

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2
дисциплины
«Основы кроссплатформенного программирования»

Выполнил:
Бакулин Вадим Романович
2 курс, группа ИТС-б-о-23-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и
сети»,
очная форма обучения

(подпись)

Проверил:
Доцент департамента цифровых,
робототехнических систем и
электроники Воронкин Р.А

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Тема: Исследование возможностей Git для работы с локальными репозиториями

Цель: исследовать базовые возможности системы контроля версий Git для работы с локальными репозиториями.

Порядок выполнения работы:

1. Были изучены основные теоретические сведения.
2. Был создан общедоступный репозиторий:

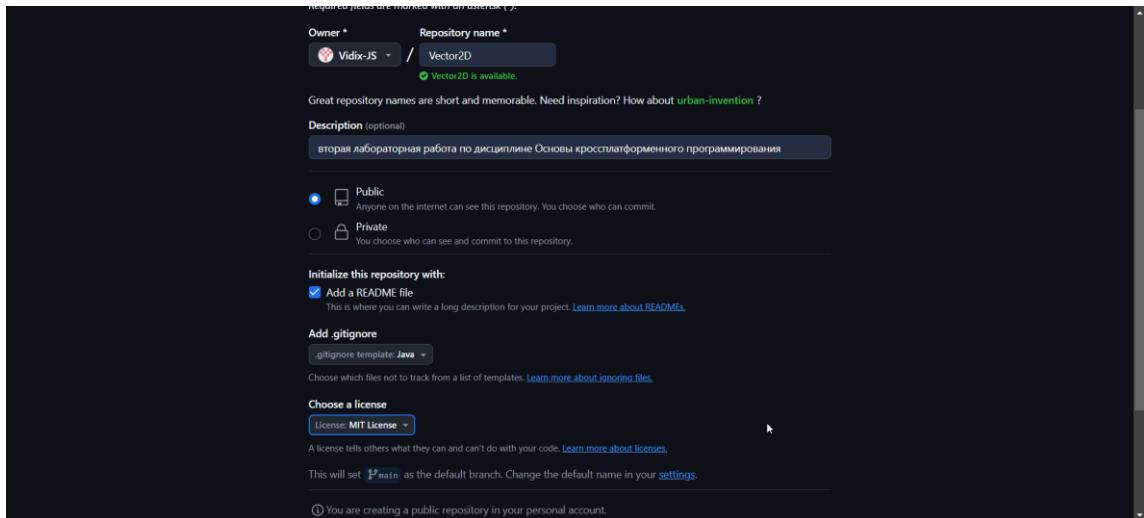


Рисунок 1. Создание общедоступного репозитория

3. Был скопирован репозиторий на компьютер:

```
c:\Users\vadim\Workspace\Vector2D>git clone https://github.com/Vidix-JS/Vector2D.git
Cloning into 'Vector2D'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (5/5), done.
```

Рисунок 2. Клонирование репозитория

4. Была добавлена в файл README.md дополнительная информация:

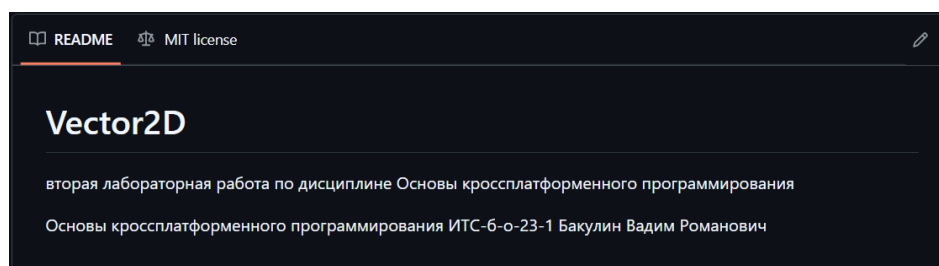


Рисунок 3. Информация в файле README

5. Была написана программа на Java и зафиксированы изменения:

