

Прашања прв колоквиум

1. Кои се општите правила (rules of thumb) на анализата?

*Одложување на барања за инфраструктурата, намалување на поврзувањето(coupling) на модулите и едноставност на моделот

*Вклучување на барањата на инфраструктурата, намалување на поврзувањето(coupling) на модулите и едноставноста на моделот

*Одложување на барањата за инфраструктурата, зголемување на поврзувањето(coupling) на модулите и едноставноста на моделот

*Вклучување на барањата за инфраструктурата, зголемување на поврзувањето(coupling) на модулите и едноставноста на моделот

2. Што преставува инженерството на барањата?

*Процес на дефинирање на корисничките барања

*Процес на дефинирање на системските барања

*Процес на утврдувања на ограничувањата под кои ќе работи и ќе се развива системот

*Процес на утврдување кои услуги ги бара клиентот од системот и под кои ограничувања ќе работи и ќе се развива системот

3. Кои од следниве фактори НЕ се однесуваат на големите системи кои треба да се реализираат со агилните методологии:

*Систем во систем

*Независност од законските ограничувања

*Различни чинители

*Системска конфигурација.

4. Кој е модалитетот на релација меѓу објектот студент и објектот професор во iknow:

*0

* m:1

*n:1

* 1

5. Што НЕ е мерливо кај нефункционалните барања ?

*Квалитет на спецификациите

* Едноставност на користење

*Робусноста

*Брзината на софтверскиот производ

6. Што од следново НЕ се однесува на агилноста на софтверскиот процес?

*Индивидуите и интеракцијата е поважна од тестирањето

*Поважно е да се изработи софтвер кој одговара на сегашните барања на клиентот отколку да се грижиме за функционалностите кои можеби би биле потребни во иднина

*Одзивот на барањата за промени е поважно отколку следењето на план

*Поважно е да се изгради софтвер кој работи отколку да се изработи обемна документација.

7.Една предност на процесите со инкрементирана испорака е што _____ тенденција да бидат најмногу тестирали.

*сигурносно-критичните системски сервиси

*сервисите испорачани во последниот инкремент

***најприоритетните системски сервиси**

*безбедносно критични системски сервиси

8.Што преставува програмирање во пар?

*Начин за поправање на грешки во програма барајќи парови на грешки

*Алгоритам кој пронаоѓа парови на броеви во множество на цели броеви

*Процес на програмирање во кој програмерот ја пишува програмата два пати за секој случај.

***Процес на развој на софтвер во кој два развивачи на софтвер работат на иста работна станица**

9.Моделирањето на однесувањето треба да претставува:

***Како системот реагира на надворешни настани**

*како системот реагира на внатрешни настани

*како системот реагира на интеракција со корисникот

*како системот реагира на тестирање

10.Која е главната одговорност на тестерите во агилната околина на SCRUM?

*Креирање автоматизирани скрипти

*крирање тест сценарија и тест случаеви

***Во SCRUM не постои улогата тестер**

*Наоѓање грешки

11.Вашата компанија почнува нов продукт за познат деловен партнери.Целиот процес вклучува неколку фази.Класифицирајте ги активностите во соодветните процеси:

*Discovery,classification,prioritazation,negotiation and specification-**ELICITATION**

*Writing down the user and system requirements-**SPECIFICATION**

*Demonstrating that the requirements fulfill customer needs-**VALIDATION**

*Gathering information about the required and existing systems and distlinng the user and system requirements-**ELICITATION**.

12.Кој од следниве термини се однесува на степен во општествениот развој за време на кој економијата се поместила од примарно обезбедување на добра кон примарно обезбедување на сервиси?

*Земјоделско општество

***Пост индустриско општество**

*Пост земјоделско општество

*Индустриско општество

13.SPICE овозможува стандарди за процесот на развој на софтвер и поврзаните функции за бизнис менаџмент -**ТОЧНО**

14.Поврзете ги следниве описи со точниот критериум за квалитет:

*Придржување кон спецификациите **ИСПРАВНОСТ**

*Способност на компјутерскиот систем да се спарви со грешките во текот на работата и со неточните влезови - **РОБУСНОСТ**

*Степен на разбираливоста,можноста за поправка и надградба-**ИСПРАВНОСТ**

*Веројатноста дека софтверскиот производ ќе работи без грешки во даден временски период-**НАДЕЖНОСТ**

*Интерфејс што ги поседува овие атрибути:едноставен,чист,интуитивен и надежден-**НАСОЧЕН КОН КОРИСНИКОТ**

