
 System.*out*.println("Введите кол-во столбцов двухмерного массива - ");  
 **int** k = *intCheck*(in1);  
 **if** (k<=0) {  
 System.*out*.println("Ошибка ввода! Кол-во строк не может быть ранов 0 или быть меньше 0! Введите число снова:");  
 **return** *firsthight2*(in1);  
 }  
 **double** [][] x = **new double**[n][k];  
 **for** (**int** i =0;i<n;i++) {  
 **for** (**int** j = 0; j<k;j++) {  
 System.*out*.println("Введите x" +"["+(i+1)+"]"+"["+(j+1)+"] =");  
 x[i][j] = *scanNumber*(in1);  
 **if** (x[i][j]<0) {  
 x[i][j] = Math.*abs*(x[i][j]);  
 } **else** {  
 x[i][j] = x[i][j] \* (-1);  
 }  
 }  
 }  
 **for** (**int** i =0;i<n;i++) {  
 System.*out*.println();  
 **for** (**int** j = 0; j<k;j++) {  
 System.*out*.print(x[i][j]+"\t");  
 }  
 }  
 System.*out*.println();  
 System.*out*.println();  
 **return** 0;  
}

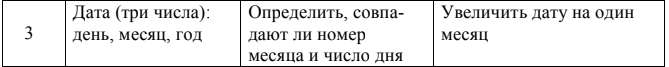
**Задание 7. Вариант 3.**

Создание данных типа “класс”



**static class** ClassTaskOne {  
 **protected int** Price;  
 **protected int** Count;  
  
 ClassTaskOne(){  
 Price = 0;  
 Count = 0;  
 }  
  
 **public void** edit(**int** price\_, **int** count\_) {  
 Price= price\_;  
 Count=count\_;  
 }  
  
 **int** sumPrice (){  
 **return** Price\*Count;  
 }  
  
 **public void** memo() {  
  
 System.*out*.println("Цена товара = " + Price + "\nКол-во товара = " + Count);  
 }  
}  
  
**static int** firstlow3 (Scanner in1){  
 ClassTaskOne tsk = **new** ClassTaskOne();  
 System.*out*.println("Введите пожалуйста цену товара = ");  
 **int** x = *intCheck*(in1);  
 System.*out*.println("Введите пожалуйста кол-во товара = ");  
 **int** y = *intCheck*(in1);  
 **if** ((x<0)||(y<0)) {  
 System.*out*.println("Ошибка! Указанные вам значения меньше 0, попробуйте снова:");  
 **return** *firstlow3*(in1);  
 }  
 tsk.edit(x,y);  
 tsk.memo();  
 System.*out*.println("Общая стоимость товара: " + tsk.sumPrice());  
 **return** 0;  
}

**Задание 8. Вариант 3.**

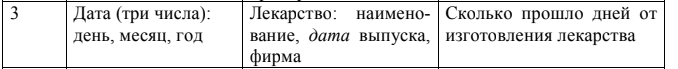


**static class** ClassTaskTwo{  
 **private int** day;  
 **private int** mounth;  
 **private int** year;  
 ClassTaskTwo() {  
 day = 0;  
 mounth = 0;  
 year = 0;  
 }  
 ClassTaskTwo(**int** day\_, **int** mounth\_, **int** year\_) {  
 day = day\_;  
 mounth = mounth\_;  
 year = year\_;  
 }  
 **public void** edit(**int** day\_, **int** mounth\_, **int** year\_) {  
 day = day\_;  
 mounth = mounth\_;  
 year = year\_;  
 }  
 **public void** memo() {  
 System.*out*.println("Дата - " +day+"."+mounth+"."+year);  
  
 }  
 **public void** sovpadenieDayMounth() {  
 **if** (day==mounth) {  
 System.*out*.println("Номер месяца и число дня совпадают!");  
 } **else** {  
 System.*out*.println("Номер месяца и число дня не совпадают!");  
 }  
 }  
 **public void** upLvlMounth () {  
 **if** (mounth==12) {  
 mounth = 1;  
 year++;  
 } **else** {  
 mounth++;  
 }  
 System.*out*.println("Дата увееличина на один месяц!" + "\nДата - " +day+"."+mounth+"."+year);  
 }  
  
 **protected void** finalize(){  
 day = 0;  
 mounth = 0;  
 year = 0;  
 System.*out*.println("Объект уничтожен");  
 }  
}  
  
**static int** firstmid3 (Scanner in1){  
 System.*out*.println("Введите номер дня (1-31) - ");  
 **int** day = *intCheck*(in1);  
 System.*out*.println("Введите номер месяца(1-12) - ");  
 **int** mounth = *intCheck*(in1);  
 System.*out*.println("Введите год (>0) - ");  
 **int** year = *intCheck*(in1);  
 **if** ((mounth>12)||(mounth<1) || (day>31) || (day<1) || (year<1)) {  
 System.*out*.println("Ошибка! Вы вышли за рамки диапазона! Попробуйте снова: ");  
 **return** *firstmid3*(in1);  
 }  
 ClassTaskTwo ctt = **new** ClassTaskTwo() ;  
 ctt.edit(day,mounth,year);  
 ctt.memo();  
 ctt.sovpadenieDayMounth();  
 ctt.upLvlMounth();  
 **return** 0;  
}

**Задание 9. Вариант 3.**



**static class** Taskfree **extends** ClassTaskOne {  
 **protected int** Yearfrom;  
  
 Taskfree() {  
 **super**();  
 Yearfrom = 0;  
 }  
  
 Taskfree(**int** price\_, **int** count\_, **int** yearfrom\_) {  
 Price = price\_;  
 Count = count\_;  
 Yearfrom = yearfrom\_;  
 }  
  
 **public void** edit(**int** price\_, **int** count\_, **int** yearfrom\_) {  
 **super**.edit(price\_, count\_);  
 Yearfrom = yearfrom\_;  
 }  
  
 **public void** memo() {  
 **super**.memo();  
 System.*out*.println("Год выпуска товара - " + Yearfrom);  
 }  
  
 **public void** searchYear() {  
 **int** nowYear = 2021;  
 System.*out*.println("Товару уже - " + (nowYear-Yearfrom) + " лет");  
 }  
}  
  
**static int** firstlow4(Scanner in1) {  
 System.*out*.println("Введите пожалуйста цену товара = ");  
 **int** x = *intCheck*(in1);  
 System.*out*.println("Введите пожалуйста кол-во товара = ");  
 **int** y = *intCheck*(in1);  
 System.*out*.println("Введите пожалуйста год выпуска товара = ");  
 **int** z = *intCheck*(in1);  
 **if** ((x<0)||(y<0) || (z>2021) || (z<1)) {  
 System.*out*.println("Ошибка! Указанные вам значения неверны, попробуйте снова:");  
 **return** *firstlow4*(in1);  
 }  
 Taskfree tsk = **new** Taskfree(x,y,z);  
 tsk.memo();  
 tsk.searchYear();  
 **return** 0;  
}

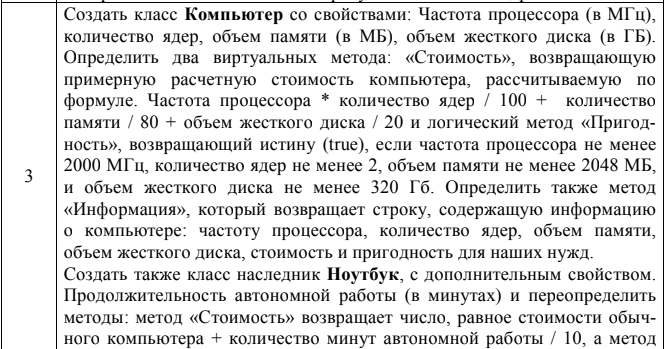
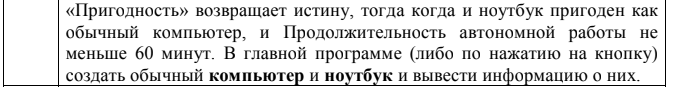
**Задание 10. Вариант 3.** 

