

Teme laborator Testarea sistemelor software

Cerințe

- Pentru realizarea unei teme se pot forma echipe de maxim 5 studenți. Este de preferat ca temele să fie realizate în echipă, dar nu obligatoriu.
Fiecare echipă va alege una dintre temele precizate mai jos. O temă poate fi aleasă de mai multe echipe.
- Prima pagină din documentație (fișier pdf/docx) trebuie să conțină componența echipei și tema aleasă. Un singur membru din echipă va trimite rezolvarea. Fiecare membru va primi același punctaj, indiferent de contribuția sa.
- Tehnoredactarea textului va fi făcută cu fontul Times New Roman cu caractere de 12 puncte tipografice (cu excepția secvențelor de cod sursă), având spațiere verticală la 1,5 rânduri și aliniere Justify.
- Secvențele de cod sursă vor fi integrate în document folosind o celulă (tabel cu o linie și o coloană). Nu se acceptă capturi de ecran ale codului.
Pentru textul din celulă se va folosi fontul Courier New/Consolas cu caractere de 11 puncte tipografice, având spațiere verticală la 1 rând ca în exemplul:
- Pentru fiecare cerință se va crea o secțiune cu un titlu scurt care va fi adăugat automat în Cuprins.

```
def test_sum():  
    assert sum([1, 2, 3]) == 6, "Should be 6"
```

- Grafurile vor fi realizate cu un utilitar (<https://app.diagrams.net>, <https://www.lucidchart.com/pages/>, yEd, Microsoft Visio et al.). Nu se acceptă imagini fotografiate/scanate.
- Pot fi utilizate alte referințe decât cele indicate: documentații oficiale ale tool-urilor, articole științifice și cărți (<https://dblp.uni-trier.de>, <https://scholar.google.ro>).
Articolele științifice și cărți care nu sunt free pot fi găsite la <https://sci-hub.st>, <http://libgen.st>
- Tema (documentația) va fi predată în **format electronic până la finalul semestrului (29 aprilie 2022, 23:59)**.
- **Link upload:** <https://forms.gle/ReDzxkv1eKGLrsPm6>

Tema 1: Search-based software testing

Se consideră următoarea problemă: date fiind un program și o cale în graful asociat programului, se urmărește generarea unor date de test ce execută această cale.

- Să se prezinte rezolvarea problemei folosind algoritmi genetici. Să se explice forma funcției de fitness folosite.
- Să se implementeze un algoritm care rezolvă problema de mai sus și să se illustreze funcționarea sa pentru un program dat. Să se comenteze rezultatele, sugerând îmbunătățiri.

Referințe:

Tema 2: Search-based software testing

- Să se prezinte metoda Chaining Approach pentru generare de date de test. Să se explice utilitatea metodei.
- Să se implementeze algoritmul și să se ilustreze funcționarea sa pentru un program dat.

Referințe:

Tema 3: Finite state machine based testing

- Să se prezinte metodele W și Wp de generare de teste pe baza unei specificații de forma unui FSM (Finite State Machine). Să se descrie condițiile de aplicare a metodelor. Să se explice, pe un exemplu, aplicarea metodei pentru diagrama de stare a unei clase.
- Să se implementeze metoda W.

Referințe:

Tema 4: Finite state machine based testing

- Să se prezinte conceptul de secvență UIO (Unique Input/Output) și relevanța acestuia pentru testarea de conformanță. Să se prezinte modul de construcție a unei secvențe UIO folosind un State Splitting Tree.
- Să se prezinte modul de construcție a unei secvențe UIO folosind un algoritm genetic.
- Să se implementeze algoritmul genetic și să se evalueze funcționarea acestuia.

Referințe:

Tema 5: Code coverage and mutation testing

- Să se realizeze un studiu comparativ a cel puțin 3 code coverage tools, evidențiindu-se utilitatea și ușurința în folosire a fiecăruia. Pe baza unor exemple de cod, se vor evidenția diferențele dintre tool-uri.
- Să se realizeze un studiu comparativ a cel puțin 3 mutation tools, evidențiindu-se utilitatea și ușurința în folosire a fiecăruia. Pe baza unor exemple de cod, se vor compara operatorii de mutație ai fiecăruia, evidențiindu-se utilitatea acestor operatori pentru evaluarea testelor și detectarea erorilor.
- Să se evidențieze, folosind exemple, legătura dintre code coverage și mutation testing.

Notă: nu se ia în considerare traducerea instrucțiunilor de utilizare a utilităților respective.

Referințe:

Tema 6: Model based Testing with GraphWalker

- Prezentați utilitarul de generare de teste GraphWalker din modele de formă (Extended) Finite State Machines.
- Prezentați generatoarele și criteriile de oprire (stop criteria) disponibile. Ilustrați cu exemple adecvate.
- Prezentați și ilustrați cu exemple generarea online de teste (online test sequence generation).
- Utilitarele și conceptele prezentate vor fi ilustrate pe exemple proprii (create de echipă).

Referințe:

Tema 7: Test Driven Development in C#

- Prezentați această abordare a dezvoltării de software ce combină Test First Development (TFD) cu refactorizarea.
- Conceptele prezentate vor fi ilustrate pe exemple proprii (create de echipă).

Referințe:

Tema 8: Unit Testing with C#

- Prezentați facilitățile unui tool pentru testarea unitară în C#.
- Ilustrați strategiile de generare de teste pe exemple proprii (create de echipă).

Referințe:

Tema 9: Unit Testing with with Python

- Prezentați facilitățile unui tool pentru testarea unitară în Python.
- Ilustrați strategiile de generare de teste pe exemple proprii (create de echipă).

Referințe:

Tema 10: Automated testing in web applications

- Prezentați facilitățile unui tool pentru testarea automată a paginilor web.
- Ilustrați strategiile de generare de teste pe exemple proprii (create de echipă).

Referințe:

Tema 11: Automated testing in mobile applications

- Prezențați facilitățile unui tool pentru testarea automată a aplicațiilor mobile.
- Ilustrați strategiile de generare de teste pe exemple proprii (create de echipă).

Referințe:

Tema 12: Mutation testing of Android Apps

- Prezențați tool-urile de analiză a mutanților pentru aplicații Android și evidențiați utilitatea operatorilor de mutație pentru evaluarea testelor și detectarea erorilor pe baza unor exemple de cod proprii (create de echipă).

Referințe:

Tema 13: Integration testing

- Prezențați tipurile de teste de integrare pe care le utilizează în mod regulat dezvoltatorii.
- Ilustrați strategiile de generare de teste pe exemple proprii (create de echipă).

Referințe:

Tema 14: Model based testing with Rodin Platform

- Prezențați facilitățile platformei Rodin pentru modelarea și verificarea consistenței modelelor discrete și hibride.
- Să se prezinte modul de verificare al unor proprietăți de safety cu ajutorul model checker-ului ProB pe baza unui exemplu dat de sistem discret precum și al unui hibrid. Să se comenteze rezultatele.

Referințe:

Tema 15: Model based testing with Rodin Platform

- Prezențați facilitățile, dar și neajunsurile platformei Rodin pentru modelarea și verificarea consistenței modelelor hibride și necesitatea cosimulării.
- Să se prezinte modul de validare al unui sistem hibrid pe baza unui exemplu dat cu ajutorul plug-in-ului Multisim. Să se comenteze rezultatele.

Referințe: