Министерство образования Республики Беларусь

Учреждения образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №1

По дисциплине ССП за 5 семестр

Тема: «Разработка консольных приложений»

Выполнил:

Студент группы ИИ-21

Кирилович А. А.

Проверил:

Якимук А.В.

Брест 2023

**Цель:** Создать консольные приложения для выполнения поставленных задач.

**Задание:**

1. Реализовать программу для создания массива переменных String и инициализировать массив с названиями месяцев от января до декабря. Создать массив, содержащий 12 случайных десятичных значений между 0.0 и 100.0. Вывести название каждого месяца вместе с соответствующим десятичным значением. Вычислить и вывести среднее значение 12 значений.
2. Реализовать программу для создания массива из десяти переменных String, каждая из которых содержит произвольную строку - месяц/день/год, например 01/09/2022. Проанализировать каждый элемент в массиве и вывести представление даты в формате 01 сентября 2022.
3. Написать программу для создания случайной последовательности N прописных букв, которая не включает гласные буквы.
4. Создать объект типа String и проинициализировать его текстовой строкой. Определить количество гласных, пробелов и общее количество букв.
5. Создать массив объектов типа String и проинициализировать его текстовой строкой. Извлечь из массива строки по введенному разделителю.

**Код 1:**  
use rand::Rng;

fn main() {

let months = [

"январь", "февраль", "март", "апрель", "май", "июнь",

"июль", "август", "сентябрь", "октябрь", "ноябрь", "декабрь"

];

let mut rng = rand::thread\_rng();

let mut values = [0.0; 12];

for i in 0..12 {

values[i] = rng.gen\_range(0.0..100.0);

}

for i in 0..12 {

println!("{}: {:.2}", months[i], values[i]);

}

let sum: f64 = values.iter().sum();

let average = sum / 12.0;

println!("Среднее значение: {:.2}", average);

}

**Код 2:**  
fn main() {

let dates: [&str; 10] = [

"01/09/2022", "15/05/2023", "30/11/2022", "07/03/2024", "20/12/2023",

"10/08/2022", "25/06/2023", "18/10/2024", "12/04/2022", "05/07/2023",

];

let months = [

"января", "февраля", "марта", "апреля", "мая", "июня",

"июля", "августа", "сентября", "октября", "ноября", "декабря"

];

for date in &dates {

let parts: Vec<&str> = date.split('/').collect();

if parts.len() == 3 {

if let (Ok(day), Ok(month), Ok(year)) = (

parts[0].parse::<usize>(),

parts[1].parse::<usize>(),

parts[2].parse::<usize>(),

) {

if month >= 1 && month <= 12 {

println!("{:02} {} {}", day, months[month - 1], year);

} else {

println!("Ошибка: Некорректный месяц: {}", month);

}

} else {

println!("Ошибка: Неверный формат даты: {}", date);

}

} else {

println!("Ошибка: Неверный формат даты: {}", date);

}

}

}

**Код 3:**  
extern crate rand;

use rand::Rng;

fn main() {

let n = 10;

let consonants = "bcdfghjklmnpqrstvwxz";

let mut rng = rand::thread\_rng();

let mut result = String::new();

for \_ in 0..n {

let index = rng.gen\_range(0..consonants.len());

let consonant = &consonants[index..index + 1];

result.push\_str(consonant);

}

println!("Random sequence of consonant letters: {}", result);

}

**Код 4:**  
fn main() {

let text = String::from("Text with vowels and spaces");

let char\_collection: Vec<char> = text.chars().collect();

fn is\_vowel(c: char) -> bool {

matches!(c, 'a' | 'e' | 'i' | 'o' | 'u' | 'y' |

'A' | 'E' | 'I' | 'O' | 'U' | 'Y' )

}

let mut vowel\_count = 0;

let mut space\_count = 0;

for &c in &char\_collection {

if is\_vowel(c) {

vowel\_count += 1;

} else if c == ' ' {

space\_count += 1;

}

}

let total\_letter\_count = char\_collection.len() - space\_count;

println!("Input: {}", text);

println!("Amount of vowel letters: {}", vowel\_count);

println!("Amount of spaces: {}", space\_count);

println!("Amount of letters: {}", total\_letter\_count);

}

**Код 5:**

fn main() {

let text\_array: Vec<String> = vec![

String::from("1"),

String::from("2"),

String::from("3"),

];

println!("Введите разделитель: ");

let mut delimiter = String::new();

std::io::stdin().read\_line(&mut delimiter).expect("Не удалось считать строку");

delimiter = delimiter.trim().to\_string();

let mut extracted\_lines: Vec<String> = Vec::new();

for line in &text\_array {

let split\_lines: Vec<&str> = line.split(&delimiter).collect();

for split\_line in split\_lines {

extracted\_lines.push(split\_line.to\_string());

}

}

println!("Извлеченные строки:");

for extracted\_line in &extracted\_lines {

println!("{}", extracted\_line);

}

}

Вывод: Научился создавать консольные приложения для разных задач.