**static class** Taskfour **extends** ClassTaskTwo{  
 **protected** String Name,Firma;  
 **protected int** Day1,Mounth1,Year1;  
 Taskfour(){  
 **super**();  
 Name = "NAME";  
 Firma = "FIRMA";  
 Day1 = 0;  
 Mounth1 = 0;  
 Year1 = 0;  
 }  
 Taskfour(String Name\_, String Firma\_, **int** Day1\_, **int** Mounth1\_, **int** Year1\_){  
 **super**();  
 Name = Name\_;  
 Firma= Firma\_;  
 Day1 = Day1\_;  
 Mounth1 = Mounth1\_;  
 Year1 = Year1\_;  
 }  
 **public void** Srok() {  
 LocalDate today = LocalDate.*now*();  
 **int** srok1 = Year1\*365+Mounth1\*30+Day1;  
 **int** srok2 = today.getYear()\*365+today.getMonthValue()\*30+today.getDayOfMonth();  
 System.*out*.println("Дней с момента изготовления лекарства - " + (srok2-srok1));  
 }  
 **public void** memo() {  
 System.*out*.println("Название лекарства - " + Name);  
 System.*out*.println("Название фирмы - " + Firma);  
 System.*out*.println("Дата изготовления лекарства - " + Day1+"."+Mounth1+"."+Year1);  
 }  
}  
  
**static int** firstmid4(Scanner in1) {  
 String z = *readString*("лекарства");  
 System.*out*.println("Введите название фирмы - ");  
 String x = *readString*("фирмы");  
 System.*out*.println("Введите номер дня (1-31) - ");  
 **int** q = *intCheck*(in1);  
 System.*out*.println("Введите номер месяца (1-12) - ");  
 **int** w = *intCheck*(in1);  
 System.*out*.println("Введите номер года (1-2021) - ");  
 **int** e = *intCheck*(in1);  
 **if** ((q<0)||(w<0) || (e>2021) || (e<1)) {  
 System.*out*.println("Ошибка! Указанные вам значения неверны, попробуйте снова:");  
 **return** *firstmid4*(in1);  
 }  
 Taskfour tsk = **new** Taskfour(z,x,q,w,e);  
 tsk.memo();  
 tsk.Srok();  
 **return** 0;  
}

**Задание 11. Вариант 3.**

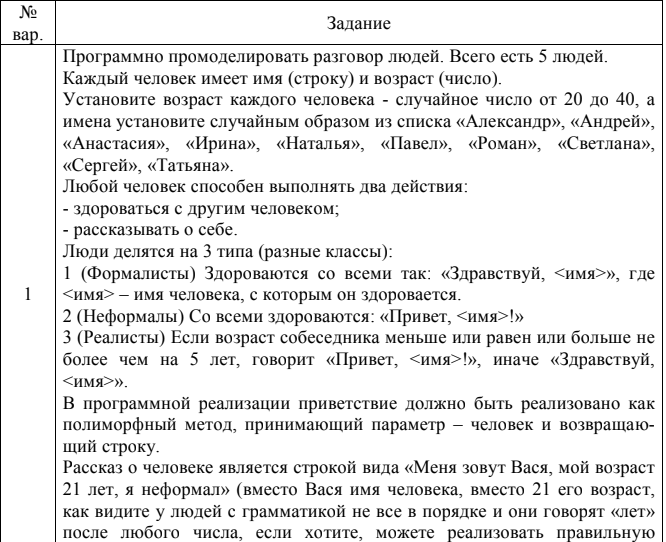


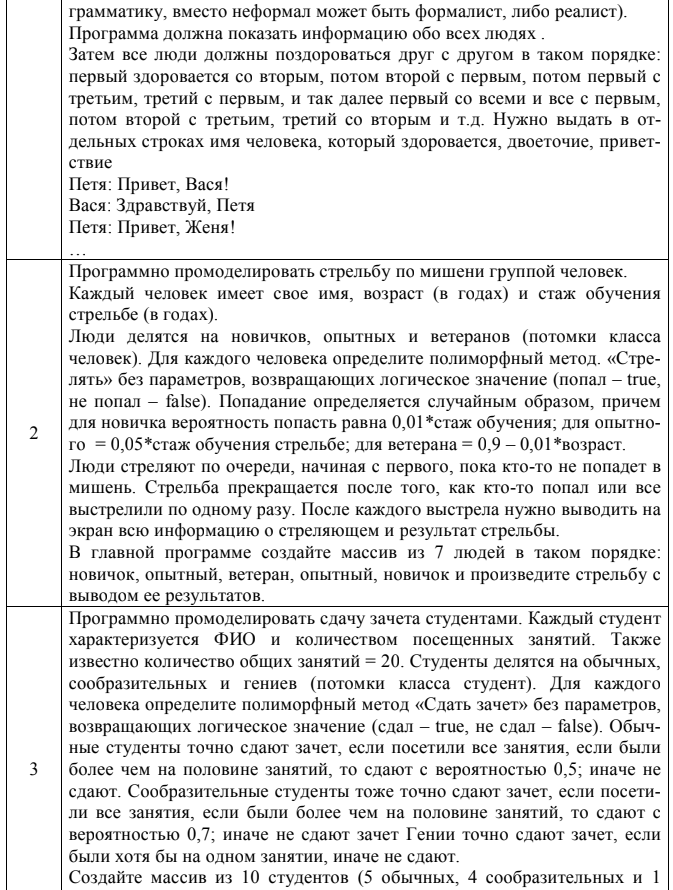
**static class** Product {  
 **protected** String name;  
 **protected int** price;  
 **protected int** count;  
 Product(){  
 name = "Name";  
 price = 0;  
 count = 0;  
 }  
 Product(String name\_, **int** price\_, **int** count\_) {  
 name = name\_;  
 price=price\_;  
 count=count\_;  
 }  
 **public double** Q(){  
 **return** (**float**)(price/count);  
 }  
 **public void** memo(){  
 System.*out*.println("Наименование товара - " + name);  
 System.*out*.println("Цена товара - " + price);  
 System.*out*.println("Кол-во товара - " + count);  
 }  
}  
  
**static class** Product2 **extends** Product{  
 **protected int** yearOfProduction;  
 Product2(){  
 **super**();  
 yearOfProduction = LocalDate.*now*().getYear();  
 }  
 Product2(String name\_, **int** price\_, **int** count\_, **int** yearOfProduction\_){  
 **super**(name\_, price\_, count\_);  
 yearOfProduction = yearOfProduction\_;  
 }  
 **public double** Q(){  
 **return** (**super**.Q()+0.5\*(LocalDate.*now*().getYear()-yearOfProduction));  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Год выпуска товара - " + yearOfProduction);  
 }  
}  
  
**static double** firstlow5(Scanner in1) {  
 String z = *readString*("Продукта");  
 System.*out*.println("Введите цену товара - ");  
 **int** x = *intCheck*(in1);  
 System.*out*.println("Введите кол-во товара - ");  
 **int** c = *intCheck*(in1);  
 **if** ((x<0)||(c<0)) {  
 System.*out*.println("Ошибка! Указанные вам значения меньше 0, попробуйте снова:");  
 **return** *firstlow5*(in1);  
 }  
 Product pr = **new** Product(z,x,c);  
 pr.memo();  
 System.*out*.println("Цена / кол-во товара - " + pr.Q());  
 System.*out*.println("Введите год изготовления товара - ");  
 **int** year = *intCheck*(in1);  
 **if** ((year>2021)||(year<0)) {  
 System.*out*.println("Ошибка! Вы неверно ввели год, попробуйте снова:");  
 **return** *firstlow5*(in1);  
 }  
 Product2 pr2 = **new** Product2(z,x,c,year);  
 pr2.memo();  
 System.*out*.println("(Цена/кол-во товара)+0.5\*(Текущий год - год выпуска товара) - "+pr2.Q());  
 **return** 0;  
}

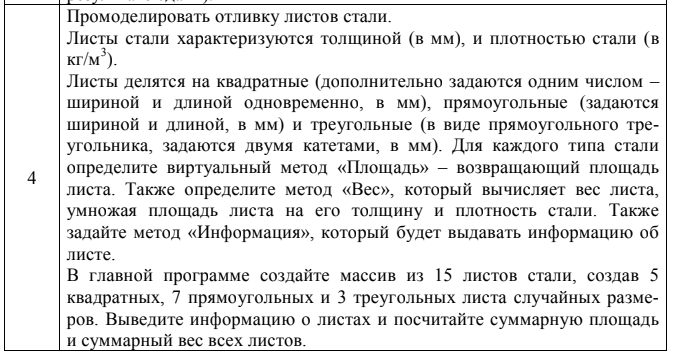
**Задание 12. Вариант 3.**  

**static class** Computer {  
 **protected double** mhz;  
 **protected int** core;  
 **protected int** memmory;  
 **protected int** harddisck;  
 Computer() {  
 mhz =0;  
 core =0;  
 memmory=0;  
 harddisck =0;  
 }  
 Computer(**double** mhz\_,**int** core\_, **int** memmory\_, **int** harddisck\_) {  
 mhz = mhz\_;  
 core=core\_;  
 memmory = memmory\_;  
 harddisck = harddisck\_;  
 }  
 **public double** pricePC() {  
 **return** (**float**)(mhz\*core/100+memmory/80+harddisck/20);  
 }  
 **public boolean** Prig() {  
 **return** (mhz >= 2000) && (core >= 2) && (memmory >= 2048) && (harddisck >= 320);  
 }  
 **public void** memo(){  
 System.*out*.println("Частота процессора - "+mhz);  
 System.*out*.println("Кол-во ядер - " + core);  
 System.*out*.println("Объём оперативной памяти - "+ memmory);  
 System.*out*.println("Объём памяти жёсткого диска - " +harddisck);  
 }  
}  
  
**static class** Laptop **extends** Computer{  
 **protected int** time ;  
 Laptop() {  
 **super**();  
 time = 0;  
 }  
 Laptop(**int** time\_,**double** mhz\_,**int** core\_, **int** memmory\_, **int** harddisck\_ ) {  
 **super**(mhz\_,core\_,memmory\_,harddisck\_);  
 time = time\_;  
 }  
 **public double** pricePC(){  
 **return** (**float**)(**super**.pricePC()+time/10);  
 }  
 **public boolean** Prig() {  
 **return** (**super**.Prig() && (time>=60));  
 }  
 **public void** memo() {  
 **super**.memo();  
 System.*out*.println("Время автономной работы - " + time);  
 }  
}  
  
**static double** firstmid5(Scanner in1) {  
 System.*out*.println("Введите частоту процессора - ");  
 **double** z = *scanNumber*(in1);  
 System.*out*.println("Введите кол-во ядер процессора - ");  
 **int** x = *intCheck*(in1);  
 System.*out*.println("Введите кол-во оперативной памяти - ");  
 **int** c = *intCheck*(in1);  
 System.*out*.println("Введите кол-во памяти жесткого диска - ");  
 **int** v = *intCheck*(in1);  
 **if** ((z<0)||(x<0)||(c<0)||(v<0)) {  
 System.*out*.println("Ошибка! Указанные вам значения меньше 0, попробуйте снова:");  
 **return** *firstmid5*(in1);  
 }  
 Computer cmp = **new** Computer(z,x,c,v);  
 cmp.memo();  
 System.*out*.println("Цена на компьютер ~ "+cmp.pricePC());  
 System.*out*.println("Пригодность - " + cmp.Prig());  
 System.*out*.println("Введите кол-во минут автономной работы ноутбука - ");  
 **int** b = *intCheck*(in1);  
 **if** (b<0) {  
 System.*out*.println("Ошибка! Указанные вам значения меньше 0, попробуйте снова:");  
 **return** *firstmid5*(in1);  
 }  
 Laptop lpt = **new** Laptop(b,z,x,c,v);  
 lpt.memo();  
 System.*out*.println("Цена на ноутбук ~ "+lpt.pricePC());  
 System.*out*.println("Пригодность - " + lpt.Prig());  
 **return** 0;  
}

**Задание 13-16. Вариант 3.**



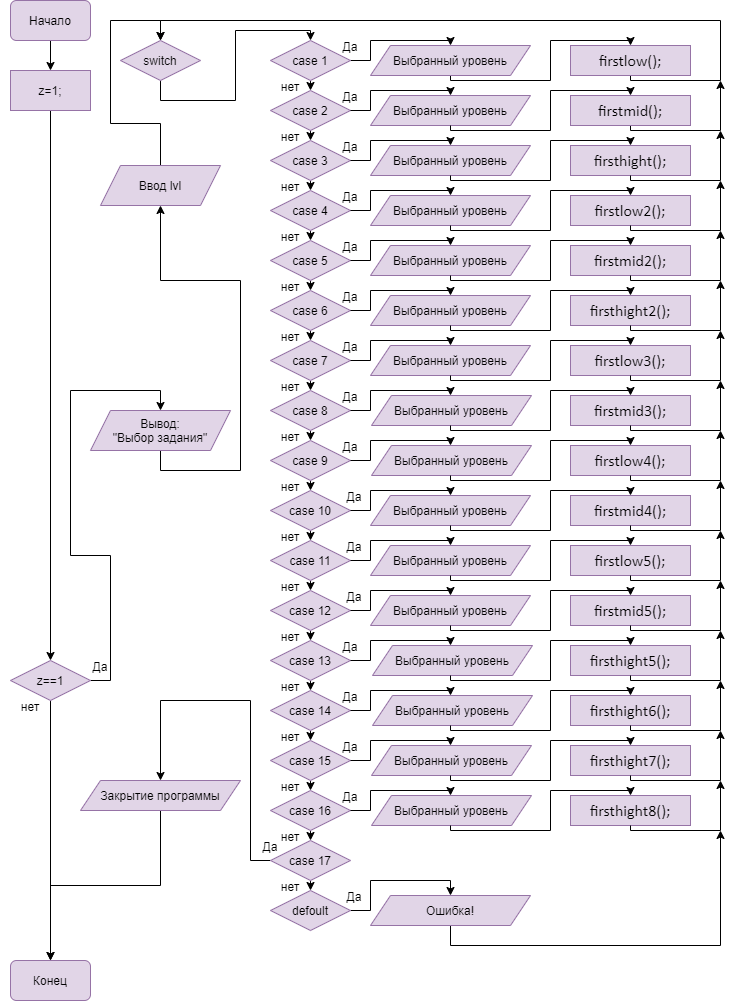


**static abstract class** Person{  
 **protected** String \_name;  
 **protected int** \_age;  
 **protected** String \_type;  
 Person(String name){  
 \_name = name;  
 \_age = (**int**)(Math.*random*()\*20 + 20);  
 }  
 **public abstract** String sayHello(Person pers);  
 **public** String about(){  
 **return** ("Меня зовут " + **this**.\_name + ", мой возраст " +  
 **this**.\_age + " лет, я " + **this**.\_type);  
 }  
}  
**static class** Formalist **extends** Person{  
 Formalist(String name){  
 **super**(name);  
 \_type = "формалист";  
 }  
 **public** String sayHello(Person pers){  
 **return** ("Здравствуй, " + pers.\_name);  
 }  
  
}  
**static class** Informalist **extends** Person{  
 Informalist(String name){  
 **super**(name);  
 \_type = "неформал";  
 }  
 **public** String sayHello(Person pers){  
 **return** ("Привет, " + pers.\_name + "!");  
 }  
}  
**static class** Realist **extends** Person{  
 Realist(String name){  
 **super**(name);  
 \_type = "реалист";  
 }  
 **public** String sayHello(Person pers){  
 **if** (**this**.\_age - pers.\_age >= -5)  
 **return** ("Привет, " + pers.\_name + "!");  
 **else  
 return** ("Здравствуй, " + pers.\_name);  
 }  
}  
  
**static abstract class** Shooter{  
 **protected** String \_name;  
 **protected int** \_age;  
 **protected int** \_experience;  
 Shooter(String name){  
 \_name = name;  
 \_age = (**int**)(Math.*random*()\*20+20);  
 }  
 **public abstract boolean** shoot();  
 **public void** memo(){  
 System.*out*.println("Имя: " + \_name);  
 System.*out*.println("Возраст: " + \_age);  
 System.*out*.println("Опыт: " + \_experience);  
 }  
}  
**static class** Newbie **extends** Shooter{  
 Newbie(String name){  
 **super**(name);  
 \_experience = (**int**)(Math.*random*()\*4);  
 }  
 **public boolean** shoot(){  
 **return** (Math.*random*() < 0.01\***this**.\_experience);  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Тип: Новичок");  
 }  
}  
**static class** Experienced **extends** Shooter{  
 Experienced(String name){  
 **super**(name);  
 \_experience = (**int**)(Math.*random*()\*6+4);  
 }  
 **public boolean** shoot(){  
 **return** (Math.*random*() < 0.05\***this**.\_experience);  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Тип: Опытный");  
 }  
}  
**static class** Veteran **extends** Shooter{  
 Veteran(String name){  
 **super**(name);  
 \_experience = (**int**)(Math.*random*()\*(\_age-10-10)+10);  
 }  
 **public boolean** shoot(){  
 **return** (Math.*random*() < (0.9-0.01\***this**.\_age));  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Тип: Ветеран");  
 }  
}  
  
**static abstract class** Student{  
 **protected** String \_fullName;  
 **protected int** \_visits;  
 Student(String fullName, **int** visits){  
 \_fullName = fullName;  
 \_visits = visits;  
 }  
 **public abstract boolean** passed();  
 **public void** memo(){  
 System.*out*.println("Полное имя : " + \_fullName);  
 System.*out*.println("Посещений: " + \_visits);  
 }  
}  
**static class** DefaultStudent **extends** Student{  
 DefaultStudent(String name, **int** visits){  
 **super**(name, visits);  
 }  
 **public boolean** passed(){  
 **return** \_visits == 20 || (Math.*random*() < 0.5 && \_visits > 10);  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Тип: Обычный студент");  
 }  
}  
**static class** BrightStudent **extends** Student{  
 BrightStudent(String name, **int** visits){  
 **super**(name, visits);  
 }  
 **public boolean** passed(){  
 **return** \_visits == 20 || (Math.*random*() < 0.7 && \_visits > 10);  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Тип: Сообразительный студент");  
 }  
}  
**static class** GeniusStudent **extends** Student{  
 GeniusStudent(String name, **int** visits){  
 **super**(name, visits);  
 }  
 **public boolean** passed(){  
 **return** \_visits > 1;  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Тип: Гениальный студент");  
 }  
}  
  
**static abstract class** Steel{  
 **protected double** \_thickness;  
 **protected double** \_density;  
 Steel(**double** thickness, **double** density){  
 \_thickness = thickness;  
 \_density = density;  
 }  
 **public abstract double** area();  
 **public abstract double** weight();  
 **public void** memo(){  
 System.*out*.println("Толщина: " + String.*format*("%.3f",\_thickness));  
 System.*out*.println("Длина: " + String.*format*("%.3f",\_density));  
 }  
}  
**static class** SquareSheet **extends** Steel{  
 **protected double** \_side;  
 SquareSheet(**double** thickness, **double** density, **double** side){  
 **super**(thickness, density);  
 \_side = side;  
 }  
 **public double** area(){  
 **return** \_side \* \_side;  
 }  
 **public double** weight(){  
 **return this**.area()\*\_density\*\_thickness;  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Тип: Квадратный лист");  
 System.*out*.println("Размеры: " + String.*format*("%.3f",\_side));  
 }  
}  
**static class** RectangleSheet **extends** Steel{  
 **protected double** \_side1;  
 **protected double** \_side2;  
 RectangleSheet(**double** thickness, **double** density, **double** side1, **double** side2){  
 **super**(thickness, density);  
 \_side1 = side1;  
 \_side2 = side2;  
 }  
 **public double** area(){  
 **return** \_side1 \* \_side2;  
 }  
 **public double** weight(){  
 **return this**.area()\*\_density\*\_thickness;  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Тип: Прямоугольный лист");  
 System.*out*.println("Рамзер 1 стороны: " + String.*format*("%.3f",\_side1));  
 System.*out*.println("Рамзер 2 стороны: " + String.*format*("%.3f",\_side2));  
 }  
}  
**static class** TriangleSheet **extends** Steel{  
 **protected double** \_kat1;  
 **protected double** \_kat2;  
 TriangleSheet(**double** thickness, **double** density, **double** kat1, **double** kat2){  
 **super**(thickness, density);  
 \_kat1 = kat1;  
 \_kat2 = kat2;  
 }  
 **public double** area(){  
 **return** \_kat1 \* \_kat2 \* 0.5;  
 }  
 **public double** weight(){  
 **return this**.area()\*\_density\*\_thickness;  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Тип: Треугольный лист");  
 System.*out*.println("Размер 1 катита: " + String.*format*("%.3f",\_kat1));  
 System.*out*.println("Размер 2 катита: " + String.*format*("%.3f",\_kat2));  
 }  
}  
  
**static void** firsthight5(){  
 Person[] array = **new** Person[10];  
 array[0] = **new** Formalist("Александр");  
 array[1] = **new** Formalist("Андрей");  
 array[2] = **new** Formalist("Анастасия");  
 array[3] = **new** Informalist("Ирина");  
 array[4] = **new** Informalist("Наталья");  
 array[5] = **new** Informalist("Павел");  
 array[6] = **new** Realist("Роман");  
 array[7] = **new** Realist("Светлана");  
 array[8] = **new** Realist("Сергей");  
 array[9] = **new** Realist("Татьяна");  
 **for** (Person el : array) {  
 System.*out*.println(el.about());  
 }  
 **for** (**int** i = 0; i < 9; ++i) {  
 **for** (**int** j = i + 1; j < 10; ++j) {  
 System.*out*.println(array[i].\_name + " - " + array[i].sayHello(array[j]));  
 System.*out*.println(array[j].\_name + " - " + array[j].sayHello(array[i]));  
 }  
 }  
}  
  
**static void** firsthight6(){  
 Shooter[] array = **new** Shooter[5];  
 array[0] = **new** Newbie("Стрелок 1");  
 array[1] = **new** Experienced("Стрелок 2");  
 array[2] = **new** Veteran("Стрелок 3");  
 array[3] = **new** Experienced("Стрелок 4");  
 array[4] = **new** Newbie("Стрелок 5");  
 **for** (Shooter el : array){  
 el.memo();  
 **if** (el.shoot()){  
 System.*out*.println("Попадание!");  
 **break**;  
 }  
 **else**{  
 System.*out*.println("Промах!");  
 }  
 }  
}  
  
**static void** firsthight7(){  
 Student[] array = **new** Student[10];  
 array[0] = **new** DefaultStudent("Студент 1", 11);  
 array[1] = **new** DefaultStudent("Студент 2", 4);  
 array[2] = **new** DefaultStudent("Студент 3", 10);  
 array[3] = **new** DefaultStudent("Студент 4", 20);  
 array[4] = **new** DefaultStudent("Студент 5", 9);  
 array[5] = **new** BrightStudent("Студент 6", 20);  
 array[6] = **new** BrightStudent("Студент 7", 11);  
 array[7] = **new** BrightStudent("Студент 8", 15);  
 array[8] = **new** BrightStudent("Студент 9", 5);  
 array[9] = **new** GeniusStudent("Студент 10", 6);  
 **for** (Student el : array){  
 el.memo();  
 System.*out*.println("Результат - " + el.passed());  
 }  
}  
  
**static void** firsthight8(){  
 Steel[] array = **new** Steel[15];  
 **for** (**int** i = 0; i < 5; ++i){  
 **double** tmp = (i + 1);  
 array[i] = **new** SquareSheet(tmp, tmp, tmp);  
 }  
 **for** (**int** i = 0; i < 7; ++i){  
 **double** tmp = (i + 1);  
 array[5+i] = **new** RectangleSheet(tmp,tmp,tmp,tmp+0.1);  
 }  
 **for** (**int** i = 0; i < 3; ++i){  
 **double** tmp = (i + 1);  
 array[12+i] = **new** TriangleSheet(tmp, tmp, tmp,tmp+0.1);  
 }  
 **for** (Steel el : array){  
 el.memo();  
 el.memo();  
 System.*out*.println("Область - " + String.*format*("%.3f",el.area()));  
 System.*out*.println("Вес - " + String.*format*("%.3f",el.weight()));  
 }  
}

**Порядок выполнения операций:**

1. Скобки (()),
2. Умножение, деление \*, /
3. Сложение, вычитание + , -
4. **Блок схемы.**

**public static void** main(String[] args)



1. **Исходный код**

**import** java.io.BufferedReader;  
**import** java.io.IOException;  
**import** java.io.InputStreamReader;  
**import** java.time.LocalDate;  
**import** java.util.Scanner;  
  
**public class** Java\_lab2 {  
  
 **static double** scanNumber(Scanner sc) {  
 **while** (sc.hasNextLine()) {  
 String z = sc.nextLine();  
 **double** j;  
 **try** {  
 j = Double.*parseDouble*(z);  
 } **catch** (NumberFormatException ex) {  
 System.*out*.println("Ошибка ввода! Используйте правильный формат! Введите число снова:");  
 **continue**;  
 }  
 **if** ((Double.*isInfinite*(j)) || (Double.*isNaN*(j))) {  
 System.*out*.println("Ошибка! Вы ввели число за рамками допустимого диапазона! Введите число снова:");  
 } **else** {  
 **return** j;  
 }  
 }  
 **return** 0;  
 }  
  
 **static int** intCheck(Scanner ik) {  
 **while** (ik.hasNextLine()) {  
 String z = ik.nextLine();  
 **int** x;  
 **try** {  
 x = Integer.*parseInt*(z);  
 } **catch** (NumberFormatException ex) {  
 System.*out*.println("Ошибка ввода! Используйте правильный формат! Введите число снова:");  
 **continue**;  
 }  
 **if** (Math.*ceil*(x) == x) {  
 **return** x;  
 } **else** {  
 System.*out*.println("Ошибка! Вы ввели не целое число! Введите число снова:");  
 }  
 }  
 **return** 0;  
 }  
  
 **public static** String readString(String varName){  
 BufferedReader br;  
 String str;  
 br = **new** BufferedReader(**new** InputStreamReader(System.*in*), 32);  
 **do** {  
 System.*out*.println("Введите название " + varName);  
 **try** {  
 str = br.readLine();  
 br = **null**;  
 **break**;  
 } **catch** (IOException err) {  
 err.printStackTrace();  
 }  
 } **while** (**true**);  
 **return** str;  
 }  
  
 **public static boolean** validBinary(String value) {  
 **try** {  
 Long.*parseLong*(value);  
 } **catch** (NumberFormatException ex) {  
 System.*out*.println("Ошибка ввода! Скорее всего вы вели очень большое число, не делайте так пожалуйста! Введите число снова:");  
 **return false**;  
 }  
 **for**(**int** i = 0; i < value.length(); i++) {  
 **int** tempB = value.charAt(i);  
 **if** (tempB == '0' || tempB == '1') {  
 **continue**;  
 }  
 **return false**;  
 }  
 **return true**;  
 }  
  
 **static void** firstlow(Scanner in1) {  
 **int**[] x =**new int**[11];  
 **double** z =0;  
 **int** k=0;  
 **for** (**int** i=0; i<11; i++){  
 System.*out*.println("Введите x" + (i+1) + " = ");  
 x[i] = *intCheck*(in1);  
 z += x[i];  
 }  
 z=z/11;  
 **for** (**int** i=0; i<11;i++) {  
 **if** (z<x[i]) {  
 k++;  
 }  
 }  
 System.*out*.println("Кол-во элементов, абсолютное значение которых больше среднего арифметического - " + k );  
 }  
  
 **static int** firstmid(Scanner in1) {  
 **int**[] x = **new int**[17];  
 **int** z =0;  
 System.*out*.println("Введите двухзначное целое число:");  
 **for**(**int** i =0; i<17; i++) {  
 System.*out*.println("Введите x" + (i+1) + " = ");  
 x[i] = *intCheck*(in1);  
 **if** ((x[i]>99)||(x[i]<10)) {  
 System.*out*.println("Ошибка! Вы ввели не двухзначное целое число! Попробуйте снова:");  
 **return** *firstmid*(in1);  
 }  
 z += x[i]%10;  
 z += x[i]/10%10;  
 }  
 System.*out*.println("Сумма цифр массива - " + z);  
 **return** 0;  
 }  
  
 **static double** firsthight(Scanner in1) {  
 System.*out*.println("Введите число в двоичной форме записи:");  
 String dv = in1.nextLine();  
 **boolean** x = *validBinary*(dv);  
 **if** (!x) {  
 **return** *firsthight*(in1);  
 }  
 System.*out*.println("Двоичная запись: " + dv + " -> Шестнадцатеричная запись - " + Integer.*toHexString*(Integer.*parseInt*(dv, 2)));  
 **return** 0;  
 }  
  
 **static void** firstlow2(Scanner in1){  
 **double**[][] x= **new double**[5][6];  
 **double** sum;  
 **for** (**int** i=0;i<5;i++){  
 sum = 0;  
 **for** (**int** j=0;j<6;j++) {  
 System.*out*.println("Введите x" +"["+(i+1)+"]"+"["+(j+1)+"] =");  
 x[i][j]=*scanNumber*(in1);  
 **if** (x[i][j]>0) {  
 sum += x[i][j];  
 }  
 }  
 sum = sum/6;  
 System.*out*.println("Среднее арифметическое положительных элементов столбца " + (i+1) + " = " + String.*format*("%.3f",sum));  
 }  
 }  
  
 **static void** firstmid2(){  
 System.*out*.println("Двухмерный массив:");  
 String [][] x = {{"01","02","03","04"},  
 {"12","13","14","05"},  
 {"11","16","15","06"},  
 {"10","09","08","07"}};  
 **for** (**int** i=0;i<4;i++){  
 System.*out*.println();  
 **for** (**int** j =0; j<4; j++){  
 System.*out*.print(x[i][j]+"\t");  
 }  
 }  
 System.*out*.println();  
 System.*out*.println();  
 }  
  
 **static double** firsthight2(Scanner in1){  
 System.*out*.println("Введите кол-во строк двухмерного массива - ");  
 **int** n = *intCheck*(in1);  
 **if** (n<=0) {  
 System.*out*.println("Ошибка ввода! Кол-во строк не может быть ранов 0 или быть меньше 0! Введите число снова:");  
 **return** *firsthight2*(in1);  
 }  
 System.*out*.println("Введите кол-во столбцов двухмерного массива - ");  
 **int** k = *intCheck*(in1);  
 **if** (k<=0) {  
 System.*out*.println("Ошибка ввода! Кол-во строк не может быть ранов 0 или быть меньше 0! Введите число снова:");  
 **return** *firsthight2*(in1);  
 }  
 **double** [][] x = **new double**[n][k];  
 **for** (**int** i =0;i<n;i++) {  
 **for** (**int** j = 0; j<k;j++) {  
 System.*out*.println("Введите x" +"["+(i+1)+"]"+"["+(j+1)+"] =");  
 x[i][j] = *scanNumber*(in1);  
 **if** (x[i][j]<0) {  
 x[i][j] = Math.*abs*(x[i][j]);  
 } **else** {  
 x[i][j] = x[i][j] \* (-1);  
 }  
 }  
 }  
 **for** (**int** i =0;i<n;i++) {  
 System.*out*.println();  
 **for** (**int** j = 0; j<k;j++) {  
 System.*out*.print(x[i][j]+"\t");  
 }  
 }  
 System.*out*.println();  
 System.*out*.println();  
 **return** 0;  
 }  
  
 **static class** ClassTaskOne {  
 **protected int** Price;  
 **protected int** Count;  
  
 ClassTaskOne(){  
 Price = 0;  
 Count = 0;  
 }  
  
 **public void** edit(**int** price\_, **int** count\_) {  
 Price= price\_;  
 Count=count\_;  
 }  
  
 **int** sumPrice (){  
 **return** Price\*Count;  
 }  
  
 **public void** memo() {  
  
 System.*out*.println("Цена товара = " + Price + "\nКол-во товара = " + Count);  
 }  
 }  
  
 **static int** firstlow3 (Scanner in1){  
 ClassTaskOne tsk = **new** ClassTaskOne();  
 System.*out*.println("Введите пожалуйста цену товара = ");  
 **int** x = *intCheck*(in1);  
 System.*out*.println("Введите пожалуйста кол-во товара = ");  
 **int** y = *intCheck*(in1);  
 **if** ((x<0)||(y<0)) {  
 System.*out*.println("Ошибка! Указанные вам значения меньше 0, попробуйте снова:");  
 **return** *firstlow3*(in1);  
 }  
 tsk.edit(x,y);  
 tsk.memo();  
 System.*out*.println("Общая стоимость товара: " + tsk.sumPrice());  
 **return** 0;  
 }  
  
 **static class** ClassTaskTwo{  
 **private int** day;  
 **private int** mounth;  
 **private int** year;  
 ClassTaskTwo() {  
 day = 0;  
 mounth = 0;  
 year = 0;  
 }  
 ClassTaskTwo(**int** day\_, **int** mounth\_, **int** year\_) {  
 day = day\_;  
 mounth = mounth\_;  
 year = year\_;  
 }  
 **public void** edit(**int** day\_, **int** mounth\_, **int** year\_) {  
 day = day\_;  
 mounth = mounth\_;  
 year = year\_;  
 }  
 **public void** memo() {  
 System.*out*.println("Дата - " +day+"."+mounth+"."+year);  
  