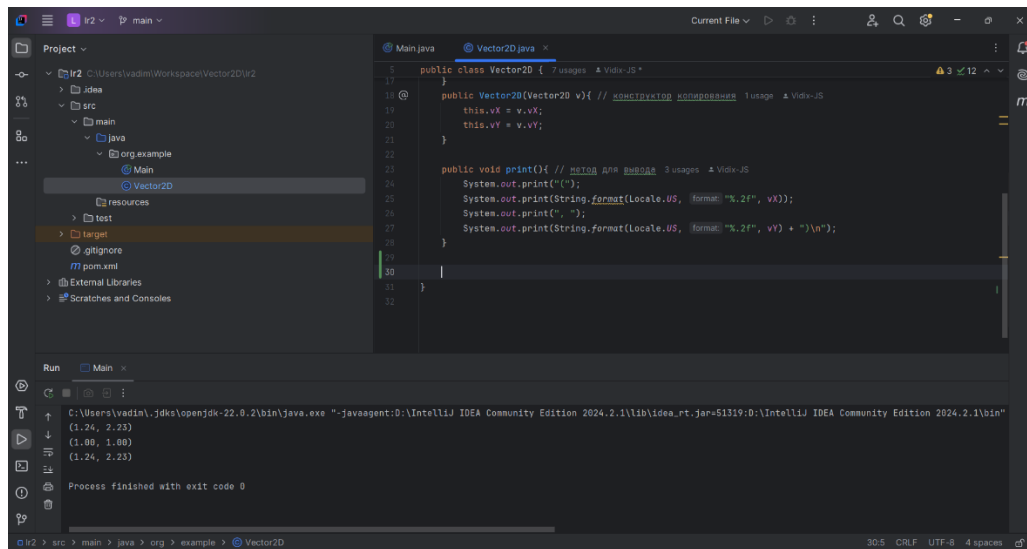


Рисунок 4. Программа, написанная на Java

```
C:\Users\vadim>cd c:\users\vadim\workspace\vector2d
c:\Users\vadim\Workspace\Vector2D>git add .
c:\Users\vadim\Workspace\Vector2D>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md

c:\Users\vadim\Workspace\Vector2D>git commit -m "Изменение файла README.md"
[main 2a36067] Изменение файла README.md
 1 file changed, 4 insertions(+)

c:\Users\vadim\Workspace\Vector2D>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 471 bytes | 235.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Vidix-JS/Vector2D.git
 4872aa6..2a36067  main -> main

c:\Users\vadim\Workspace\Vector2D>
```

Рисунок 5. Фиксирование изменений программы

6. Были добавили теги:

```

c:\Users\vadim\Workspace\Vector2D>git commit -m "Добавление конструктора по умолчанию, конструктора при двух параметрах, конструктора копирования и метод для вывода вектора и использование их в методе main"
[main 07b9a01] Добавление конструктора по умолчанию, конструктора при двух параметрах, конструктора копирования и метод для вывода вектора и использование их в методе main
2 files changed, 30 insertions(+), 1 deletion(-)

c:\Users\vadim\Workspace\Vector2D>git tag -a v1.0 "Базовые функции при работе с векторами"
fatal: Failed to resolve 'Базовые функции при работе с векторами' as a valid ref.

c:\Users\vadim\Workspace\Vector2D>git tag -a v1.0 "version 1.0"
fatal: Failed to resolve 'version 1.0' as a valid ref.

c:\Users\vadim\Workspace\Vector2D>git tag -a v1.0 -m "Базовые функции при работе с векторами"

c:\Users\vadim\Workspace\Vector2D>git push
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (10/10), 1.20 KiB | 409.00 KiB/s, done.
Total 10 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Vidix-JS/Vector2D.git
 9d6ee93..07b9a01  main -> main

c:\Users\vadim\Workspace\Vector2D>git push origin v1.0
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 227 bytes | 227.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Vidix-JS/Vector2D.git
  v1.0 -> v1.0
* [new tag]

```

Рисунок 6. Добавление тега

7. Были созданы не менее 7 коммитов:







 Vidix-JS Добавление метода по умножению вектора	eee10d0 · 1 hour ago	 9 Commits
 lr2	Добавление метода по умножению вектора	1 hour ago
 .gitignore	Initial commit	2 hours ago
 LICENSE	Initial commit	2 hours ago
 README.md	Изменение файла README.md	1 hour ago

Рисунок 7. Созданы 9 коммитов

8. Была просмотрена история коммитов с помощью команды git log

```

c:\Users\vadim\Workspace\Vector2D>git log --graph --pretty=oneline --abbrev-commit
* eee10d0 (HEAD -> main, tag: v1.0.4, origin/main, origin/HEAD) Добавление метода по умножению вектора
* 6a4b8b8 (tag: v1.0.3) Добавил метод во вычитания одного вектора из другого и проверил работу в методе main
* b1e0dca (tag: v1.0.2) Добавил метод для вычисления длины вектора
* 1924cf9 (tag: v1.0.1) Добавил метод для суммы двух векторов и проверил его работу в методе main
* c5d4eb4 Добавление геттеров и сеттеров для определения X и Y
* 07b9a01 (tag: v1.0) Добавление конструктора по умолчанию, конструктора при двух параметрах, конструктора копирования и метод для вывода вектора и использова
ние их в методе main
* 9d6ee93 Создание проекта для расчета и работы с векторами
* 2a36067 Изменение файла README.md
* 4872aa6 Initial commit

