

git request-pull

- git-request-pull - Generates a summary of pending changes

```
git request-pull <start> <url> [<end>]
```

- Inizialmente git non era pensato per avere un ambiente di hosting centralizzato...
- l'interazione era stata pensata molto più *peer to peer*... o tramite vecchi canali quali ad esempio la posta.



Esempio

```
'git request-pull 5a548dc git@bitbucket.org:emanuelemazzola/p2-lab05-2019.git consegna_16:master
The following changes since commit 5a548dc21910dd1e7466323812282909c0e6fd2a:

  start (2019-11-13 08:22:34 +0100)

are available in the Git repository at:

  git@bitbucket.org:emanuelemazzola/p2-lab05-2019.git master

for you to fetch changes up to 7c2095152be13b030bb42804acf7d8bc7988b6ed:

  Risoluzione di alcuni errori: Codizione per la quale il mazziere gioca o meno (almeno un giocatore non deve aver sballato), errata condizione di gioco nella classe mazziere

-----
Emanuele Mazzola (1):
  README.md edited

emanuele.mazzola (5):
  Prima versione
    Prima versione tutti i file
    Seconda versione Distribuzione delle carte iniziali. Modifiche visualizzazione punteggi
    Nome errato
      Risoluzione di alcuni errori: Codizione per la quale il mazziere gioca o meno (almeno un giocatore non deve aver sballato), errata condizione di gioco nella classe maz

.idea/gradle.xml |  1 ++
README.md        |  4 +---
src/main/java/it/unimi/di/prog2/blackjack/BlackJack.java | 29 ++++++-----+
src/main/java/it/unimi/di/prog2/blackjack/Giocatore8J.java |  3 ++
src/main/java/it/unimi/di/prog2/blackjack/MaggioreStrategy.java | 24 ++++++-----+
src/main/java/it/unimi/di/prog2/blackjack/Mazziere.java     | 23 ++++++-----+
src/main/java/it/unimi/di/prog2/blackjack/MinoreStrategy.java | 23 ++++++-----+
src/main/java/it/unimi/di/prog2/blackjack/MultiMazzo.java   | 14 ++++++-----+
src/main/java/it/unimi/di/prog2/blackjack/RandomStrategy.java|  2 ++
src/main/java/it/unimi/di/prog2/blackjack/RischioStrategy.java| 27 ++++++-----+
src/main/java/it/unimi/di/prog2/blackjack/Sfidante.java      | 34 ++++++-----+
src/main/java/it/unimi/di/prog2/blackjack/Strategia.java    |  6 ++++++
12 files changed, 186 insertions(+), 18 deletions(-)
create mode 100644 src/main/java/it/unimi/di/prog2/blackjack/MaggioreStrategy.java
create mode 100644 src/main/java/it/unimi/di/prog2/blackjack/MinoreStrategy.java
create mode 100644 src/main/java/it/unimi/di/prog2/blackjack/RischioStrategy.java
```

Progetti Open Source

- Soffrono in particolar modo per questi limiti
- Gli ambienti di hosting hanno cercato soluzioni alternative:
 - inventandosi nuovi meccanismi
 - provando a “imporre” nuovi workflow (GitHub Flow, GitLab Flow, etc)



