****

**Колледж космического машиностроения и технологий**

**ОТЧЕТ**

**по экзамену ПМ.03**

**специальность** **09.02.03 «Программирование в компьютерных системах»**

Студент 4 курса группы П3-16

Поддубный Даниил Юрьевич

Итоговая оценка по экзамену \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Типы данных VBA**

VBA использует тип Date для хранения дат и времени.

При работе с этим типом данных следует иметь ввиду, что VBA-типы Date не являются такими же типами, как в рабочих листах Excel, хотя во многом и схожи с ними. Например, базовой датой для VBA-типа Date является 30 декабря 1899 года, а в Excel - 1 января 1900 года.

VBA-тип Date является типом последовательных дат. VBA использует отрицательные числа для представления дат ранее базовой даты (30 декабря 1899), и положительные - для дат после базовой. Число 0 представляет саму дату 30.12.1899.

В значении последовательной даты целая часть - это общее число дней от базовой даты. Дробная часть (цифра справа от десятичного знака) - эти цифры обозначают время дня как часть дня. Один час - это 1/24 дня, одна минута - 1/1440 дня, секунда - 1/86400 дня.

Можно вычитать одну дату из другой, добавлять к дате или вычитать числа для изменения ее значения. В VBA имеется несколько встроенных процедур для отдельного извлечения года, месяца, дня, часов, минут и секунд переменной типа Date.

Числа

VBA имеет шесть различных численных типов данных: Byte, Integer, Long,Single, Double, Currency.

Они используются для хранения чисел в различных форматах, в зависимости от конкретного типа.

Целые числа

Integer - это целое число без дробной части (целые числа никогда не содержат десятичного знака, даже если десятичная часть равна нулю).

VBA предоставляет три типа целых данных: Byte, Integer, Long.

Byte - наименьший из трех типов целых данных, предназначен для хранения числе от 0 до 255. Этот тип обычно используют для хранения двоичных данных.

Integer использует диапазон целых чисел от -32768 до 32767.

Long использует диапазон целых чисел от -2147483648 до 2147483647.

Byte, Integer, Long типы требуют меньше памяти для хранения чисел, чем другие численные типы данных VBA, а математические операции и операции сравнения над числами указанных типов быстрее, чем эти же операции для численных типов данных с плавающей точкой.

VBA автоматически преобразует данные типов Byte, Integer, Long в текст, когда они выводятся на экран, используя такие процедуры, как MsgBox.

Числа с плавающей точкой

Числа с плавающей точкой могут иметь любое число цифр до или после десятичной точки (в пределах границ конкретного типа данных). Иногда их называют действительными числами. Этот тип данных используют тогда, когда требуется хранить числа, имеющие дробную часть.

VBA имеет два типа данных с плавающей точкой: Single, Double.

Single используется для хранения:

отрицательных чисел от -3,402823\*1038 до -1,401298\*10-45

и положительных от 1,401298\*10-45 до 3,402823\*1038.

Числа, хранимые с использованием типа Single, называются числами одинарной точности.

Double используется для хранения:

отрицательных чисел от -1,79769313486232\*10308 до -4,94065645841247\*10-324

и положительных от 4,94065645841247\*10-324 до 1,79769313486232\*10308.

Числа, хранимые с использованием типа Double, называются числами двойной точности.

Следует иметь ввиду, что операции, выполняемые над числами с плавающей точкой, немного медленнее подобных операций над другими численными типами данных. Кроме того, числа, хранимые как типы данных с плавающей точкой, могут быть подвержены ошибкам округления. Если число с плавающей точкой очень большое или очень малое, VBA отображает его в экспоненциальном представлении.

В экспоненциальном представлении значения записываются без начальных и конечных нулей, и слева от десятичного знака имеется только одна цифра. Число умножается на 10 в некоторой степени, чтобы показать, где на самом деле находится десятичный знак. Следует помнить, что отрицательная степень приводит в результате к меньшему числу, а положительная - к большему. Вместо надстрочного символа показателя степени используется буква Е:

4.1Е3=4100

2.01Е-2=0.0201

Тип данных Currency

Это число с фиксированной точкой, т.е., десятичная точка всегда находится в одном и том же месте - справа от точки всегда имеются четыре цифры. Этот тип данных используется при денежных вычислениях, когда требуется высокая точность.