```

Рисунок 8. Просмотр истории коммитов

9. Просмотр последнего коммита с помощью команды git show HEAD:

```
c:\Users\vadim\Workspace\Vector2D>git show HEAD
commit eee10d021b9da240c8072484090278bfc823a4ec (HEAD -> main, tag: v1.0.4, origin/main, origin/HEAD)
Author: Vidix-JS <vadim.bakulin.1996@mail.ru>
Date: Thu Dec 5 22:48:39 2024 +0300

    Добавление метода по умножению вектора

diff --git a/lr2/src/main/java/org/example/Main.java b/lr2/src/main/java/org/example/Main.java
index bc73b4a..1b8a1cd 100644
--- a/lr2/src/main/java/org/example/Main.java
+++ b/lr2/src/main/java/org/example/Main.java
@@ -11,5 +11,7 @@ public class Main {
     System.out.println(vB.length());
     vA.sub(vC);
     vA.print();
+    vC.scale(2);
+    vC.print();
 }

\ No newline at end of file
diff --git a/lr2/src/main/java/org/example/Vector2D.java b/lr2/src/main/java/org/example/Vector2D.java
index eedc42b..ee7ef59 100644
--- a/lr2/src/main/java/org/example/Vector2D.java
+++ b/lr2/src/main/java/org/example/Vector2D.java
@@ -58,4 +58,9 @@ public class Vector2D {
     return Math.sqrt((Math.pow(this.vX, 2) + Math.pow(this.vY, 2)));
 }

+ public void scale(double scaleFactor){ // Метод умножающий текущий вектор на scaleFactor.
+     this.vX = vX * scaleFactor;
+     this.vY = vY * scaleFactor;
+ }
+ }
```

Рисунок 9. Последний коммит

10. Просмотр предпоследнего коммита:

```
Командная строка - git show
c:\Users\vadim\Workspace\Vector2D>git show HEAD~1
commit 6a4b8b8495093bd066c49640448295d644d63755 (tag: v1.0.3)
Author: Vidix-JS <vadim.bakulin.1996@mail.ru>
Date: Thu Dec 5 22:45:51 2024 +0300

    Добавил метод во вычитанию одного вектора из другого и проверил работу в методе main

diff --git a/lr2/src/main/java/org/example/Main.java b/lr2/src/main/java/org/example/Main.java
index 13564f1..bc73b4a 100644
--- a/lr2/src/main/java/org/example/Main.java
+++ b/lr2/src/main/java/org/example/Main.java
@@ -9,5 +9,7 @@ public class Main {
     vA.add(vB);
     vA.print();
     System.out.println(vB.length());
+    vA.sub(vC);
+    vA.print();
 }

\ No newline at end of file
diff --git a/lr2/src/main/java/org/example/Vector2D.java b/lr2/src/main/java/org/example/Vector2D.java
index e518e28..eedc42b 100644
--- a/lr2/src/main/java/org/example/Vector2D.java
+++ b/lr2/src/main/java/org/example/Vector2D.java
@@ -6,21 +6,22 @@ public class Vector2D {
     private double vX; // Значение проекции вектора оси
     private double vY; // Значение проекции вектора оси

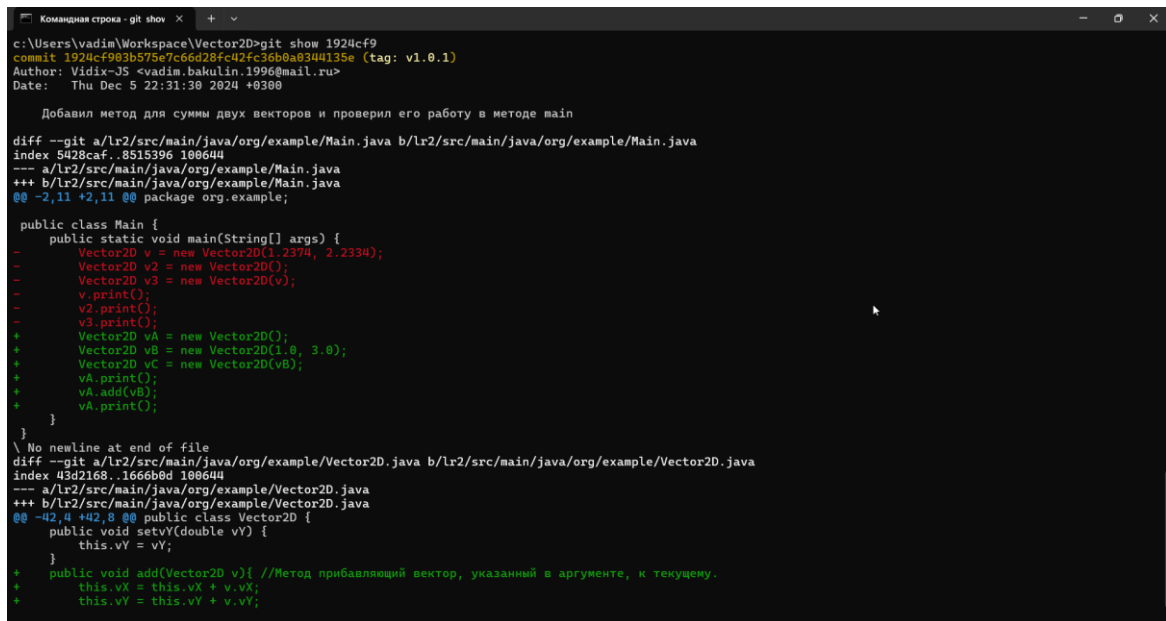
-    public Vector2D() { // конструктор без параметров
+    public Vector2D() { // конструктор без параметров
         this.vX = 1;
         this.vY = 1;
     }

    Date: Thu Dec 5 22:45:51 2024 +0300

Author: Vidix-JS <vadim.bakulin.1996@mail.ru>
commit 6a4b8b8495093bd066c49640448295d644d63755 (tag: v1.0.3)
Author: Vidix-JS <vadim.bakulin.1996@mail.ru>
Date: Thu Dec 5 22:45:51 2024 +0300
```