Fork

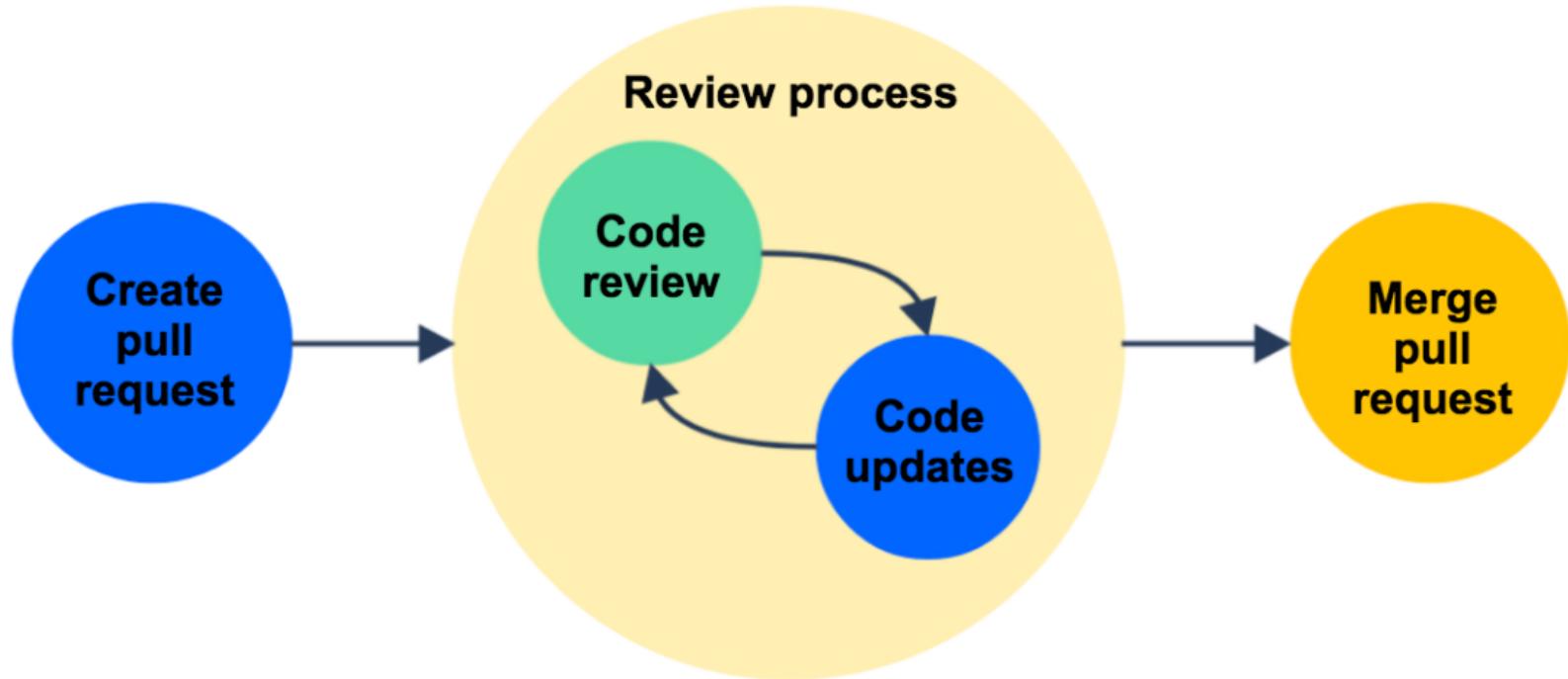
- Risolve un primo problema di autorizzazioni
 - permette di mantenere legami tra repository su sito di hosting ma con owner e autorizzazioni diversi
- Ottimizzazioni
 - condivisione dello spazio degli oggetti

[https://github.com/torvalds/linux
/blob/b4061a10fc29010a610ff2b5b20160d7335e69bf/drivers/hid/hid-samsung.c](https://github.com/torvalds/linux/blob/b4061a10fc29010a610ff2b5b20160d7335e69bf/drivers/hid/hid-samsung.c)

[https://github.com/torvalds/linux
/tree/8bcab0346d4fcf21b97046eb44db8cf37ddd6da0](https://github.com/torvalds/linux/tree/8bcab0346d4fcf21b97046eb44db8cf37ddd6da0)



Tra creazione e deploy va prevista una fase di review





(fork &)Pull requests

Permette di gestire interazioni lasche tra sviluppatori mediate dal sito di hosting

New Merge Request

Source branch
carlobelletti/bebras_local_editor Select source branch

Target branch
aladdin-bebras/bebras_local_editor master

totti console log di debug
Carlo Belletti authored 1 day ago a82d8d94

Open Opened 5 months ago by Matteo Zola Edit Mark as ready

WIP: Align Checkstyle rule #3 to PMD

1 unresolved thread Edit Mark as ready

Overview 4 Commits 9 Changes 43

10 Jun, 2020 9 commits

- Amend due to wrong merge 1b@e4b14
- Several fixes 70944ea6
- PRIMITIVES, parameter check, return type check, variable check a86e1c05
- PRIMITIVES, parameter check, return type check f2f673d7
- Fixed class fields like PMD 6c7ee707
- set of the new tests e8d841c8

<https://help.github.com/articles/what-is-a-good-git-workflow/>

verificaEconsalda > objectCalsterics > Merge Requests > t30

Open Opened 5 months ago by Matteo Zola Developer

WIP: Align Checkstyle rule #3 to PMD

Overview 4 Commits 9 Changes 43

Edit Mark as ready

1 unresolved thread

@Le5fa and I (@teozola) propose a merge request to align the Checkstyle version with PMD one to agree on rule 3 constrains given in #11.