15.АПстрактните изрази од високо ниво се дел од формализираните софтверски барања,додека пак формалните барања се представени од детални математичко функционални спецификации.**ТОЧНО**

16.Структирани,базирани на формулар,табуларни спецификации и кориснички сценарија се дел од документот за софтверските барања(SRD).Изберете ги точните факти за SRD:

***SRD ги вклучува корисничките и системските побарувања**

***SRD поставува што треба да работи системот**

*SRD поставува како треба да работи системот

*SRD е документ за дизајн

***SRD е официјален документ наменет за развивачи на софтвер**

17.Кој пар од наведените модели(Обата) не би можеле да го дадат посакуваниот излезд без инволвираност на корисникот за целото времетраје на проектот.

***XP & Prototyping**

***Waterfall & Spiral**

*XP & Waterfall

*XP & Spiral

18.Кои се предностите на инкрементираниот развој ?

*Структурата на системот не деградира при внесувањето нови инкременти

*Процесот кај инкременталниот развој е целосно видлив

*Повратната информација за сработеното полесно се добива

*Прилагодувањето кон променливите барања на клиентите е поефтина

19.Кога се користи Scrum,врз основа на што се прави проекција на датумот на изданието во кое ќе бидат имплементирани дадени (фиксирани) функционалности:

*Брзината

*Бројот на спринтови до сеа

*големината на Scrum тимот

*Бројот на планирани идни спринтови

What is Extreme Programming? // Што е екстремно програмирање?

Select one:

- a. It is a set of software development practices that are aimed to create reliable software using extensive feedback, developer engagement, and small development iterations. // Множество од практики за развој на софтвер които се целат кон креирање на доверлив софтвер со користење на обемен фидбек, ангажирање на девелопери и мали итерации на развој.
- b. It is a problem-solving process based on investigating hundreds of possibilities before choosing one. // Тоа е процес на решавање на проблеми/задачи што вклучува истражување на стотици можности пред да се избере вистинското решение.
- c. It is programming method that involves programming during very long hours. // Метод на програмирање што вклучува програмирање кое што може да трае со часови.
- d. It is a programming method that involves searching for very complicated algorithms to solve problems. // метод на програмирање што вклучува пребарување на многу комплицирани алгоритми за решавање на различни проблеми.

(slika za 4to prashanje)

Tochno e pod a.

Поврзете ја секоја карактеристика на еден добар VCS со соодветната команда која ја овозможува таа карактеристика во Git VCS. // Match the property of a good version control system with the corresponding command that applies that property in the Git version control system.

Синхронизација/Synchronization:

git push/pull

Short-term undo (Краткотрајно враќање):

revert

Track Changes (Следење на промените):

status

Long-term undo (Долготрајно враќање):

reset

Backup and Restore (Резервна копија и враќање):

clone

Branching (разгранување):

checkout -b

Merging of branches (спојување на гранки):

merge

Which one of the following is NOT a fundamental activity for software processes in software engineering? / Која од следните НЕ е фундаментална активност кај софтверските процеси во софтверското инженерство?

Select one: a.) Software evolution / Еволуција

b.) Software Verification / Верификација - ова не е fundamentalna

c.) Software Validation / Валидација

d.) Software design and implementation / Дизајн и имплементација

Failure rate is crucial for software reliability. How does it behave during the useful life of one software product? // Количеството неуспешност е клучно за софтверската надежност. Каква е неговата вредност додека софтверскиот производ е во употреба?

Select one or more:

- a. It becomes stable whenever it reaches its time of obsolescence // Се стабилизира секогаш кога го достигнува периодот на одвршност
- b. It reaches its minimum prior to delivery // Го достигнува својот минимум пред испораката
- c. It is higher than the failure rate during testing and debugging // Повисока е отколку во периодот на тестирањето и дебагирањето
- d. It increases after each upgrade // Се зголемува по секоја надградба
- e. It decreases after each upgrade // Се намалува по секоја надградба

Кои се четирите клучни професионални обврски? / Which are the four major issues of professional responsibility?

Select one or more:

- a. Стручност, право на приватност, право на безбедност / Competence, Privacy rights, Security rights
- b. Стручност, доверливост, злоупотреба на компјутерите / Competence, Confidentiality, Computer misuse
- c. Стручност, злоупотреба на компјутерите, право на интелектуална сопственост / Competence, Computer misuse, IP rights
- d. Злоупотреба на компјутерите, право на интелектуална сопственост, право на приватност / Computer misuse, IP rights, Privacy rights

Кои се четирите клучни професионални обврски? / Which are the four major issues of professional responsibility?