Рисунок 10. Предпоследний коммит

11. Просмотр коммита через указанный хэш:



```
Командная строка - git show
c:\Users\vadim\Workspace\Vector2D>git show 1924cf9
commit 1924cf9b575e7c66d28fc42fc36b0a0344135e (tag: v1.0.1)
Author: Vidix-JS <vadim.bakulin.1996@mail.ru>
Date: Thu Dec 5 22:31:30 2024 +0300

Добавил метод для суммы двух векторов и проверил его работу в методе main

diff --git a/lr2/src/main/java/org/example/Main.java b/lr2/src/main/java/org/example/Main.java
index 5428caf..8515396 100644
--- a/lr2/src/main/java/org/example/Main.java
+++ b/lr2/src/main/java/org/example/Main.java
@@ -2,11 +2,11 @@ package org.example;

public class Main {
    public static void main(String[] args) {
-        Vector2D v = new Vector2D(1.2374, 2.2334);
-        Vector2D v2 = new Vector2D();
-        Vector2D v3 = new Vector2D(v);
+        Vector2D vA = new Vector2D();
+        Vector2D vB = new Vector2D(1.0, 3.0);
+        Vector2D vC = new Vector2D(vB);
+        vA.print();
+        vA.add(vB);
+        vA.print();
    }
}
\ No newline at end of file
diff --git a/lr2/src/main/java/org/example/Vector2D.java b/lr2/src/main/java/org/example/Vector2D.java
index 4342169..1666b0d 100644
--- a/lr2/src/main/java/org/example/Vector2D.java
+++ b/lr2/src/main/java/org/example/Vector2D.java
@@ -42,4 +42,8 @@ public class Vector2D {
    public void setvY(double vY) {
        this.vY = vY;
    }
+    public void add(Vector2D v){ //Метод прибавляющий вектор, указанный в аргументе, к текущему.
+        this.vX = this.vX + v.vX;
+        this.vY = this.vY + v.vY;
    }
}
```

Рисунок 11. Коммит с указанным хэшем

12. Освоил возможность отката к заданной версии. Удалил весь код из одного из файлов программы репозитория и сохранил этот файл.