We will port all the 34 test cases to ensure the same behavior for both Checkstyle and PMD.

Request to merge Le5fa:checkstyle into checkstyle

The source branch is 14 commits behind the target branch

Open in Web IDE Check out branch

Checking pipeline status.

Approve Approval is optional

Merge This merge request is still a draft. Mark as ready

Draft merge requests can't be merged.

You can merge this merge request manually using the command line

Oldest first Show all activity

Matteo Zola @teozola added 1 commit 5 months ago

• 6c7ee707 - Fixed class fields like PMD

Compare with previous version

Ann Boselli @Le5fa added 2 commits 5 months ago

• f2f673d7 - PRIMITIVES, parameter check, return type check
• a86e1c05 - PRIMITIVES, parameter check, return type check, variable check

Compare with previous version

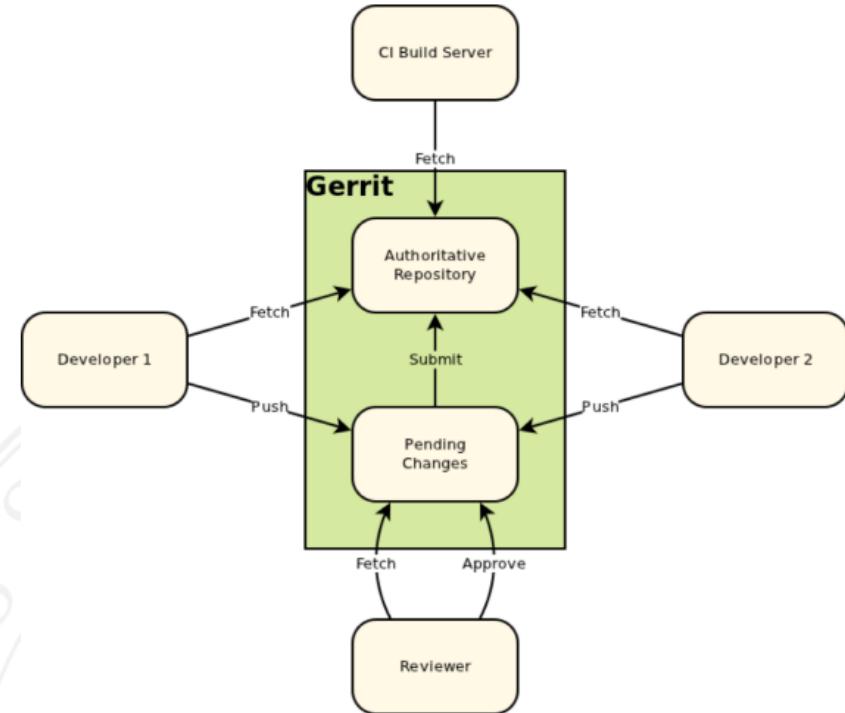
Matteo Zola @teozola added 1 commit 5 months ago

• 70944ea6 - Several fixes

Gerrit

<https://www.gerritcodereview.com/about.html>
<https://gerrit-review.googlesource.com/Documentation/>
<https://source.android.com/docs/setup/contribute/submit-patches>

- progetto di Google
 - AOSP
- peer review delle sottomissioni
 - In grandi progetti non si può pensare che ci sia una persona singola a decidere...
 - Gerrit è al tempo stesso un server Git (due server git) e un sito web di peer review.



Verifier: Verifying a change

- If you are assigned to be the Verifier for a change, you need to do the following:
 - Patch the change into your local client using one of the Download commands.
 - Build and test the change.

Within Gerrit use Publish Comments to mark the commit as "Verified" or "Fails," and add a message explaining what problems were identified.



Approver: Reviewing a change

If you are assigned to be the Approver for a change, you need to determine the following:

- Does this change fit within this project's stated purpose?
- Is this change valid within the project's existing architecture?
- Does this change introduce design flaws that will cause problems in the future?
- Does this change follow the best practices that have been established for this project?
- Is this change a good way to perform the described function?
- Does this change introduce any security or instability risks?