Currency хранит числа в диапазоне от -922337203685477,5808 до 922337203685477,5807.

Текстовые строки

Любые текстовые данные, сохраняемые в VBA, называются строками. Для их хранения предназначен тип данных String.

Строка может содержать текстовые символы любых типов: буквы, цифры, знаки пунктуации, разделительные символы и пр.

Строки всегда заключаются в двойные кавычки.

Существует две категории строк: строки переменной длины и строки фиксированной длины. По умолчанию в VBA все строки переменной длины.

Следует сказать, что тип данных String является очень важным типом, т.к., как правило, информация, вводимая пользователем - это строковые данные. Кроме того, поскольку на экране монитора отображается только текст, то все другие типы данных должны быть преобразованы в строковые данные перед выводом на экран.

Логические значения

VBA-программа принимает то или иное "решение", основываясь на различных условиях. Любое условие может принимать два значения: True (Истина) или False (Ложь). Логические значения True и False называются булевскими значениями, а тип данных - Boolean.

Булевские значения получают как результат операции сравнения. Операции сравнения имеет место при сравнении одного с другим, например, при сравнении двух чисел для определения, которое из них больше.

Тип данных Variant

Это особый тип данных, который может сохранять любые типы, за исключением типа Object. VBA использует этот тип данных для всех переменных, если только они не объявлены явно.

Данные типа Variant принимают характеристики определенного типа, который они сохраняют в данный момент. Например, если данные типа Variant содержат строковые данные, Variant принимает характеристики типа String. VBA использует для данных типа Variant наиболее компактное представление, возможное для конкретных значений, находящихся в данных.

Несмотря на то, что типы Variant удобны и избавляют от некоторой части работы при написании кода, они требуют большего объема памяти, чем любой другой тип данных, за исключением больших строк. Кроме того, математические операции и операции сравнения над данными типа Variant выполняются медленнее, чем подобные операции над данными любого другого типа. Следует избегать использования переменных Variant.

1. **Разработать класс Student, хранящий информацию о студенте. Определить нужные поля. Включить в класс методы set (…), get (…), show (…). Определить другие необходимые методы. Создать массив объектов. Вывести: по каждой группе - список студентов, получающих стипенди.**

#include<stdio.h>

#include<string.h>

#include<conio.h>

#include<clocale>

class Student

{

public:

char family[20], name[20], surname[20], addres[20], phone[20], faculty[30];

int course, dd, mm, yyyy, step;

void set()

{

printf("Ввод данных о студенте:\n");

printf(" Фамилия: ");

scanf("%s",family);

printf(" Имя: ");

scanf("%s",name);

printf(" Отчество: ");

scanf("%s",surname);

printf(" Дата рождения (ДД ММ ГГГГ): ");

scanf("%d%d%d",&dd, &mm, &yyyy);

printf(" Адрес: ");

scanf("%s",addres);

printf(" Нормер телефона: ");

scanf("%s",phone);

printf(" Факультет: ");

scanf("%s",faculty);

printf(" Курс: ");

scanf("%d",&course);

printf(" Размер стипендии: ");

scanf("%d",&step);

printf("\n");

}

bool get(int y)

{

if (y < yyyy)

return true;

else

return false;

}

void show(char \*fac)

{

if(strcmp(fac,faculty)==0)

{

printf("\nФИО: %s %s %s\n",family,name,surname);

printf("Дата рождения: %d/%d/%d\n",dd,mm,yyyy);

printf("Адрес: %s\n",addres);

printf("Номер телефона: %s\n",phone);

printf("Факультет: %s\n",faculty);

printf("Курс: %d\n",course);

printf("Стипендия: %d\n",step);

}

}

void show()

{

printf("\nФИО: %s %s %s\n",family,name,surname);

printf("Дата рождения: %d/%d/%d\n",dd,mm,yyyy);

printf("Адрес: %s\n",addres);

printf("Номер телефона: %s\n",phone);

printf("Факультет: %s\n",faculty);

printf("Курс: %d\n",course);

printf("Стипендия: %d\n",step);