13. Удалил все несохраненные изменения в файле командой `git checkout`:

```
C:\Users\vadim\Workspace\Vector2D\lr2\src\main\java\org\example>git checkout -- Main.java
C:\Users\vadim\Workspace\Vector2D\lr2\src\main\java\org\example>
```

Рисунок 12. Использование команды `git checkout`

14. Откатил состояние хранилища к предыдущей версии:

```

C:\Users\vadim\Workspace\Vector2D>git add .
C:\Users\vadim\Workspace\Vector2D>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   lr2/src/main/java/org/example/Main.java

C:\Users\vadim\Workspace\Vector2D>git commit -m "Удаление из файла"
[main c2ac76b] Удаление из файла
 1 file changed, 1 insertion(+), 11 deletions(-)

C:\Users\vadim\Workspace\Vector2D>git reset --hard HEAD~1
HEAD is now at eee10d0 Добавление метода по умножению вектора

C:\Users\vadim\Workspace\Vector2D>

```

Рисунок 13. Откат изменений

Вывод: в результате использования команды `git reset --hard` отменяется последние коммиты и сбрасывается рабочая копия до определенного состояния.

Ответы на контрольные вопросы:

1. Как выполнить историю коммитов в Git? Какие существуют дополнительные опции для просмотра истории коммитов?

Для просмотра истории коммитов в Git используется команда `git log`. Она отображает список коммитов в хронологическом порядке, начиная с самого последнего.

При помощи аргумента `-p` можно посмотреть разницу внесенную в каждый коммит.

При помощи параметра `-n`, где `n` это число, можно посмотреть историю последних `n`-коммитов.

2. Как ограничить вывод при просмотре истории коммитов?

Для ограничения вывода при просмотре истории коммитов в Git можно использовать несколько опций:

- Ограничение количества коммитов; `-(n)`;
- Фильтрация по времени;

С помощью `-since`, `-after` показывает те коммиты, которые сделаны после указанной даты

С помощью `--until`, `--before` показывает только те коммиты, которые были сделаны до указанной даты.

- Фильтрация по автору;
`--author`
- Фильтрация по сообщению коммита;
`--committer`
- Ограничение вывода по формату.
`--pretty=format'''`

3. Как внести изменения в уже сделанный коммит?

- `git commit --amend`;
- `git reset <имя_коммита>`.

4. Как отменить индексацию файла в Git?

`git checkout`. Эта команда отменит все изменения, внесенные в файлы, вернув их к состоянию в последнем коммите.

`git reset HEAD`. Эта команда удалит файл из индекса, но оставит его в рабочей области.

5. Как отменить изменения в файле?

В Git есть несколько способов отменить изменения в файле:

- `git checkout -- <имя файла>`;
- `git reset HEAD <имя файла>`;
- `git restore <имя файла>`;
- `git revert <имя файла>`;
- `git stash`

6. Что такое удаленный репозиторий Git?

Удаленный репозиторий Git – это хранилище кода, доступное с разных компьютеров через сеть.

7. Как выполнить просмотр удаленных репозиториях данного локального репозитория?

Для просмотра удаленных репозиториях, связанных с вашим локальным репозиторием Git, используйте команду `git remote`. Эта команда выведет

список имен, удаленных репозиторий, которые вы добавили к своему локальному репозиторию.

8. Как добавить удаленный репозиторий для данного локального репозитория?

Чтобы добавить удаленный репозиторий к вашему локальному репозиторию Git, используйте команду `git remote add`: `git remote add <имя_удаленного_репозитория> <URL_удаленного_репозитория>`

9. Как выполнить отправку/получение изменений с удаленного репозитория?

Отправка изменений (push): `git push <имя_удаленного_репозитория><ветвь>`: Эта команда отправляет ваши локальные изменения в указанную ветвь на удаленный репозиторий.

Получение изменений (pull): `git pull <имя_удаленного_репозитория><ветвь>`: Эта команда получает изменения из указанной ветви на удаленном репозитории и объединяет их с вашей локальной веткой

10. Как выполнить просмотр удаленного репозитория?

- С помощью команды `git log`;
- С помощью команды `git fetch`;
- С помощью команды `git remote show`.

11. Каково назначение тэгов Git?

Тэги Git – это метки, которые позволяют пометить определенные коммиты в репозитории. Они служат для идентификации важных моментов в истории проекта, таких как релизы, версии или отправные точки для ветвления.

12. Как осуществляется работа с тэгами Git?

- Создание тега;
- Просмотр тегов;
- Перемещение тега;
- Удаление тега;

- Отправка тегов на удаленный репозиторий;
- Просмотр информации о теге;
- Переключение на коммит, помеченный тегом.

13. Самостоятельно изучите назначение флага `--prune` в командах `git fetch` и `git push`. Каково назначение этого флага?

Флаг `--prune` в командах `git fetch` и `git push` используется для удаления удаленных веток, которые уже не существуют на удаленном сервере.

Вывод: в ходе работы исследовал базовые возможности системы контроля версий Git для работы с локальными репозиториями.

Ссылка на GitHub: <https://github.com/Vidix-JS/Vector2D>