Select one or more:

- a. Злоупотреба на компјутерите, право на интелектуална сопственост, право на приватност / Computer misuse, IP rights, Privacy rights
- b. Стручност, право на приватност, право на безбедност / Competence, Privacy rights, Security rights
- c. Стручност, доверливост, злоупотреба на компјутерите / Competence, Confidentiality, Computer misuse
- d. Стручност, злоупотреба на компјутерите, право на интелектуална сопственост / Competence, Computer misuse, IP rights

C, D

Next page

Која е кардиналноста на релацијата помеѓу ентитетите студент и професор (на еден факултет)? / What is the cardinality of the relation between the entities student and professor (on one faculty)?

Select one:

- a. 1:M
- b. M:1
- c. 1:1
- d. M:M

B
D

Што означува кратенката CASE во софтверското инженерство? / What does CASE mean in software engineering?

Select one:

- a. Copy And Steal Everything
- b. Софтверски системи наменети да обезбедат автоматизирана поддршка за софтверските процесни активности / Software systems intended to provide automated process activities
- c. Вообичани елементи за апликациските сервиси / Common Application Service Elements
- d. Проценка на усогласеноста на сигурносните аспекти / Conformity Assessment Security Evaluation

B

Согласно со извештајот на Global Information Technology Industry, глобалната вредност на ИТ индустријата надминува 5 трилиони американски долари. Споредена со земјоделието, ИТ индустријата вреди: / According to Global Information Technology Industry, IT industry global value is over 5 trillion US\$. Compared with agriculture, IT industry worth is:

Select one:

- a. повеќе од двојно / more than twice as much
- b. приближно еднакво / approximately the same
- c. скоро двојно повеќе / almost twice as much
- d. половина од вредноста на земјоделието / half the value of agriculture

[Clear my choice](#)

A

PODA E TOCHNO zemjodelskata e 2.4 trilioni a IT 5.2

Кои се засегнатите страни кои што пристапуваат со корисничките и системските барања?
and system requirements?

C i A i B i D

Select one or more:

- a. Client managers // менаџерите на клиентска страна
- b. System architects // архитектите на системот
- c. System end-users // крајните корисници на системот
- d. Contractor managers // менаџерите на извршувачите

100% tochno

site se tochni

VAKA KJE BESHE AKO E SO ILI... TOCHNO E POD B I C SAMO!!!!

Работите во софтверска компанија која што го употребува **Gitflow workflow** моделот за колаборација на проект. Добивате нова апликација. Подредете ги git командите како што би ги употребиле за да се справите со овој предизвик. Се претпоставува дека commit. Моментално се наоѓате на гранката master. Користете една команда за креирање и префрлување на нова гранка за реш workflow. You're assigned a task to develop a new web page for registration of VIP users on your application. Sort the git commands that will be simple and fast and you'll need only one git commit for it. At the moment you are on the master branch. Use only one command to

- | | | | |
|---|--|---|---|
| 1 | git checkout develop | + | ✓ |
| 2 | git checkout -b feature | + | ✓ |
| 3 | git add & git commit -m "some message" | + | ✓ |
| 4 | git checkout develop | + | ✓ |
| 5 | git merge --no-ff release | + | ✗ |
| 6 | git branch -d release | + | ✗ |

Give your reasons

Според Gitflow workflow кога развиваме нова задача не треба да работиме директно на мастер, туку веќе има предодредени правила. Доколку сакаме да додадеме нов feature ја разгрануваме гранката develop и таму работиме, откако завршуваме со feature, вклучиме феатурот во новата верзија.

Your answer is partially correct.

You have correctly selected 4.

The correct answer is: 1 → git checkout develop, 2 → git checkout -b feature, 3 → git add & git commit -m "some message", 4 → git c

1 Monday, 14.06.2019

Работите во софтверска компанија каде што го употребува Gitflow workflow моделот за колаборација на проект. Дебивате нова задача да развиете нова страна за регистрација на VIP корисници на вашата апликација. Подредете ги git командите како што би ги употребиле за да се справите со овој предизвик. Се претставува дека решението ќе биде брзо и едноставно и ќе ја имајте готова една commit. Моментално се наоѓате на гранката master. Користете една команда за креирање и префлување на нова гранка за решението да биде во 6 чекори! // You're working for a software company that uses Gitflow workflow. You're assigned a task to develop a new web page for registration of VIP users on your application. Sort the git commands that you would use in order to complete your task. It is assumed that the solution of the task will be simple and fast and you'll need only one git commit for it. At the moment you are on the master branch. Use only one command to create and checkout to a new branch, so that the solution can be in 6 steps!