If you approve of the change, mark it with LGTM ("Looks Good to Me") within Gerrit.

Build automation

- Come proteggersi da checkin di una versione non funzionante?
 - Automatizzando
 - la ricompilazione
 - il testing
- Molti tool disponibili
 - make
 - Ant
 - Gradle



Make

- Comandi di shell
- dipendenza (tra file)

```
hellomake: hellomake.c hellofunc.c  
        gcc -o hellomake hellomake.c hellofunc.c -I.
```

```
CC=gcc  
CFLAGS=-I.
```

```
%.o: %.c $(DEPS)  
      $(CC) -c -o $@ $< $(CFLAGS)
```

```
hellomake: hellomake.o hellofunc.o  
          $(CC) -o hellomake hellomake.o hellofunc.o -I.
```

generazione dei makefile

- A monte della costruzione, c'è la definizione di come farlo
 - differenze per le varie macchine su cui si fa deployment
 - hanno la tale libreria? che versione?
 - hanno la tale funzione?
 - hanno...
- Automake, autoconf
- imake



Ant

- Ant nasce per supportare il progetto Tomcat
- Quando Tomcat entra in Jakarta viene capita l'utilità di Ant e viene reso indipendente (Luglio 2000)
- È scritto in Java per progetti Java
- Supporta CVS, Junit, FTP, JavaDOCS, JAR, etc...
 - Non solo compilazione, ma test e deploy



Concetti base di Ant

- Ogni progetto contiene più target :
 - compiling
 - deploying
- I target possono avere dipendenze da altri
- Targets contengono i task che fanno effettivamente il lavoro Si possono aggiungere nuovi tipi di task definendo (o trovando in giro) nuove classi Java



Un piccolo build file

```
<?xml version="1.0"?>
<project name="Hello" default="compile">
    <target name="clean" description="remove intermediate files">
        <delete dir="classes"/>
    </target>
    <target name="clobber" depends="clean" description="remove all artifact files">
        <delete file="hello.jar"/>
    </target>
    <target name="compile" description="compile the Java source code to class files">
        <mkdir dir="classes"/>
        <javac srcdir=". " destdir="classes"/>
    </target>
    <target name="jar" depends="compile" description="create a Jar file for the application">
        <jar destfile="hello.jar">
            <fileset dir="classes" includes="**/*.class"/>
            <manifest>
                <attribute name="Main-Class" value="HelloProgram"/>
            </manifest>
        </jar>
    </target>
</project>
```

Gradle

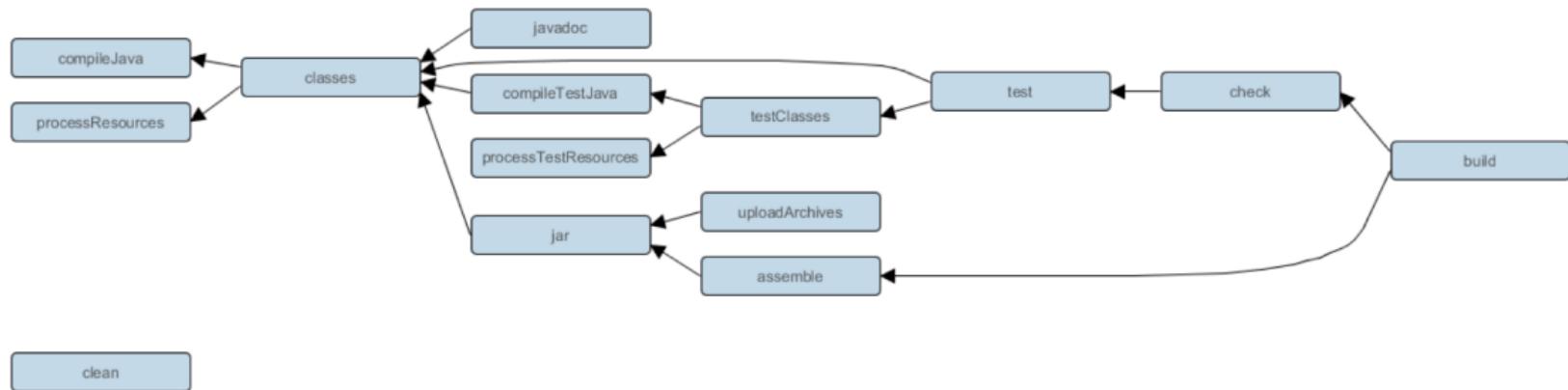
- Usa linguaggi Groovy (o Kotlin)
- Approccio dichiarativo e fortemente basato su convenzioni (*build-by-convention*)
- Definisce un linguaggio specifico per le dipendenze
- Multi-project builds
- Estendibile tramite plugin



plugin Java

- Definisce
 - sourceSet
 - task

Directory	Meaning
src/main/java	Production Java source
src/main/resources	Production resources
src/test/java	Test Java source
src/test/resources	Test resources



