Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики» (СибГУТИ)

Лабораторная работа №1

по дисциплине

«Программирование мобильных устройств»

Выполнил: студент 4 курса

ИВТ, гр. ИП-813

Лесковец Д.М.

Проверила:

Павлова У.В.

Новосибирск 2021

Оглавление

[Задание 3](#_Toc36477188)

[Выполнение 4](#_Toc36477189)

[Листинг проекта 5](#_Toc36477190)

# Задание

Часть 1

Задание:

1. Написать программу Hello world под android и реализовать функции min, max.
2. Написать 4 unit теста на каждую функцию.
3. Создать пустые проекты на github и gitlab.
4. С помощью средства SmartGit загрузить созданные проекты из локального репозитория на удаленные (github и gitlab).
5. Создать с помощью JavaDoс файл html документации на функции min, max и загрузить их в корневую папку проектов (github и gitlab).
6. Создать в корневой папке gitlab проекта файл [**.gitlab-ci.yml**](https://gitlab.com/Ivandows/test15/-/blob/master/.gitlab-ci.yml) из шаблона android.

Результатом должно быть успешное выполнение компиляции и прогонки тестов в Pipeline.

Часть 2

Задание:

1. Написать VisualStudio программу Helloworld на C#. (например, название проекта – ConsoleApp2)
2. Создать пустые проекты на github и gitlab.
3. С помощью средства Smartgit загрузить созданные проекты из локального репозитория на удаленные (github и gitlab).
4. Создать в локальной корневой папке проекта файл [**.gitlab-ci.yml**](https://gitlab.com/Ivandows/test15/-/blob/master/.gitlab-ci.yml) со следующим содержимым.

*stages:*

*- build*

*build\_job:*

*stage: build*

*only:*

*- branches*

*script:*

*- 'cd ConsoleApp2'*

*- '"C:\Tools\Nuget\nuget.exe" restore'*

*- '"C:\Program Files (x86)\Microsoft Visual Studio\2017\Community\MSBuild\15.0\Bin\msbuild.exe" ConsoleApp2.sln'*

*artifacts:*

*paths:*

*- 'ConsoleApp2\ConsoleApp2\bin\Debug\ConsoleApp2.exe'*

1. Запустите консоль cmd от имени администратора. Установите и запустите gitlab-runner.

*gitlab-ci-multi-runner-windows-amd64 install*

*gitlab-ci-multi-runner-windows-amd64 start*

1. Зарегистрируйте проект и запустите gitlab-runner.

*gitlab-ci-multi-runner-windows-amd64 register*

*https://gitlab.com/*

*t9VVooF1Zu\_G4Sa6MqLX (лежит в gillab→Settings→CI/CD→runner)*

*PMU\_lab1*

*.Net4,Windows*

*true*

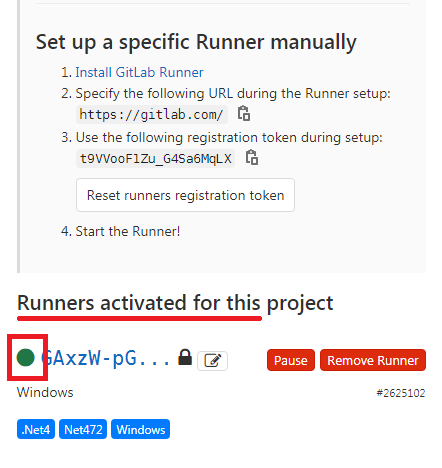
*false*

*shell*

*gitlab-ci-multi-runner-windows-amd64 run*

Убедитесь, что runner подключился

gillab.com→Settings→CI/CD→runner

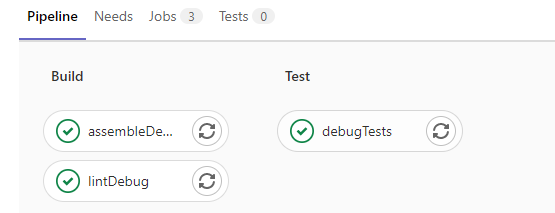


1. Обновите проекты репозиториев github и gitlab или вручную запустите Pipeline.

Результатом должно быть успешное выполнение компиляции C# программы и появление исполняемого файла в архиве artifacts.