1	git checkout develop	✓
2	git checkout -b feature	✓
3	git add & git commit -m "some message"	✓
4	git checkout develop	✓
5	git merge --no-ff feature	✗
6	git branch -d feature	✗

Give your reasoning

Според Gitflow workflow кога развишаме нова задача не треба да работиме директно на мастер, туку ако има предопределено граници каде што тоа треба да се случува. Развиваме на гранката девелоп, а доколку сакаме да додадеме нов feature ја разгрунуваме гранката девелоп и таму работиме, откако завршуваате со featureот го спојуваме со гранката девелоп, тестираме и одлучуваме дали ќе го вклучиме фичерот во новата верзија.

Your answer is partially correct.

You have correctly selected 4.

The correct answer is: 1 → git checkout develop, 2 → git checkout -b feature, 3 → git add & git commit - m "some message", 4 → git checkout develop, 5 → git merge --no-ff feature, 6 → git branch -d feature

Прашања од снимки

- 1) Поврзете ги соодветното тврдење со системот за контрола на верзии за кој важи
-Секој што го развива софтверот(developer) има локална копија на софтверот и историја на сите промени. **GIT**
-Комплетно сите акции мора да се реализираат онлајн **SVN**
- При креирањето, датотеките автоматски стануваат дел од системот за контрола на верзии. **NITU EDEN**

- 2) Дадена е папката SI_prv_kolokvium (воедни и локален гит репо) со следната содржина:
Папката SI_prv_kolokvium содржи и друга папка именувана срц:
Во продолжение се дадените излези од две извршување на командата git status
Кога од staged → unmodified (git commit) станува збор
Кога од untracked → staged (git add)
Направен commit
Git add *.txt
git add “*.java”-recursive

- 3) Преземете ја датотеката и со цмд да користиме гит команди

*Напишете го името на втората гранка која постои во репото(покрај мастер)

За да ги видиме сите гранки GIT BRANCH

*На која гранка се наоѓа HEADOT моментално? GIT STATUS bugfix

*Колку комити има на гранката мастер? -Се префрлуваме на мастер и со git log добиваме колку комити има на мастер.2Комити.

*Колку комити има на bugfix?-Се префрлуваме на гранката bugfix и праввиме git log.3 vo slucajov

*Репото е поврзано со ремоте репо? -Со командата git remote -v

*Спојувањето на втората гранка кон гранката мастер резултира со конфликт и која е стратегијата?

-ФФ(Fast forward) Нема да резултира со конфликт.

- 4) Во директориумот(воедно и локален git repo)SI_K1 се наоѓа програмата Variables.java со следната содржина

Public class variables{

 Int a=10;

 Int b=7;

 Int c=8;

}

Локалното репо е поврзано со ремот репото origin.

Оваа верзија од датотеката е зачувана со првиот комит на master.(git commit -m “Initial commit”).По тој комит,корисникот ги извршува следните команди

Git commit -am “Add d=4”

Git checkout -b feature (site promeni so gi ima na master ke gi ima i na feature)

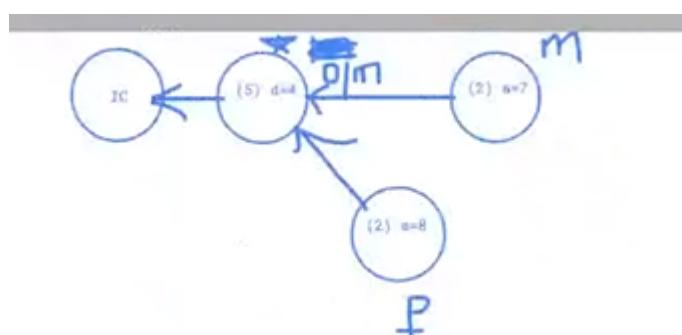
Git commit -am “Change a to 8”

Git push origin master

Git commit -am “Change a to 7”

Пораката во формат Add x=y означува додавање на нова променлива на крајот на класата,Пораката во форматот “Change X to Y” значи дека вредноста на променливата X е променета во Y.

master (local)	feature(local)	master(remote)	feature(remote)
A – 7	8	10	/
B – 7	7	7	/
C – 8	8	8	/
D – 4	4	4	/



*Recursive (при merge ќе се додади)

1.1) Сценарио 1 : Корисникот сака да ги изврши командите git checkout master & git merge feature?

*Не, ќе се јави конфликт

1.2) По која стратегија би се извршило спојувањето на гранките?