 }  
 **public void** sovpadenieDayMounth() {  
 **if** (day==mounth) {  
 System.*out*.println("Номер месяца и число дня совпадают!");  
 } **else** {  
 System.*out*.println("Номер месяца и число дня не совпадают!");  
 }  
 }  
 **public void** upLvlMounth () {  
 **if** (mounth==12) {  
 mounth = 1;  
 year++;  
 } **else** {  
 mounth++;  
 }  
 System.*out*.println("Дата увееличина на один месяц!" + "\nДата - " +day+"."+mounth+"."+year);  
 }  
  
 **protected void** finalize(){  
 day = 0;  
 mounth = 0;  
 year = 0;  
 System.*out*.println("Объект уничтожен");  
 }  
 }  
  
 **static int** firstmid3 (Scanner in1){  
 System.*out*.println("Введите номер дня (1-31) - ");  
 **int** day = *intCheck*(in1);  
 System.*out*.println("Введите номер месяца(1-12) - ");  
 **int** mounth = *intCheck*(in1);  
 System.*out*.println("Введите год (>0) - ");  
 **int** year = *intCheck*(in1);  
 **if** ((mounth>12)||(mounth<1) || (day>31) || (day<1) || (year<1)) {  
 System.*out*.println("Ошибка! Вы вышли за рамки диапазона! Попробуйте снова: ");  
 **return** *firstmid3*(in1);  
 }  
 ClassTaskTwo ctt = **new** ClassTaskTwo() ;  
 ctt.edit(day,mounth,year);  
 ctt.memo();  
 ctt.sovpadenieDayMounth();  
 ctt.upLvlMounth();  
 **return** 0;  
 }  
  
 **static class** Taskfree **extends** ClassTaskOne {  
 **protected int** Yearfrom;  
  
 Taskfree() {  
 **super**();  
 Yearfrom = 0;  
 }  
  
 Taskfree(**int** price\_, **int** count\_, **int** yearfrom\_) {  
 Price = price\_;  
 Count = count\_;  
 Yearfrom = yearfrom\_;  
 }  
  
 **public void** edit(**int** price\_, **int** count\_, **int** yearfrom\_) {  
 **super**.edit(price\_, count\_);  
 Yearfrom = yearfrom\_;  
 }  
  
 **public void** memo() {  
 **super**.memo();  
 System.*out*.println("Год выпуска товара - " + Yearfrom);  
 }  
  
 **public void** searchYear() {  
 **int** nowYear = 2021;  
 System.*out*.println("Товару уже - " + (nowYear-Yearfrom) + " лет");  
 }  
 }  
  
 **static int** firstlow4(Scanner in1) {  
 System.*out*.println("Введите пожалуйста цену товара = ");  
 **int** x = *intCheck*(in1);  
 System.*out*.println("Введите пожалуйста кол-во товара = ");  
 **int** y = *intCheck*(in1);  
 System.*out*.println("Введите пожалуйста год выпуска товара = ");  
 **int** z = *intCheck*(in1);  
 **if** ((x<0)||(y<0) || (z>2021) || (z<1)) {  
 System.*out*.println("Ошибка! Указанные вам значения неверны, попробуйте снова:");  
 **return** *firstlow4*(in1);  
 }  
 Taskfree tsk = **new** Taskfree(x,y,z);  
 tsk.memo();  
 tsk.searchYear();  
 **return** 0;  
 }  
  
 **static class** Taskfour **extends** ClassTaskTwo{  
 **protected** String Name,Firma;  
 **protected int** Day1,Mounth1,Year1;  
 Taskfour(){  
 **super**();  
 Name = "NAME";  
 Firma = "FIRMA";  
 Day1 = 0;  
 Mounth1 = 0;  
 Year1 = 0;  
 }  
 Taskfour(String Name\_, String Firma\_, **int** Day1\_, **int** Mounth1\_, **int** Year1\_){  
 **super**();  
 Name = Name\_;  
 Firma= Firma\_;  
 Day1 = Day1\_;  
 Mounth1 = Mounth1\_;  
 Year1 = Year1\_;  
 }  
 **public void** Srok() {  
 LocalDate today = LocalDate.*now*();  
 **int** srok1 = Year1\*365+Mounth1\*30+Day1;  
 **int** srok2 = today.getYear()\*365+today.getMonthValue()\*30+today.getDayOfMonth();  
 System.*out*.println("Дней с момента изготовления лекарства - " + (srok2-srok1));  
 }  
 **public void** memo() {  
 System.*out*.println("Название лекарства - " + Name);  
 System.*out*.println("Название фирмы - " + Firma);  
 System.*out*.println("Дата изготовления лекарства - " + Day1+"."+Mounth1+"."+Year1);  
 }  
 }  
  
 **static int** firstmid4(Scanner in1) {  
 String z = *readString*("лекарства");  
 System.*out*.println("Введите название фирмы - ");  
 String x = *readString*("фирмы");  
 System.*out*.println("Введите номер дня (1-31) - ");  
 **int** q = *intCheck*(in1);  
 System.*out*.println("Введите номер месяца (1-12) - ");  
 **int** w = *intCheck*(in1);  
 System.*out*.println("Введите номер года (1-2021) - ");  
 **int** e = *intCheck*(in1);  
 **if** ((q<0)||(w<0) || (e>2021) || (e<1)) {  
 System.*out*.println("Ошибка! Указанные вам значения неверны, попробуйте снова:");  
 **return** *firstmid4*(in1);  
 }  
 Taskfour tsk = **new** Taskfour(z,x,q,w,e);  
 tsk.memo();  
 tsk.Srok();  
 **return** 0;  
 }  
  
 **static class** Product {  
 **protected** String name;  
 **protected int** price;  
 **protected int** count;  
 Product(){  
 name = "Name";  
 price = 0;  
 count = 0;  
 }  
 Product(String name\_, **int** price\_, **int** count\_) {  
 name = name\_;  
 price=price\_;  
 count=count\_;  
 }  
 **public double** Q(){  
 **return** (**float**)(price/count);  
 }  
 **public void** memo(){  
 System.*out*.println("Наименование товара - " + name);  
 System.*out*.println("Цена товара - " + price);  
 System.*out*.println("Кол-во товара - " + count);  
 }  
 }  
  
 **static class** Product2 **extends** Product{  
 **protected int** yearOfProduction;  
 Product2(){  
 **super**();  
 yearOfProduction = LocalDate.*now*().getYear();  
 }  
 Product2(String name\_, **int** price\_, **int** count\_, **int** yearOfProduction\_){  
 **super**(name\_, price\_, count\_);  
 yearOfProduction = yearOfProduction\_;  
 }  
 **public double** Q(){  
 **return** (**super**.Q()+0.5\*(LocalDate.*now*().getYear()-yearOfProduction));  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Год выпуска товара - " + yearOfProduction);  
 }  
 }  
  
 **static double** firstlow5(Scanner in1) {  
 String z = *readString*("Продукта");  
 System.*out*.println("Введите цену товара - ");  
 **int** x = *intCheck*(in1);  
 System.*out*.println("Введите кол-во товара - ");  
 **int** c = *intCheck*(in1);  
 **if** ((x<0)||(c<0)) {  
 System.*out*.println("Ошибка! Указанные вам значения меньше 0, попробуйте снова:");  
 **return** *firstlow5*(in1);  
 }  
 Product pr = **new** Product(z,x,c);  
 pr.memo();  
 System.*out*.println("Цена / кол-во товара - " + pr.Q());  
 System.*out*.println("Введите год изготовления товара - ");  
 **int** year = *intCheck*(in1);  
 **if** ((year>2021)||(year<0)) {  
 System.*out*.println("Ошибка! Вы неверно ввели год, попробуйте снова:");  
 **return** *firstlow5*(in1);  
 }  
 Product2 pr2 = **new** Product2(z,x,c,year);  
 pr2.memo();  
 System.*out*.println("(Цена/кол-во товара)+0.5\*(Текущий год - год выпуска товара) - "+pr2.Q());  
 **return** 0;  
 }  
  
 **static class** Computer {  
 **protected double** mhz;  
 **protected int** core;  
 **protected int** memmory;  
 **protected int** harddisck;  
 Computer() {  
 mhz =0;  
 core =0;  
 memmory=0;  
 harddisck =0;  
 }  
 Computer(**double** mhz\_,**int** core\_, **int** memmory\_, **int** harddisck\_) {  
 mhz = mhz\_;  
 core=core\_;  
 memmory = memmory\_;  
 harddisck = harddisck\_;  
 }  
 **public double** pricePC() {  
 **return** (**float**)(mhz\*core/100+memmory/80+harddisck/20);  
 }  
 **public boolean** Prig() {  
 **return** (mhz >= 2000) && (core >= 2) && (memmory >= 2048) && (harddisck >= 320);  
 }  
 **public void** memo(){  
 System.*out*.println("Частота процессора - "+mhz);  
 System.*out*.println("Кол-во ядер - " + core);  
 System.*out*.println("Объём оперативной памяти - "+ memmory);  
 System.*out*.println("Объём памяти жёсткого диска - " +harddisck);  
 }  
 }  
  