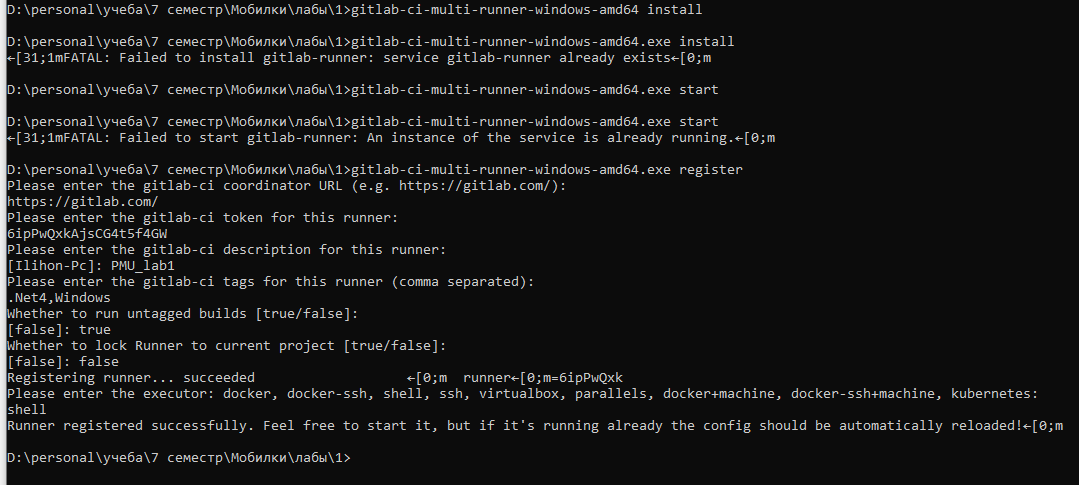
# Выполнение

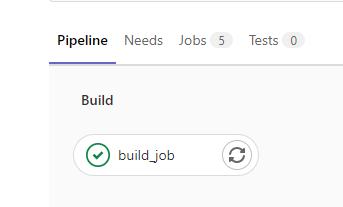
**Часть 1**



Был создан локальный репозиторий, после чего был произведен коммит и загрузка на удаленный репозиторий в github, после чего был создан путем импорта репозиторий на gitlab, где было добавлено CI. Прошла удаленная сборка и пройденны тесты.

**Часть 2**





Был создан локальный репозиторий, после чего был произведен коммит и загрузка на удаленный репозиторий в github, после чего был создан путем импорта репозиторий на gitlab. Был создан файл [**.gitlab-ci.yml**](https://gitlab.com/Ivandows/test15/-/blob/master/.gitlab-ci.yml)**, было установлено приложение gitlab ci runner и затем связано с репозиторием на gitlab. После коммита файла запустилась и прошла сборка.**

# Листинг проекта

MainActivity.java

package com.example.lab1;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.view.ViewDebug;  
import android.widget.TextView;  
  
public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 //TextView textView = (TextView)findViewById(R.id.textView);  
 // textView.setText(textView.getText() + " " + String.valueOf(Max(4,5)+Min(4,5)));  
 }  
  
 /\*\*  
 \*  
 \* @param first integer number  
 \* @param second integer number  
 \* @return maximum between this two numbers  
 \*/  
 public static int Max(int a, int b){  
 if(a>=b){  
 return a;  
 }  
 else{  
 return b;  
 }  
 }  
  
 /\*\*  
 \*  
 \* @param first integer number  
 \* @param second integer number  
 \* @return minimum between this two numbers  
 \*/  
 public static int Min(int a, int b){  
 if(a<=b){  
 return a;  
 }  
 else{  
 return b;  
 }  
 }  
}

MainActivityTests.java

package com.example.lab0;

import junit.framework.TestCase;

import org.junit.Assert;

public class ExampleUnitTest {  
 @Test  
 public void addition\_isCorrect() {  
 assertEquals(5,MainActivity.Max(4,5));  
 assertEquals(8,MainActivity.Max(8,5));  
 assertEquals(4,MainActivity.Max(4,4));  
 assertEquals(5,MainActivity.Max(-5,5));  
 assertEquals(4,MainActivity.Min(4,5));  
 assertEquals(5,MainActivity.Min(8,5));  
 assertEquals(4,MainActivity.Min(4,4));  
 assertEquals(-5,MainActivity.Min(-5,5));  
 }  
}

Program.cs

using System;

namespace Lab1Part2

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Hello World!");

}

}

}