TEST

*“Code a little, test a little”*

Il test automatico abbassa i costi

I test sono scritti bottom-up

I test vanno accumulati, il controllo della correttezza deve essere automatico.

Sono solitamente facili da scrivere.

NON mischiare i test con il codice produttivo

Si usa un framework di test, JUnit nel nostro caso.

ESEMPIO

Immagine che contiene testo

Descrizione generata automaticamente

Classe di test

Immagine che contiene testo

Descrizione generata automaticamente

I metodi sono public void, devono avere l’annotazione @Test

Si crea una struttura di test testando casi limite.

Si usa assertEquals(…), o altri … di funzioni assert ci sono di diversi tip, i risultati sono verificati dal framework, primo parametro cosa ci si aspetta secondo è la funzione in ingresso.

I test vanno in una directory a parte rispetto al software effettivo

La signature del test non ha parametri

**@Before** e **@BeforeEach** sono dei metodi, le annotazioni consentono di creare metodi custom, che vengono chiamati prima del metodo di test