**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 1**

по дисциплине «Введение в облачные вычисления»

на тему: «Платформа *Windows Azure*»

Выполнил: студент гр. ИТП-31

Коркуц С. И.

Принял: преподаватель

Гуменников Е.Д.

Гомель 2020

**Цель работы:** освоить средства *Windows Azure Virtual Machines*.

**Задание:** Согласно варианту (рисунок 1), разработать программу, реализующую указанный функционал, и опубликовать её на виртуальной машине указанной операционной системы. Для подключения к виртуальной машине использовать SSH.

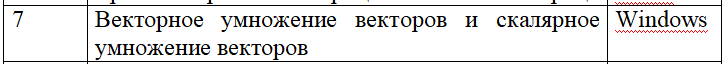


Рисунок 1 – Вариант задания

**Ход работы:**

Был установлен сервер *OpenSSH* в *Windows* через *PowerShell*.

Add-WindowsCapability -Online -Name OpenSSH.Server\*

Далее было проверено установлен ли *OpenSSH*.

Get-WindowsCapability -Online | ? Name -like 'OpenSSH.Ser\*'



Далее через *Filezilla* был передан исполняемый файл программы (рисунок 2).

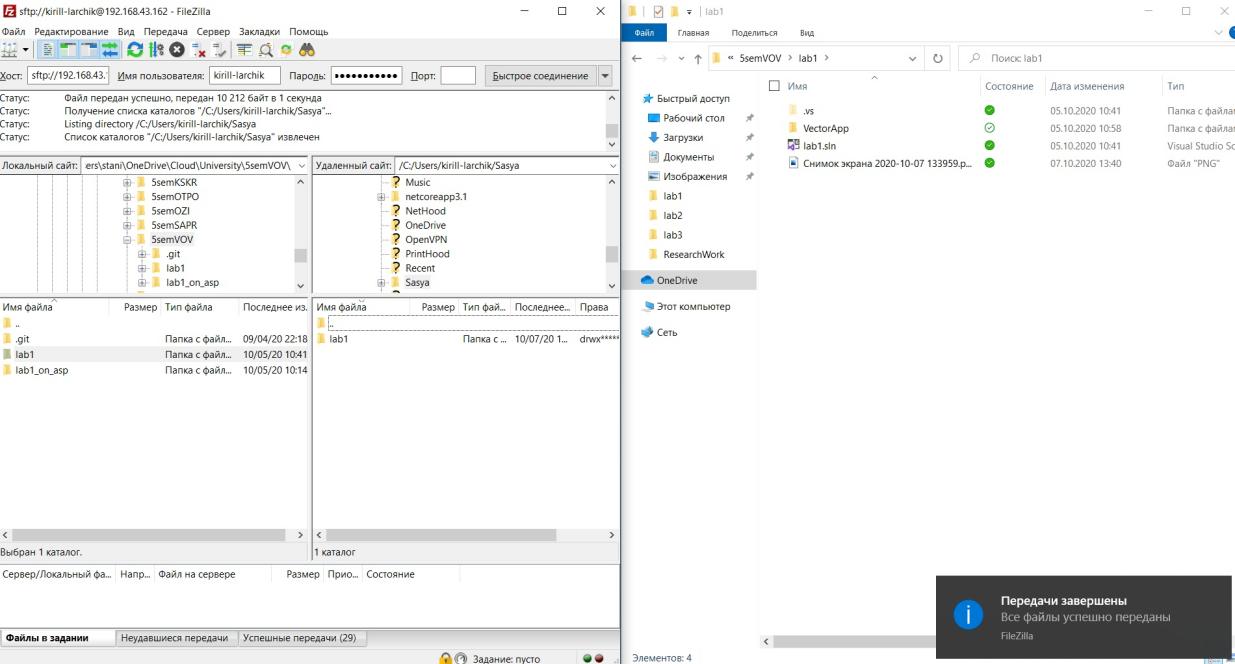


Рисунок 2 – Успешная передача исполняемого файла программы на виртуальную машину

После было установлено *ssh* подключение (рисунок 3).