*Рекурзивно

1.3) Напишете ги вредностите на проемнливите a,b,c,d на гранката master по спојувањето?

a=/(конфликт) b=7 c=8 d=4

1.4) По спојувањето на двете гранки ќе има вкупно 5 комити.

2. Сценарио 2. Корисникот сака да ги изврши комадните git checkout master & git reset –hard HEAD^^

Напишете ги вредностите на променливите a,b,c,d на гранката feature

По извршувањето на командите

a=10 b=7 c=8 d=/

3. Сценариот 3. Корисникот сака да ги изврши командите git checkout master & git rebase feature.

Во кој редослед ќе бидат commit-ите на гранките master по извршувањето на командите:

1 → Initial commit

2 → Add d=4

3 → Change a=8

4 → Change a=7

5 → Нема да постои (Merge commit се одбегнуваат)

5) Претпоставите дека работите на некој гит репо чии последни промени изгледаат како што е прикажано подолу. За секој изглед на едиторот во кој би ја промениле историјата на репото, изберете ја точната команда со која е повикан Interactive rebase.

6 A7328aha Add new feature X

5 Dbeb4b343 Fix bug in feature Y

4 Iaaubuwee23 Refactor code for improved performance

3 Eih34hgj3j Update documentation for feature Z

2 Wugeug34 Implement feature W

1 N1jh2119b Initial commit

1.1) A7328aha Add new feature X

Git rebase -i HEAD^

1.2) за овој изглед

6 A7328aha Add new feature X

5 Dbeb4b343 Fix bug in feature Y

4 Iaaubuwee23 Refactor code for improved performance

Git rebase -i HEAD-3

6) Дадено е сценариото на поправање на грешки во gitflow workflow преку соодветни Git команди :

1. Git checkout develop
2. Git checkout -b hotfix123
3. Git add .
4. Git commit -m "Fix bug"
5. Git checkout master
6. Git merge --no-ff hotfix123
7. Git tag v1.1
8. Git checkout develop
9. Git merge --no-ff hotfix123

*Која команда е погрешна?1

*Која гит команда треба да стои место погрешната ?

Git checkout master

--no-ff flag избегнува fast forward спојување

*Bo gitflow workflow на гранката develop се наоѓа целосната историја на работа на проектот(развојот на сите features, поправки на багови)

7) Сите се untracked

So git add одат во staged

So git commit одат во unmodified

Git reset -soft не се menuва nishto

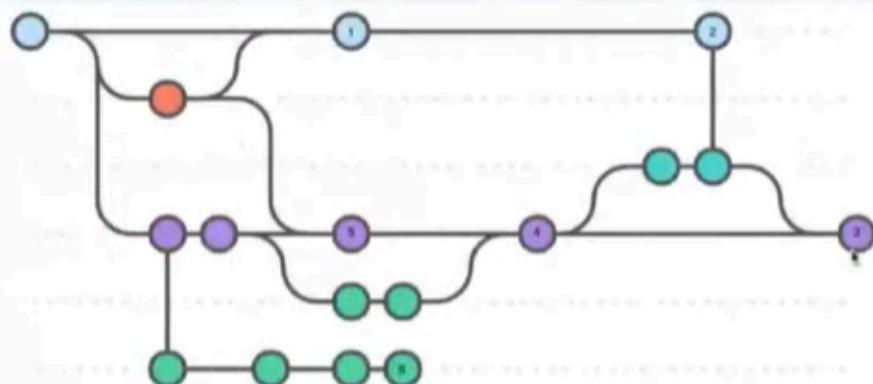
8) Se sho ne e na master odi vo temp

```
2345bcd Add README
2978ihg Compute correlation and calculate similarities
5678efg Extract embeddings
4c6f7e1 Read word pairs 2
5773afh Read word pairs 1
3456cde Create vocabulary
1234abc Initialize project

git rebase -i --root ✓

7 ✓ pick ✓ 2345bcd Add README
2 ✓ pick ✓ 3456cde Create vocabulary
1 ✓ reword ✓ 1234abc Initial commit
5 ✗ pick ✗ 7890ghi Compute correlation
6 Incorrect
The correct answer is: squash
4 Mark 0.00 out of 1.00
3 ✓ edit ✓ 4567def Read word pairs
```

На сликата е даден граф на commit-и во проект каде што се употребува Gitflow workflow. // A commits graph from a project where Gitflow is used is shown on the figure.



Кои од нумерираните commit-и не би постоеле на графот на commit-и доколку при спојување на гранките не бил употребен flag-от --no-ff? // Which of the numbered commits would not exist on the commits graph if the flag --no-ff was not used when merging the branches?