}

};

int main()

{

setlocale(LC\_ALL, "rus");

int n,step;

printf("Введите количество записей: ");

scanf("%d",&n);

Student \* A;

A = new Student[n];

printf("----------------------------------------\n");

printf(" Ввод данных:\n");

for(int i=0;i<n;i++)

A[i].set();

printf("------------------------------------------\n");

printf("Список студентов получающих стипендию:\n");

for(int i=0;i<n;i++){

if (A[i].step > 0)

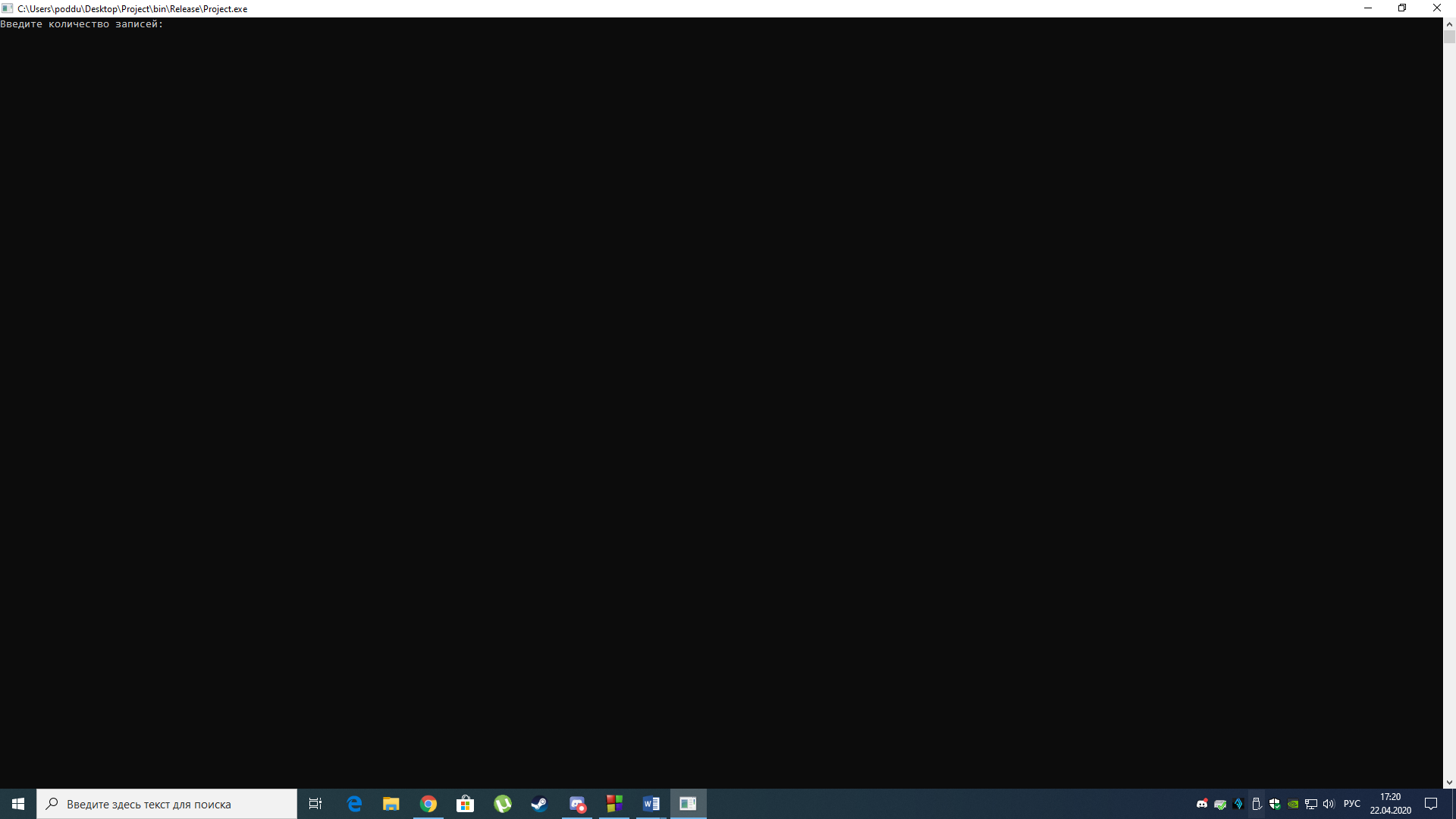
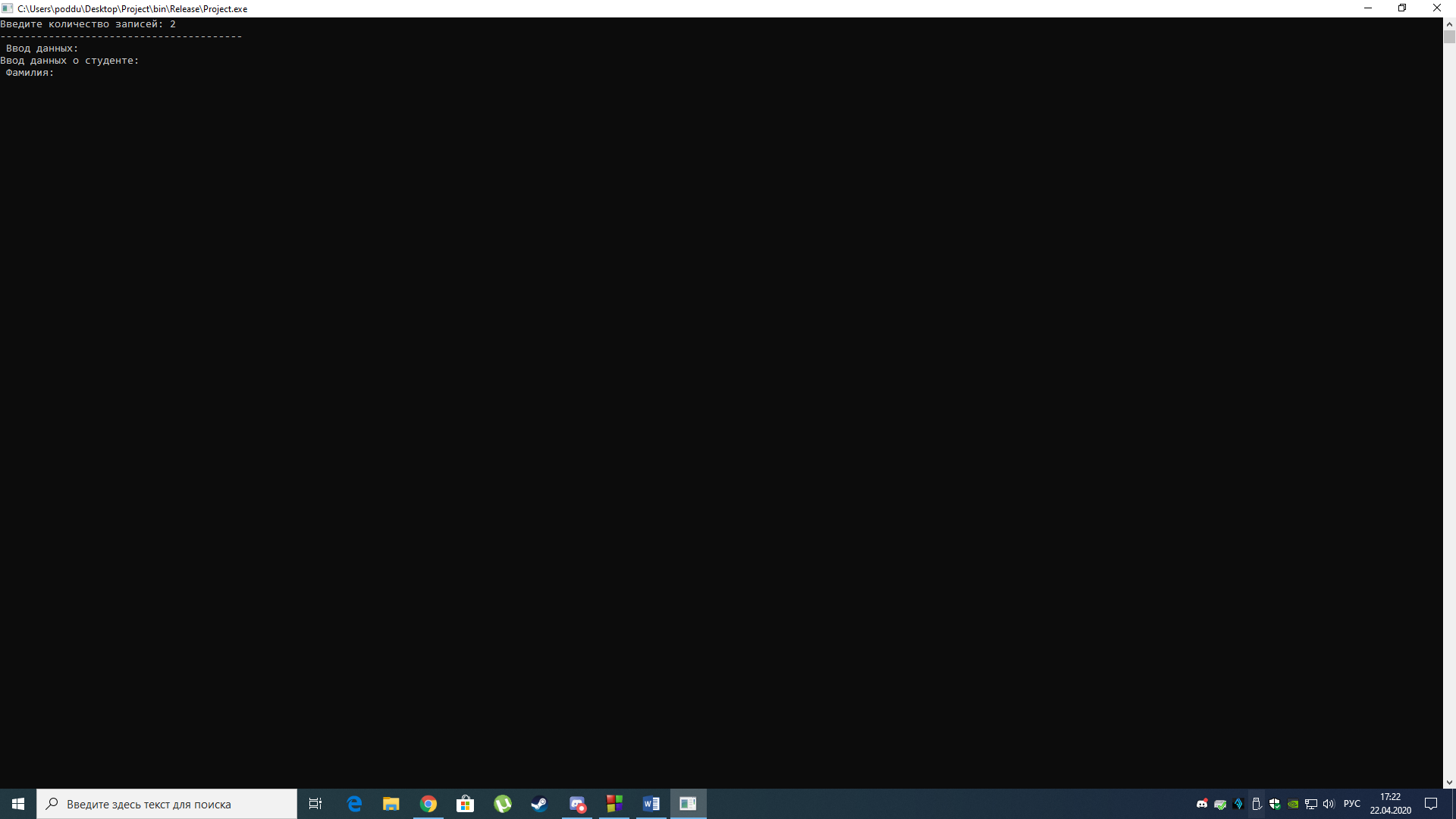
A[i].show();

