**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

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

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

по дисциплине «Технологии разработки программного обеспечения»

на тему: «Инструменты программирования и отладки. Системы

контроля версий»

Выполнил: студент гр. ИТП-21

Козлов О. В.

Принял: преподаватель

Малиновский И. Л.

Гомель 2023

**Вариант:** 11.

**Задания:**

1. Создать локальный *git* репозиторий;

2. Создать три ветки (*main*, *lr*1, *lr*3). Закоммитить лабораторные работы 1 и 3 в ветки *lr*1 и *lr*3 соответственно;

3. Создать удаленный репозиторий на *GitHub*. Привязать локальный репозиторий к удаленному. Запушить ветки в удаленный репозиторий;

4. Создать и одобрить пул-реквесты на слияние веток *lr*1 и *lr*3 с веткой main;

5. Составить отчет о проделанной работе. Отчет должен содержать описание и скриншоты всех этапов выполнения лабораторной работы;

**Ход работы:**

Что бы создать локальный *git* репозиторий необходимо использовать команду *git init*, которая и инициализирует сам репозиторий для дальнейшей работы с ним. На рисунке 1 показана инициализация репозитория.



Рисунок 1 – Инициализация репозитория

Для создания вето необходимо использовать команду *git checkout*, так же был использован ключ ­-*b* для переключения на новую ветку автоматически. На рисунке 2 показан пример создания ветки.



Рисунок 2 – Создание ветки

После внесения всех изменений в ветку, ее можно закоммитить. Для этого необходимо использовать команду *git commit*. Пример коммита показан на рисунке 3.



Рисунок 3 – Коммит ветки

Когда будет создан удаленный репозиторий, нужно будет добавить его ссылку в локальный репозиторий. Для этого используется команда *git remote add*. На рисунке 4 показано добавление связи с удаленным репозиторием.



Рисунок 4 – Связка репозиториев

Для того что бы запушить ветки из локального репозитория на удаленный необходимо использовать команду *push*. Пример пуша показан на рисунке 5.

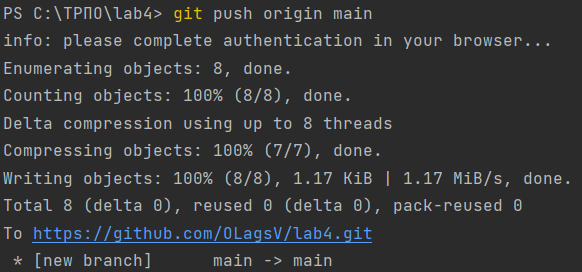


Рисунок 5 – Пуш ветки

Когда обе ветки *lr*1 и *lr*2 запушены их можно слить в одну. Для этого необходимо создать и одобрить пулл-реквесты. Пример пулл-реквеста показа на рисунке 6.