- 1
- 2
- 3
- 4
- 5
- 6

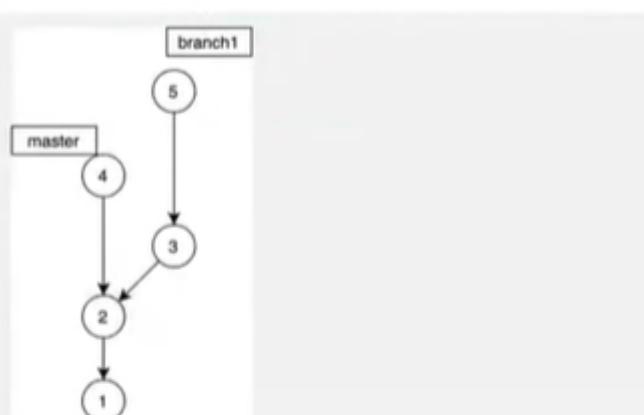
Во образложението објаснете за што точно се користи тој flag. // In the explanation explain what is the purpose of the flag.

Колку типа на гранки се користат во centralized workflow? // How many types of branches are used in the centralized workflow?

1

Во кои workflow-и мора да се користат pull/merge request? // In which of the workflows pull/merge requests must be used?

- centralized workflow
- feature branch workflow
- gitflow workflow
- forking workflow



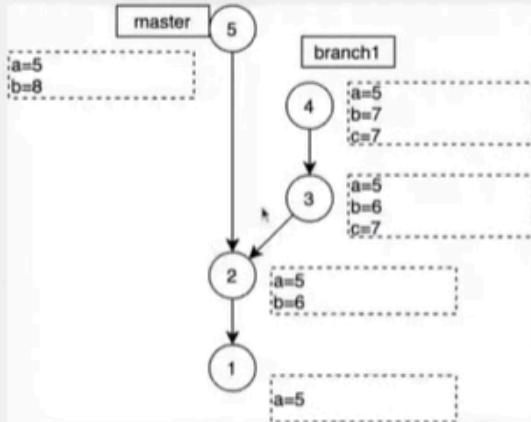
git checkout master

git merge branch1

Дали ќе се изршат успешно командите? Доколку да, по кога стратегија?

Select one:

- a. не, ќе се јави конфликт
- b. не, бидејќи не постои remote репозиториум
- c. да, fast-forward strategy
- d. да, recursive strategy



Дадена е граф од commit-и и до секој commit во кутијата со испрекинати линии е напишана содржината на датотеката numbers.txt. Во која линија од датотеката ќе се јави конфликт при спојување на гранките master и branch1 (доколку воопшто се појави конфликт)?

Select one:

- a. 1 линија (променливата a)
- b. нема да се јави конфликт, спојувањето ќе биде успешно.
- c. 3 линији (променливата c)
- d. 2 линија (променливата b)

По извршување на командата git status се добива следниот резултат:

On branch master

Your branch is up-to-date with 'origin/master'.

Untracked files:

(use "git add <file>..." to include in what will be committed)

README

Потоа се извршуваат командите

git stash save

git status

Кој ќе биде резултатот на последната наредба?

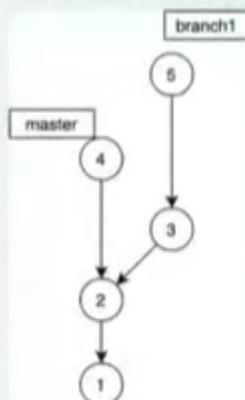
Select one:

- a. On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
(use "git add <file>..." to include in what will be committed)
README
- b. командата нема да се изврши, ќе се јави грешка
- c. On branch master
nothing to commit, working directory clean

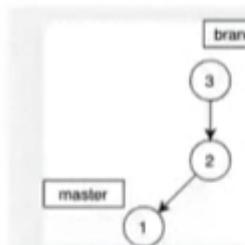
[Clear my choice](#)

Во кој редослед ќе бидат commit-ите на гранката branch1 доколку се извршат следните команди?
git checkout branch1
git rebase master

Напишете ги бројките споени без празни места. Пр. 123456789



Answer: 12435



git checkout master

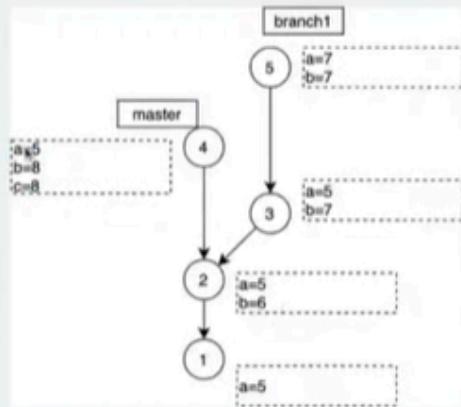
git merge branch1

Дали ќе се изршат успешно командите? Доколку да, по кога стратегија?