}

printf("------------------------------------------\n");

}

1. **Руководство оператора по работе с данной программой.**

рис.1 Вводим кол-во записей студентов

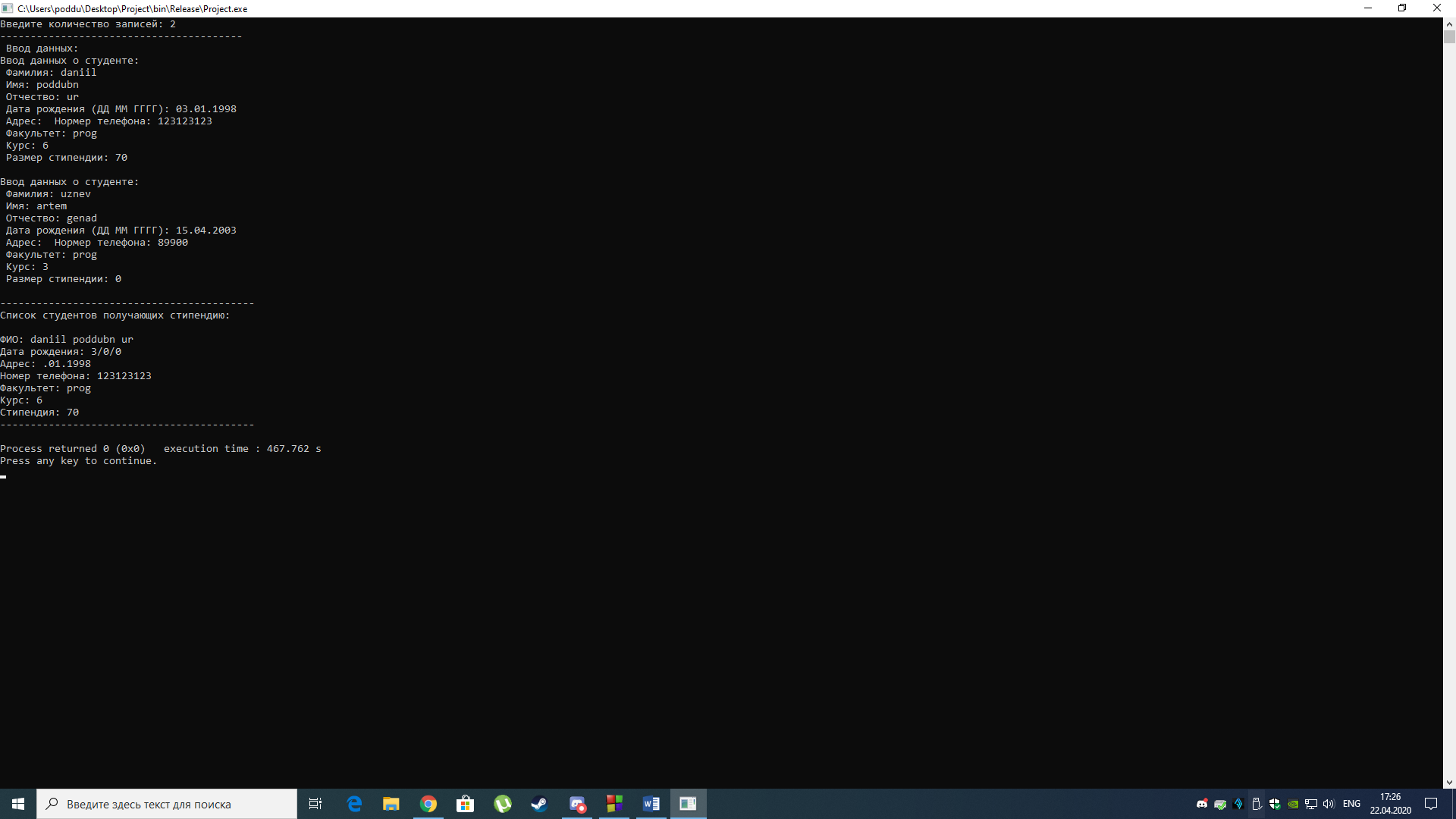
 рис.2 Вводим данные о студентах

рис.3 Выводится информация о студентах получающих стипендию