 **static class** Laptop **extends** Computer{  
 **protected int** time ;  
 Laptop() {  
 **super**();  
 time = 0;  
 }  
 Laptop(**int** time\_,**double** mhz\_,**int** core\_, **int** memmory\_, **int** harddisck\_ ) {  
 **super**(mhz\_,core\_,memmory\_,harddisck\_);  
 time = time\_;  
 }  
 **public double** pricePC(){  
 **return** (**float**)(**super**.pricePC()+time/10);  
 }  
 **public boolean** Prig() {  
 **return** (**super**.Prig() && (time>=60));  
 }  
 **public void** memo() {  
 **super**.memo();  
 System.*out*.println("Время автономной работы - " + time);  
 }  
 }  
  
 **static double** firstmid5(Scanner in1) {  
 System.*out*.println("Введите частоту процессора - ");  
 **double** z = *scanNumber*(in1);  
 System.*out*.println("Введите кол-во ядер процессора - ");  
 **int** x = *intCheck*(in1);  
 System.*out*.println("Введите кол-во оперативной памяти - ");  
 **int** c = *intCheck*(in1);  
 System.*out*.println("Введите кол-во памяти жесткого диска - ");  
 **int** v = *intCheck*(in1);  
 **if** ((z<0)||(x<0)||(c<0)||(v<0)) {  
 System.*out*.println("Ошибка! Указанные вам значения меньше 0, попробуйте снова:");  
 **return** *firstmid5*(in1);  
 }  
 Computer cmp = **new** Computer(z,x,c,v);  
 cmp.memo();  
 System.*out*.println("Цена на компьютер ~ "+cmp.pricePC());  
 System.*out*.println("Пригодность - " + cmp.Prig());  
 System.*out*.println("Введите кол-во минут автономной работы ноутбука - ");  
 **int** b = *intCheck*(in1);  
 **if** (b<0) {  
 System.*out*.println("Ошибка! Указанные вам значения меньше 0, попробуйте снова:");  
 **return** *firstmid5*(in1);  
 }  
 Laptop lpt = **new** Laptop(b,z,x,c,v);  
 lpt.memo();  
 System.*out*.println("Цена на ноутбук ~ "+lpt.pricePC());  
 System.*out*.println("Пригодность - " + lpt.Prig());  
 **return** 0;  
 }  
  
 **static abstract class** Person{  
 **protected** String \_name;  
 **protected int** \_age;  
 **protected** String \_type;  
 Person(String name){  
 \_name = name;  
 \_age = (**int**)(Math.*random*()\*20 + 20);  
 }  
 **public abstract** String sayHello(Person pers);  
 **public** String about(){  
 **return** ("Меня зовут " + **this**.\_name + ", мой возраст " +  
 **this**.\_age + " лет, я " + **this**.\_type);  
 }  
 }  
 **static class** Formalist **extends** Person{  
 Formalist(String name){  
 **super**(name);  
 \_type = "формалист";  
 }  
 **public** String sayHello(Person pers){  
 **return** ("Здравствуй, " + pers.\_name);  
 }  
  
 }  
 **static class** Informalist **extends** Person{  
 Informalist(String name){  
 **super**(name);  
 \_type = "неформал";  
 }  
 **public** String sayHello(Person pers){  
 **return** ("Привет, " + pers.\_name + "!");  
 }  
 }  
 **static class** Realist **extends** Person{  
 Realist(String name){  
 **super**(name);  
 \_type = "реалист";  
 }  
 **public** String sayHello(Person pers){  
 **if** (**this**.\_age - pers.\_age >= -5)  
 **return** ("Привет, " + pers.\_name + "!");  
 **else  
 return** ("Здравствуй, " + pers.\_name);  
 }  
 }  
  
 **static abstract class** Shooter{  
 **protected** String \_name;  
 **protected int** \_age;  
 **protected int** \_experience;  
 Shooter(String name){  
 \_name = name;  
 \_age = (**int**)(Math.*random*()\*20+20);  
 }  
 **public abstract boolean** shoot();  
 **public void** memo(){  
 System.*out*.println("Имя: " + \_name);  
 System.*out*.println("Возраст: " + \_age);  
 System.*out*.println("Опыт: " + \_experience);  
 }  
 }  
 **static class** Newbie **extends** Shooter{  
 Newbie(String name){  
 **super**(name);  
 \_experience = (**int**)(Math.*random*()\*4);  
 }  
 **public boolean** shoot(){  
 **return** (Math.*random*() < 0.01\***this**.\_experience);  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Тип: Новичок");  
 }  
 }  
 **static class** Experienced **extends** Shooter{  
 Experienced(String name){  
 **super**(name);  
 \_experience = (**int**)(Math.*random*()\*6+4);  
 }  
 **public boolean** shoot(){  
 **return** (Math.*random*() < 0.05\***this**.\_experience);  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Тип: Опытный");  
 }  
 }  
 **static class** Veteran **extends** Shooter{  
 Veteran(String name){  
 **super**(name);  
 \_experience = (**int**)(Math.*random*()\*(\_age-10-10)+10);  
 }  
 **public boolean** shoot(){  
 **return** (Math.*random*() < (0.9-0.01\***this**.\_age));  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Тип: Ветеран");  
 }  
 }  
  
 **static abstract class** Student{  
 **protected** String \_fullName;  
 **protected int** \_visits;  
 Student(String fullName, **int** visits){  
 \_fullName = fullName;  
 \_visits = visits;  
 }  
 **public abstract boolean** passed();  
 **public void** memo(){  
 System.*out*.println("Полное имя : " + \_fullName);  
 System.*out*.println("Посещений: " + \_visits);  
 }  
 }  
 **static class** DefaultStudent **extends** Student{  
 DefaultStudent(String name, **int** visits){  
 **super**(name, visits);  
 }  
 **public boolean** passed(){  
 **return** \_visits == 20 || (Math.*random*() < 0.5 && \_visits > 10);  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Тип: Обычный студент");  
 }  
 }  
 **static class** BrightStudent **extends** Student{  
 BrightStudent(String name, **int** visits){  
 **super**(name, visits);  
 }  
 **public boolean** passed(){  
 **return** \_visits == 20 || (Math.*random*() < 0.7 && \_visits > 10);  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Тип: Сообразительный студент");  
 }  
 }  
 **static class** GeniusStudent **extends** Student{  
 GeniusStudent(String name, **int** visits){  
 **super**(name, visits);  
 }  
 **public boolean** passed(){  
 **return** \_visits > 1;  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Тип: Гениальный студент");  
 }  
 }  
  
 **static abstract class** Steel{  
 **protected double** \_thickness;  
 **protected double** \_density;  
 Steel(**double** thickness, **double** density){  
 \_thickness = thickness;  
 \_density = density;  
 }  
 **public abstract double** area();  
 **public abstract double** weight();  
 **public void** memo(){  
 System.*out*.println("Толщина: " + String.*format*("%.3f",\_thickness));  
 System.*out*.println("Длина: " + String.*format*("%.3f",\_density));  
 }  
 }  
 **static class** SquareSheet **extends** Steel{  
 **protected double** \_side;  
 SquareSheet(**double** thickness, **double** density, **double** side){  
 **super**(thickness, density);  
 \_side = side;  
 }  
 **public double** area(){  
 **return** \_side \* \_side;  
 }  
 **public double** weight(){  
 **return this**.area()\*\_density\*\_thickness;  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Тип: Квадратный лист");  
 System.*out*.println("Размеры: " + String.*format*("%.3f",\_side));  
 }  
 }  
 **static class** RectangleSheet **extends** Steel{  
 **protected double** \_side1;  
 **protected double** \_side2;  
 RectangleSheet(**double** thickness, **double** density, **double** side1, **double** side2){  
 **super**(thickness, density);  
 \_side1 = side1;  
 \_side2 = side2;  
 }  
 **public double** area(){  
 **return** \_side1 \* \_side2;  
 }  
 **public double** weight(){  
 **return this**.area()\*\_density\*\_thickness;  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Тип: Прямоугольный лист");  
 System.*out*.println("Рамзер 1 стороны: " + String.*format*("%.3f",\_side1));  
 System.*out*.println("Рамзер 2 стороны: " + String.*format*("%.3f",\_side2));  
 }  
 }  
 **static class** TriangleSheet **extends** Steel{  
 **protected double** \_kat1;  
 **protected double** \_kat2;  
 TriangleSheet(**double** thickness, **double** density, **double** kat1, **double** kat2){  
 **super**(thickness, density);  
 \_kat1 = kat1;  
 \_kat2 = kat2;  
 }  
 **public double** area(){  
 **return** \_kat1 \* \_kat2 \* 0.5;  
 }  
 **public double** weight(){  
 **return this**.area()\*\_density\*\_thickness;  
 }  
 **public void** memo(){  
 **super**.memo();  
 System.*out*.println("Тип: Треугольный лист");  
 System.*out*.println("Размер 1 катита: " + String.*format*("%.3f",\_kat1));  
 System.*out*.println("Размер 2 катита: " + String.*format*("%.3f",\_kat2));  
 }  
 }  
  