Select one:

- a. не, бидејќи не постои remote репозиториум
- b. не, ќе се јави конфликт
- c. да, recursive strategy
- d. да, fast-forward strategy

[Clear my choice](#)



Даден е граф од commit-и и до секој commit во кутијата со испрекинати линии е напишана содржината на датотеката numbers.txt. Во која линија од датотеката ќе се јави конфликт при спојување на гранките master и branch1 (доколку воопшто се појави конфликт)?

Select one:

- a. нема да се јави конфликт, спојувањето ќе биде успешно.
- b. 1 линија (променливата a)
- c. 2 линија (променливата b)
- d. 3 линија (променливата c)

[Clear my choice](#)

Activate Window

По извршување на командата git status се добива следниот резултат:

On branch master

Your branch is up-to-date with 'origin/master'.

Changes to be committed:

modified: file1.txt

Changes not staged for commit:

modified: file2.txt

Потоа се извршуваат команите

git stash save --keep-index

git status

Кој ќе биде резултатот на последната наредба?

Select one:

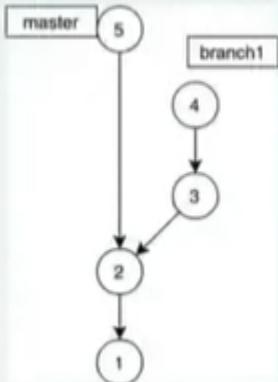
- a. On branch master
nothing to commit, working directory clean
- b. командата нема да се изврши, ќе се јави грешка
- c. On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
modified: file1.txt
Changes not staged for commit:
modified: file2.txt
- d. On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
modified: file1.txt

[Clear my choice](#)

Во кој редослед ќе бидат commit-ите на гранката master доколку се извршат следните команди?

```
git checkout master  
git rebase branch1
```

Напишете ги бројките споени без празни места. Пр. 123456789



Answer: 12345|

```
git commit -a -m "Add 1 to numbers.txt"  
git commit -a -m "Add 2 to numbers.txt"  
git checkout -b branch1  
git commit -a -m "Add 3 to numbers.txt"  
git checkout master  
git commit -a -m "Add 4 to numbers.txt"  
git push origin master  
git checkout branch1  
git checkout -b branch2  
git commit -a -m "Add 5 to numbers.txt"  
git push origin branch2  
git commit -a -m "Add 6 to numbers.txt"  
git checkout master  
git commit -a -m "Add 7 to numbers.txt"
```

Внесете ја содржината на фајлот во секоја од гранките локално и на ремоте репозиториумот. Доколку некоја од гранките не постои соодветно ставете /. Не оставяйте празни места, напишете ги само броевите споени. Пример. 1234567 // Write the content of the file in each of the branches locally and on the remote repository. If some of the branches does not exist write /. Don't leave spaces, just write the numbers. Per ex. 1234567.

Гранка (branch) /репозиториум (repo)	local	origin
master	1247	124
branch1	123	/
branch2	12356	1235

Командата git commit -a -m ги извршува: // The command git commit -a -m does:

Activate Windows

branch2

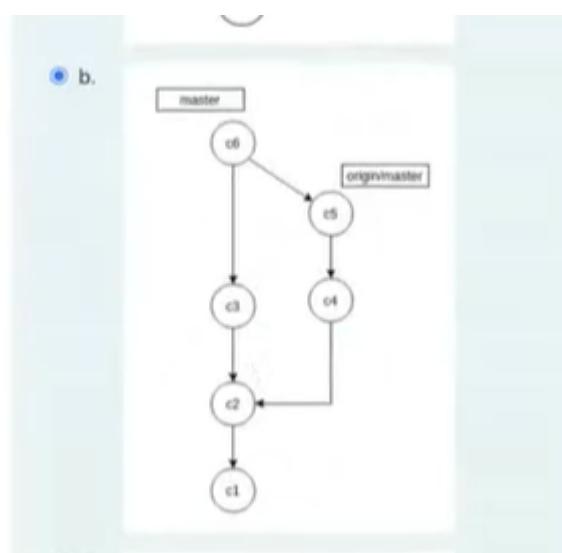
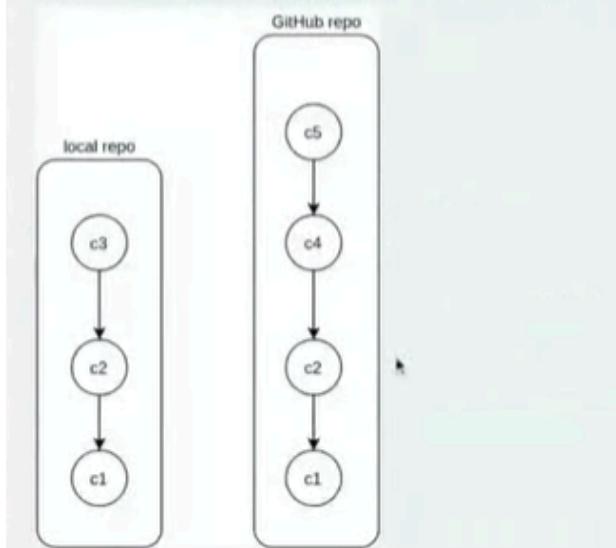
12356