Altri plugin

- Application (task: run, startScripts)
- FindBugs (<http://findbugs.sourceforge.net/>)
- PMD (<http://pmd.sourceforge.net/>)
- checkstyle (<http://checkstyle.sourceforge.net/index.html>)
- jacoco (<http://www.eclemma.org/jacoco/>)
- eclipse
- idea

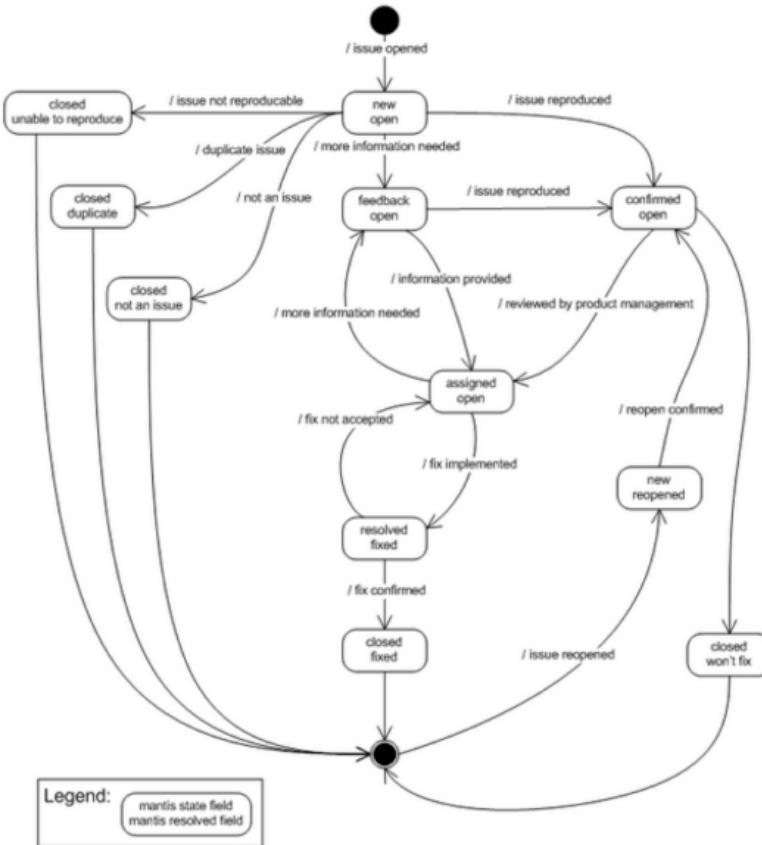
Bug tracking

- Tiene traccia e gestisce tutte le segnalazioni sui difetti di un software
 - Un database dei bug
 - Un mezzo di comunicazione per la segnalazione degli stessi
 - È uno strumento di assegnazione compiti
 - Alcuni tool:
 - BugZilla - <http://www.mozilla.org/bugs>
 - Scarab - <http://scarab.tigris.org/>
 - GNATS - <http://www.gnu.org/software/gnats>
 - BugManager - <http://www.bitmover.com>
 - Mantis - <https://www.mantisbt.org/>



Bug workflow

DAZ3D Issue Tracker Workflow



Unified Process

Altri nomi

- Unified Software Development Process (USDP)
- Rational Unified Process (RUP) [versione commerciale]

È il modello di processo definito dalle stesse persone che hanno definito UML

Non è l'unico modello che usa OO

- http://www.cetus-links.org/oo_ooa_ood_methods.html

Non è l'unico modello che usa UML

- Ad es. OOSP (Object Oriented Software Process)

Alcune risorse in rete:

- <http://www-01.ibm.com/software/awdtools/rmc/library/>
- <http://www.ambysoft.com/unifiedprocess/>