 **static void** firsthight5(){  
 Person[] array = **new** Person[10];  
 array[0] = **new** Formalist("Александр");  
 array[1] = **new** Formalist("Андрей");  
 array[2] = **new** Formalist("Анастасия");  
 array[3] = **new** Informalist("Ирина");  
 array[4] = **new** Informalist("Наталья");  
 array[5] = **new** Informalist("Павел");  
 array[6] = **new** Realist("Роман");  
 array[7] = **new** Realist("Светлана");  
 array[8] = **new** Realist("Сергей");  
 array[9] = **new** Realist("Татьяна");  
 **for** (Person el : array) {  
 System.*out*.println(el.about());  
 }  
 **for** (**int** i = 0; i < 9; ++i) {  
 **for** (**int** j = i + 1; j < 10; ++j) {  
 System.*out*.println(array[i].\_name + " - " + array[i].sayHello(array[j]));  
 System.*out*.println(array[j].\_name + " - " + array[j].sayHello(array[i]));  
 }  
 }  
 }  
  
 **static void** firsthight6(){  
 Shooter[] array = **new** Shooter[5];  
 array[0] = **new** Newbie("Стрелок 1");  
 array[1] = **new** Experienced("Стрелок 2");  
 array[2] = **new** Veteran("Стрелок 3");  
 array[3] = **new** Experienced("Стрелок 4");  
 array[4] = **new** Newbie("Стрелок 5");  
 **for** (Shooter el : array){  
 el.memo();  
 **if** (el.shoot()){  
 System.*out*.println("Попадание!");  
 **break**;  
 }  
 **else**{  
 System.*out*.println("Промах!");  
 }  
 }  
 }  
  
 **static void** firsthight7(){  
 Student[] array = **new** Student[10];  
 array[0] = **new** DefaultStudent("Студент 1", 11);  
 array[1] = **new** DefaultStudent("Студент 2", 4);  
 array[2] = **new** DefaultStudent("Студент 3", 10);  
 array[3] = **new** DefaultStudent("Студент 4", 20);  
 array[4] = **new** DefaultStudent("Студент 5", 9);  
 array[5] = **new** BrightStudent("Студент 6", 20);  
 array[6] = **new** BrightStudent("Студент 7", 11);  
 array[7] = **new** BrightStudent("Студент 8", 15);  
 array[8] = **new** BrightStudent("Студент 9", 5);  
 array[9] = **new** GeniusStudent("Студент 10", 6);  
 **for** (Student el : array){  
 el.memo();  
 System.*out*.println("Результат - " + el.passed());  
 }  
 }  
  
 **static void** firsthight8(){  
 Steel[] array = **new** Steel[15];  
 **for** (**int** i = 0; i < 5; ++i){  
 **double** tmp = (i + 1);  
 array[i] = **new** SquareSheet(tmp, tmp, tmp);  
 }  
 **for** (**int** i = 0; i < 7; ++i){  
 **double** tmp = (i + 1);  
 array[5+i] = **new** RectangleSheet(tmp,tmp,tmp,tmp+0.1);  
 }  
 **for** (**int** i = 0; i < 3; ++i){  
 **double** tmp = (i + 1);  
 array[12+i] = **new** TriangleSheet(tmp, tmp, tmp,tmp+0.1);  
 }  
 **for** (Steel el : array){  
 el.memo();  
 el.memo();  
 System.*out*.println("Область - " + String.*format*("%.3f",el.area()));  
 System.*out*.println("Вес - " + String.*format*("%.3f",el.weight()));  
 }  
 }  
  
 **public static void** main(String[] args) {  
 Scanner in = **new** Scanner(System.*in*);  
 System.*out*.println("Программа успешно запустилась!");  
 **int** z=1;  
 **while** (z==1) {  
 System.*out*.println("Укажите уровень сложности, где 1 - Базовый(1задание), 2 - средний(1задание), 3 - высокий(1задание), 4 - базвоый(2задание)," +  
 "5 - средний(2задание), 6 - высокий(2задание), 7 - закрыть программу.");  
 **int** lvl = *intCheck*(in);  
 **switch** (lvl) {  
 **case** 1:  
 System.*out*.println("Вы выбрали базовый уровень (одномерные массивы)");  
 *firstlow*(in);  
 **break**;  
 **case** 2:  
 System.*out*.println("Вы выбрали средний уровень (одномерные массивы)");  
 *firstmid*(in);  
 **break**;  
 **case** 3:  
 System.*out*.println("Вы выбрали высокий уровень (одномерные массивы)");  
 *firsthight*(in);  
 **break**;  
 **case** 4:  
 System.*out*.println("Вы выбрали базовый уровень (двухмерные массивы)");  
 *firstlow2*(in);  
 **break**;  
 **case** 5:  
 System.*out*.println("Вы выбрали средний уровень (двухмерные массивы)");  
 *firstmid2*();  
 **break**;  
 **case** 6:  
 System.*out*.println("Вы выбрали высокий уровень (двухмерные массивы) ");  
 *firsthight2*(in);  
 **break**;  
 **case** 7:  
 System.*out*.println("Вы выбрали базовый уровень (Создание данных типа класс)");  
 *firstlow3*(in);  
 **break**;  
 **case** 8:  
 System.*out*.println("Вы выбрали средний уровень (Создание данных типа класс)");  
 *firstmid3*(in);  
 **break**;  
 **case** 9:  
 System.*out*.println("Вы выбрали базовый уровень (Наследование классов)");  
 *firstlow4*(in);  
 **break**;  
 **case** 10:  
 System.*out*.println("Вы выбрали средний уровень (Наследование классов)");  
 *firstmid4*(in);  
 **break**;  
 **case** 11:  
 System.*out*.println("Вы выбрали базовый уровень (Полиморфизмы методов клаасса)");  
 *firstlow5*(in);  
 **break**;  
 **case** 12:  
 System.*out*.println("Вы выбрали средний уровень (Полиморфизмы методов клаасса)");  
 *firstmid5*(in);  
 **break**;  
 **case** 13:  
 System.*out*.println("Вы выбрали высокий уровень вариант 1 (Полиморфизмы методов клаасса)");  
 *firsthight5*();  
 **break**;  
 **case** 14:  
 System.*out*.println("Вы выбрали высокий уровень вариант 2 (Полиморфизмы методов клаасса)");  
 *firsthight6*();  
 **break**;  
 **case** 15:  
 System.*out*.println("Вы выбрали высокий уровень вариант 3 (Полиморфизмы методов клаасса)");  
 *firsthight7*();  
 **break**;  
 **case** 16:  
 System.*out*.println("Вы выбрали высокий уровень вариант 4 (Полиморфизмы методов клаасса)");  
 *firsthight8*();  
 **break**;  
 **case** 17:  
 System.*out*.println("Закрытие программы");  
 z = 0;  
 **break**;  
 **default**:  
 System.*out*.println("Вы указали неверное число, попробуйте снова:");  
 **break**;  
 }  
 }  
 }  
}

**И. Вывод.**

Программа написана на языке программирования Java. Использовалась среда программирования IntelliJ IDEA Community Edition 2021 64-bit версия для Windows. Данная программа должна решать поставленные задачи находить ошибки в случае некорректного ввода. Ввод в программу осуществляется с консоли в среде IntelliJ IDEA Community Edition 2021, вывод соответственно можно увидеть сразу после ввода всех данных в консоли.