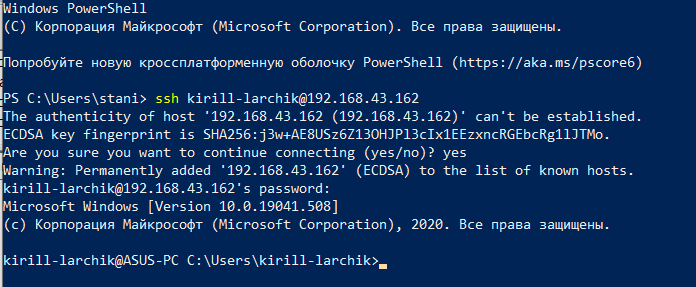


Рисунок 3 – Установленное *ssh* подключение

Следующим шагом был переход в нужную директорию и запуск исполняемого файла. Результат этого на рисунке 4.

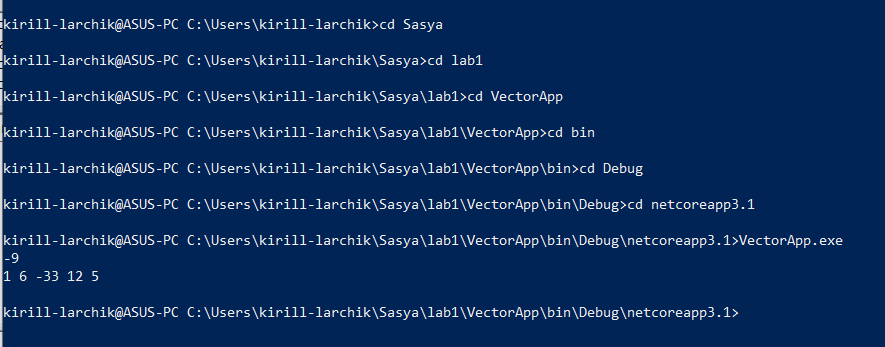


Рисунок 4 – Результат запуска исполняемого файла

**Вывод:** В результате лабораторной работы было создано программное обеспечение, выполняющее действия, заданные вариантом. Также были получены навыки по работе с удалёнными серверами.

**ПРИЛОЖЕНИЕ А**

**Листинг созданных классов**

using System;

using System.Collections.Generic;

using System.Text;

namespace VectorApp

{

    class Vector

    {

        double[] values;

        public int Length

        {

            get

            {

                return values.Length;

            }

        }

        public Vector(double[] values)

        {

            this.values = values;

        }

        public double dot(Vector vector)

        {

            if(vector.Length != this.Length)

            {

                throw new Exception("Длины векторов не совпадают");

            }

            double accum = 0;

            for(int i = 0; i < this.Length; i++)

            {

                accum += vector[i] \* this[i];

            }

            return accum;

        }

        public Vector multiply(Vector vector)

        {

            if (vector.Length != this.Length)

            {

                throw new Exception("Длины векторов не совпадают");

            }

            double[] arr = new double[this.Length];

            for(int i = 0; i < this.Length; i++)

            {

                arr[i] = vector[i] \* this[i];

            }

            return new Vector(arr);

        }

        public double this[int index]

        {

            get

            {

                return values[index];

            }

        }

        public override string ToString()

        {

            StringBuilder builder = new StringBuilder();

            for(int i = 0; i < this.Length; i++)

            {

                builder.Append($"{this[i]} ");

                if((i + 1) % 10 == 0)

                {

                    builder.Append('\n');

                }

            }

            return builder.ToString();

        }

    }

}

using System;

namespace VectorApp

{

    class Program

    {

        static void Main(string[] args)

        {

            double[] arr1 = new double[] { 1, 2, 3, 4, 5 };

            double[] arr2 = new double[] { 1, 3, -11, 3, 1 };

            Vector vector1 = new Vector(arr1);

            Vector vector2 = new Vector(arr2);

            Console.WriteLine(vector1.dot(vector2));

            Console.WriteLine(vector2.multiply(vector1));

        }

    }

}