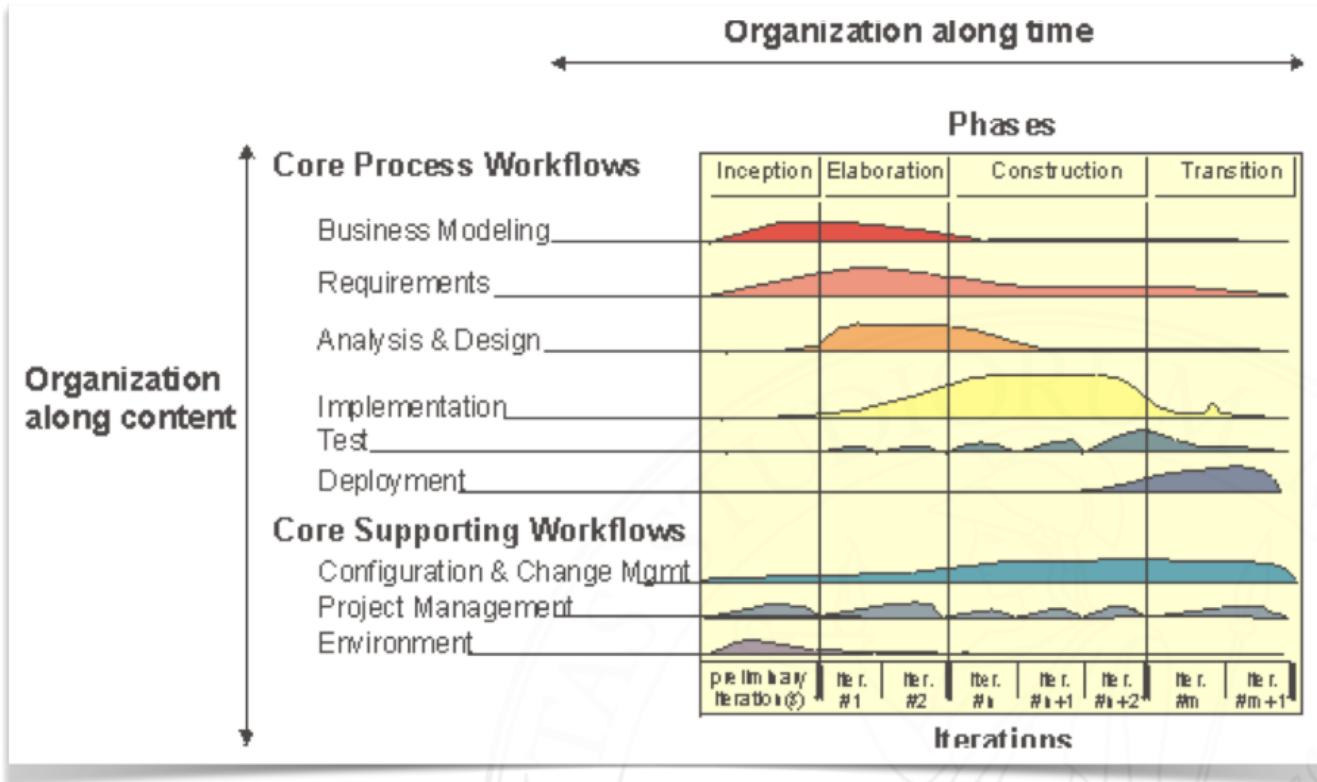
Parole chiave

È un processo che si definisce allo stesso tempo come:

- Sequenziale
 - Esistono 4 fasi svolte in sequenza
- Iterativo
 - Ogni fase è svolta in maniera iterativa In ogni iterazione si ripetono (con gradi diversi) le diverse attività (chiamate workflow o discipline) già viste
- Incrementale
 - Ogni Ciclo delle 4 fasi porta a una ‘release’



