МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Кафедра вычислительных технологий**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ОТЧЕТ**

**о выполнении лабораторной работы №1**

**по дисциплине «Паттерны программирования»**

Работу выполнила\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_студентка 39 группы, Глинова Н.В.

Работу проверил\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ст. преп., Жук А. С.

Краснодар 2020

**Задание1**

Был установлен компилятор и текстовый редактор Sublime Text3. Была реализована программа, выводящая сообщение “Hello World”

**Задание2**

Был рассмотрен тип string. Рассмотрены и получены следующие методы:

**Capitalize** – первый символ переводится в верхний регистр

**Capitalize!** – как и обычный capitalize, но если не было изменений в строке, то возвращается nil

**Center (width, padstr = "")** – размещает исходную строку в центр строки размера width, заполняя недостающие символы padstr

**Chars** – возвращает массив, содержащий символы строки, из которой вызывается метод

**Chomp** **(separator=$/)** – возвращает новую строку, которая является копией строки str в которой удален последний символ separator

**Chr** – возвращает односимвольную строку с начала строки

**Clear** – очищает строку

**Concat(\*object)** – конкатинирует строку с данным массивом

**Count(str)** – считает сколько раз встречаются данные символы в строке

**Delete(str)** – возвращает копию строки, в которой удалены все символы из str

**Downcase** – возвращает копию строки str в которой все символы верхнего регистра заменены на соответствующие символы нижнего

**Dump** – возвращает копию строки, в которой все напечатанные символы отображены в виде кодов

**Each\_byte {|fixnum| block}** – передает каждый байт строки str в блок

**Each\_char {|char| block}** – передает каждый символ строки в блок

**Empty?** – возвращает true, если строка пуста

**Eql?** – возвращает true, если строки одинаковые по длине, содержимому, кодировке

**Hash** – возвращает хэш строки

**Include?(str)** – возвращает true, если строка содержит подстроку str

**Index(str)** – возвращает первый индекс, с которого начинается искомая подстрока или первое совпадение с регулярным выражением

**Insert(index, others\_str)** – вставляет строку other\_str, начиная с index

**Length/Size** – возвращают размер строки

**Replace** – заменяет содержимое str соответствующими значениями из other\_str

**Reverse** – возвращает перевернутую строку

**Rindex (substr)** – индекс, где начинается подстрока substr в строке

**Rsjust** – удаляет пробелы и иные символы справа строки

**Slice** – удаляет заданный символ или промежуток симолов из строки и возвращает его

**Split** – делит строку на подстроки

**Strip** – возвращает копию строки, в которой убирает whitespace с начала строки

**Sub** – возвращает подстроку, удовлетворяющую данному регулярному выражению

**Swapcase** – изменяет регистр символов

**To\_f** – переводит строку в floar

**To\_i** – переводит строку в число

**To\_с** – переводит строку в комплексное число

**Upcase** – все символы нижнего регистра заменены на верхний

**Even?** – возвращает true, если число четное

**Odd?** – возвращает true, если число нечетное

**downto(limit) {|i| code }** перебирает все числа от int до limit

**Digits** -возвращает массив

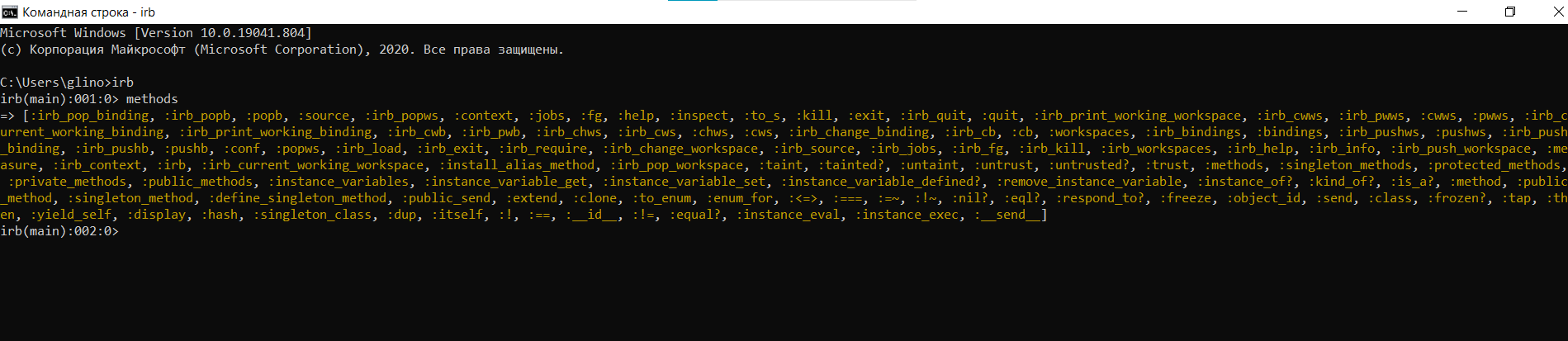
**Each** **{|x| block}** – передает текущий элемент в качестве параметра

**Select/Filter {|i| block}** – выполняет выражение для каждого элемента массива

**Inject(initial) {|acc, i| block}** – обрабатывает элементы применяя к ним блок, принимающий два параметра – аккумулятор и обрабатываемый элемент

**Partition {|x| block}** – возвращает два массива. Результат первого - true, второго – false

Рассмотрен irb режим:



**Задание3**

Были рассмотрены несколько способов построения условного оператора.

Первый способ – использование оператора if.

if condition [then]

code

elif condition [then]

code

else

code

end

Второй способ - использование оператора множественного выбора case.

case var

when somthing1

code

when somthing2

code

else

code

end

**Задание4**

Для вызова системной команды используется функция встроенная функция system.

Функция system возвращает значения true и false, в зависимости от возвращаемого кода команды; nil, если команда не найдена или не может быть исполнена.

Обратные кавычки же возвращают текст, выведенный командой в консоль.

Для выполнения команды языка, введенной пользователем, используется STDIN.

**Задание5**

Была написана программа, выводящая сумму цифр в числе. Для получения цифр в числе, использовался цикл while. Отделяясь с конца, цифра складывалась со следующей отделившейся.

**Задание6**

Была написана программа, в которой в отдельных методах был реализован поиск минимальной, максимальной цифры числа и произведение цифр числа.

**Задание7/9**

Так как 9 задание — это модифицированное 7, то сразу разберем 9. Были реализованы различные методы, такие как:

1. В функции first был использован метод нахождения четного количества чисел, т.е. метод even?, а так же метод который определяет простое число, т.е. метод .gcd.
2. В функции maximum были использованы методы digits, filter, max.

При помощи метода digits берется массив цифр числа, после происходит его обработка. Метод filter удаляет элементы, которые не подходят под условие. Метод max ищет максимальное число.

1. В функции summa использовались методы digits, inject. Метод inject обрабатывает элементы применяя к ним блок, принимающий сумму.

Функция check принимает функцию суммы, одновременно ведя сравнение, является ли сумма меньше 5.

Функция naim ищет второе число, которое не будет не взаимно простоес данным  
Функция pr объединяет в себе все вышеперечисленные функции и перемножает числа, которые подходят под все условия.

**Задание10**

1. Функция collatzSequenceLen проверяет число, и если она удовлетворяет условию, возвращает цепочку чисел.
2. Функция factorial используя метод downto(1) перемножает цифры числа. Функция factorialSum с помощью методов digits, inject складывает числа полученные из функции factorial. Дальше цикл перебирает числа от 3..2540160 ища число, которое будет удовлетворено функции factorialSum.