1235

Командата git commit -a -m ги извршува: // The command git commit -a -m does:

- додавање на сите файлови од директориумот во staging area // addition of all files from the directory into the staging area
- додавање на сите tracked файлови во staging area // addition of all tracked files into the staging area
- изменување на последниот направен commit со новите промени / amending the last made commit with the newest changes
- воспоставување конекција со remote репозиториумот / establishing connection with the remote repository
- иницијализација на git репозиториумит / initialization of the git repository
- зачувување на верзија на кодот // saving a version of the code

Нека е дадена следната ситуација со commits (лево на локалниот репозиториум, десно на remote репозиториум). Која ќе биде состојбата на commit-ите во локалното репо по извршување на наредбата **git pull origin master**? // For the given situation with commits (on the left the local repo commits graph is shown, and on the right the remote repo commits graph is shown). What will be the state of the commits on the local repo after the command **git pull origin master** is executed?



Додека сте во главната гранка правите промена во класата и истата го добива следниот изглед. За промената соодветно правите commit: // While you are on the main branch you are making the following change. You've committed the corresponding change:

```
public class SISeptember {  
    int a = 6;  
    int b = 7;  
    int c;  
  
    public SISeptember(int a, int b) {  
        this.a = a;  
        this.b = b;  
        this.c = a+b;  
    }  
}
```

Изберете ја командата со која што ќе ја споите гранката numbers кон главната гранка. // Select the command that you'll use in order to merge the numbers branch into the main branch:

git merge numbers

Спојувањето на гранката numbers кон гранката master: // The merging of the numbers towards the master branch:

нема да резултира со конфликт//won't result with conflict.

Спојување Incorrect

with the st The correct answer is: нема да се изврши, ќе се јави конфликт//the merging won't be successful, conflict will emerge

Mark -0.75 out of 3.00

is the branch master will be done

Нека е даден проект Si со само една класа напишана во Java во него. По додавање на кодот (даден подолу) во оваа класа се прави commit. // Imagine that there is a project named Si with only one Java class in it. After the code (given below) is added to the class, the change is committed.

```
public class SISeptember {  
    int a = 6;  
    int b = 7;  
  
    public SISeptember(int a, int b) {  
        this.a = a;  
        this.b = b;  
    }  
}
```

Изберете ја командата со која што директориумот Si ќе го направите да биде локалниот git репозиториум: / Select the command that you'll use in order to make the directory Si a local Git repository:

git init

Изберете ја командата со која што локалниот репозиториум ќе го поврзете со remote репозиториумот што се наоѓа на линкот https://github.com/test/test_repo.git. // Select the command that you'll use to connect the local repo to a remote repo that's located on the link https://github.com/test/test_repo.git:

git remote add origin https://github.com/test/test_repo.git

Изберете ја командата со која ќе креирате и веднаш ќе се префрлите на гранката numbers // Select the command that you'll use to create and checkout immediately to the branch named numbers:

git checkout -b numbers

Откако ќе се префрлите на numbers ја менувате вредноста на променливата b во 10, правите git add и git commit. Изберете ја командата со која ќе се префрлите на главната гранка // After checking out to the branch numbers you've changed the value of the variable b to 10 and you've executed git add and git commit.

Select the command that you'll use to checkout to the main branch:

git checkout master

Додека сте во главната гранка правите промена во класата и истата го добива следниот изглед. За промената соодветно правите commit: // While you are on the main branch you are making the following change. You've committed the corresponding change:

public class SISeptember {