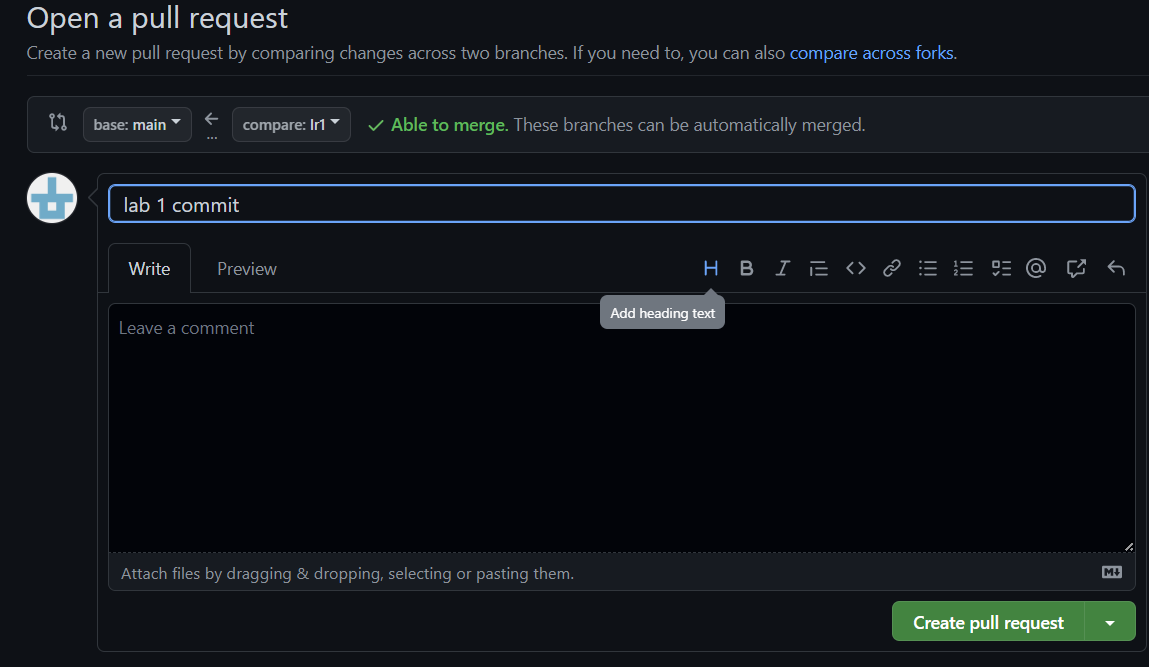


Рисунок 6 – Слияние веток

**Вывод:** в результате выполнения лабораторной работы был создан удаленный и локальный репозитории, созданы ветки, в которые были внесены различные изменении. Были произведены коммиты изменений, которые в результате были запушены в удаленный репозиторий, где было произведено их слияние.

**ПРИЛОЖЕНИЕ А**

(обязательное)

Листинг классов программы

public abstract class Pizza {

double price, weight, diameter, calories;

String name;

public Pizza(String name, double price, double weight, double diameter, double calories) {

this.name = name;

this.price = price;

this.weight = weight;

this.diameter = diameter;

this.calories = calories;

}

public abstract String getInfo();

}}

class Pepperoni extends Pizza {

Boolean sauceCarri;

public Pepperoni(String name, double price, double weight, double diameter, double calories, boolean sauceCarri) {

super(name, price, weight, diameter, calories);

this.sauceCarri = sauceCarri;

}

public String getInfo(){

return "Название: " + name + "\nЦена: " + price + "\nВес: " + weight + "\nДиаметр: " + diameter + "\nКалории: " + calories + (sauceCarri ? "\nС соусом карри" : "Без соуса");

}

}

class Meat extends Pizza {

Boolean sauceBbq;

public Meat(String name, double price, double weight, double diameter, double calories, boolean sauceBbq) {

super(name, price, weight, diameter, calories);

this.sauceBbq = sauceBbq;

}

public String getInfo(){

return "Название: " + name + "\nЦена: " + price + "\nВес: " + weight + "\nДиаметр: " + diameter + "\nКалории: " + calories + (sauceBbq ? "\nС соусом барбекю" : "Без соуса");

}

}

class Cheese extends Pizza {

Boolean sauceGarlic;

public Cheese(String name, double price, double weight, double diameter, double calories, boolean sauceGarlic) {

super(name, price, weight, diameter, calories);

this.sauceGarlic = sauceGarlic;

}

public String getInfo(){

return "Название: " + name + "\nЦена: " + price + "\nВес: " + weight + "\nДиаметр: " + diameter + "\nКалории: " + calories + (sauceGarlic ? "\nС чесночным соусом" : "Без соуса");

}

}

Листинг основной программы

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Pizza[] all\_pizzas = {

new Pepperoni("Маленькая пепперони",18.55, 300, 20, 150.33, true),

new Pepperoni("Средняя пепперони",23.50, 450, 25, 180.46, true),

new Pepperoni("Большая пепперони", 27.19, 600, 30, 240.04, true),

new Meat("Маленькая мясная",23.11, 300, 20, 180.50, true),

new Meat("Средняя мясная",26.40, 450, 25, 225.50, true),

new Meat("Большая мясная",31.20, 600, 30, 300.04, true),

new Cheese("Маленькая сырная",15.60, 300, 20, 130.43, true),

new Cheese("Средняя сырная",19.50, 450, 25, 160.76, true),

new Cheese("Большая сырная",23.60, 600, 30, 220.43, true)

};

Scanner in = new Scanner(System.in);

System.out.println("Введите вес пиццы:");

double weight = in.nextDouble();

System.out.println("Введите диаметр пиццы:");

double diameter = in.nextDouble();

int amount = 0;

for (Pizza pizza : all\_pizzas) {

if (pizza.weight > weight) {

amount++;

}

}

System.out.println("Количество пицц вес которых больше заданного:\n" + amount);

for (Pizza pizza : all\_pizzas) {

if (pizza.diameter > diameter) {

System.out.println("Информация о пиццах диаметр которых больше заданного:\n" + pizza.getInfo());

}

}

}