Vista generale





UNIVERSITÀ DEGLI STUDI
DI MILANO

Progettazione

Un programma in C

<http://mdnahas.github.io/xmassong/>
<https://youtu.be/YpuNU3y1KAK>

```
main(t,_,a ) char* a;{return!0<t?t<3?main(-79,-13,a+main(-87,
1-_ ,main(-86, 0,a+1 )+a)):1,t<_?main( t+1, _, a ):3,main(-94,
-27+t, a )&&t == 2 ?_<13 ? main ( 2, _+1,"%s %d %d\n" ):9:16:
t<0?t<-72?main( _, t,"@n'+,#'/*{}w+/w#cdnr/+,{}r/*de}+,/*{*+\n
,/w{%+,/w#q#n+,/#{l,+,/n{n+,/+#n+,/#;#q#n+,/+k#;*+,'r :'d*'\
3,}{w+K w'K:'+}e#';dq#'l q#+d'K#!/+k#;q#'r}eKK#}w'r}eKK{nl]\n
'/#;#q#n'){})#}w'){}){nl]'/+#n';d}rw' i;# ){nl]!/n{n#'; r{#w'r\
nc{nl]'/#{l,+'K {rw' iK{;[{nl]'/w#q#n'wk nw' iwk{KK{nl]!/w{\
%'l##w#' i; :{nl]'{q#'ld;r'}{nlwb!/*de}'c ;;{nl'-{}rw]'/+,\
}##'*}#nc,',#nw]'/+kd'+e}+;#'rdq#w! nr'/' ) }+}{rl#'{n' ')# \
}'+)##(!!"/):t<-50?_===*a?putchar(31[a]):main(-65,_,a+1):main
((*a == '/')+t, _,a+1):0<t?main ( 2, 2 , "%s") :*a=='/'||main
(0,main(-61,*a,"!ek;dc i@bK'(q)-[w]*%n+r3#l,{}:\nuuwloca-0;m\
.vpbks,fxntdCeghiry"),a+1);}
```

Cosa fa questo codice?

**OnlineGDB** beta

()

online compiler and debugger for
c/c++code. compile. run. debug.
share.

IDE ()

My Projects (/myfiles)

Classroom new

(/classroom)

Learn Programming <
<https://learn.onlinegdb.com>Programming Questions
<https://question.onlinegdb.com>

Blog (blog) · Terms of Use

Sign up / register

(/termsofuse) · Contact Us

Login (login)

(/contact) GDB Tutorial

[\(http://www.gdbtutorial.com/\) ·](http://www.gdbtutorial.com/)

Credits (/credits) · Privacy

(/privacy)

© 2016 - 2023 GDB Online



Run

Debug

Stop

Share

Save

Beautify



Language C

main.c

```
1 //*****  
2  
3 Online C Compiler.  
4 Code, Compile, Run and Debug C program online.  
5 Write your code in this editor and press "Run" button to compile and  
6 execute it.  
7 *****  
8  
9 #include <stdio.h>  
10  
11 int main()  
12 {  
13     printf("Hello World");  
14  
15     return 0;
```



Command line

input

Standard Input: Interactive Console Text

x (/upgrade-premium)

Altra versione

```
1  static String[] days = {"first", "second", ..., "twelfth"};
2  static String[] gifts = { "a partridge in a pear tree", "two turtle doves", ... };
3
4  static String firstLine(int day) {
5      return "On the " + days[day] +
6          " day of Christmas my true love gave to me:\n";
7  }
8
9  static String allGifts(int day) {
10     if( day == 0 ) { return "and " + gifts[0]; }
11     else { return gifts[day] + "\n" + allGifts(day-1); }
12 }
13
14 public static void main(String[] args) {
15     System.out.println(firstLine(0));
16     System.out.println(gifts[0]);
17     for (int day = 1; day < 12; day++) {
18         System.out.println(firstLine(day));
19         System.out.println(allGifts(day));
20     }
21 }
```

Refactoring

Improving the design of code without changing its functionality

1. Migliorare un design tenuto inizialmente "semplice" (o nato male)
2. Preparare il design per una funzionalità che non si integra bene in quello esistente
3. Eliminare debolezze (debito tecnico)

Design Knowledge

Dove la salviamo?

0. memoria
1. documenti di design (linguaggio naturale o diagrammi)
2. all'interno di piattaforme di discussione, issue management, version control
3. con modelli specializzati (UML). Può portare a approcci *generative programming*
4. nel codice, ma spesso difficile rappresentare le *ragioni*

Come condividiamo

Non un design specifico, ma il know how

- 0. metodi
- 1. design pattern
- 2. principi



check conoscenza di alcuni termini e concetti

- Object Orientation
 - Ereditarietà
 - Polimorfismo
 - Collegamento Dinamico
- Principi SOLID
 - Single Responsibility
 - Open Close Principle
 - Liskov Substitution Principle
 - Interface Segregation
 - Dependency Inversion