



SOFTWARE  
упражнения

# Осигуряване на качество на софтуера (Q.A.)

доц. ас. д-р Георги Шарков

гл. ас. д-р Мая Стоева





# Здравейте!

**доц. ас. д-р Георги Шарков**

**gesha@esicenter.bg**

**гл. ас. д-р Мая Стоева**

**mstoeva@uni-plovdiv.bg**

# Съдържание

1. Инструменти за тестване на софтуера
2. “Card Sorting” пример
3. Тестване с <https://www.browserstack.com>





**1**

# **Инструменти за тестване на софтуера**

# Инструменти тестване

1. Helium (<https://github.com/mherrmann/selenium-python-helium>)  
<https://testingbot.com/support/getting-started/helium.html>

Helium е инструмент за автоматизирано тестване на уебсайтове в браузъри. Той предлага прости команди като `start_chrome`, кликване и запис. Helium ни предоставя възможност да използваме команди точно, както бихме го направили, когато даваме инструкции на някой, гледащ през рамо към екрана - единствената разлика е, че ги въвеждаме. След като зададем няколко команди, можем да ги запишем във файл, който да бъде възпроизведен с едно натискане на бутон, в определен интервал от време или да се задейства от инструмент за управление на изграждане/тестване.



# Инструменти тестване

## 1. Helium (<http://heliumhq.com/>)

README.md

### Selenium-python but lighter: Helium

Selenium-python is great for web automation. Helium makes it easier to use. For example:

```
Python 3.7.3 (default, Dec 20 2019, 18:57:59)
>>> from helium import *
```

Under the hood, Helium forwards each call to Selenium. The difference is that Helium's API is much more high-level. In Selenium, you need to use HTML IDs, XPaths and CSS selectors to identify web page elements. Helium on the other hand lets you refer to elements by user-visible labels. As a result, Helium scripts are typically 30-50% shorter than similar Selenium scripts. What's more, they are easier to read and more stable with respect to changes in the underlying web page.

Because Helium is simply a wrapper around Selenium, you can freely mix the two libraries. For example:

```
# A Helium function:
driver = start_chrome()
# A Selenium API:
driver.execute_script("alert('HI!');")
```

# Инструменти тестване

## 2. Loadfocus (<https://loadfocus.com/>)

Това е cloud-базиран инструмент за stress тестване на за сайтове, уеб app-ове и API-та.



# Инструменти тестване

## 3. Ghost Inspector (<https://ghostinspector.com/>)

Това е инструмент, специализиран в automated UI testing. Тестерите му дават много добри оценки и са доволни от него. Съществува като extension или addon към Chrome-а и е много удобен и лесен за употреба. Налични са доста примери за работа с него.

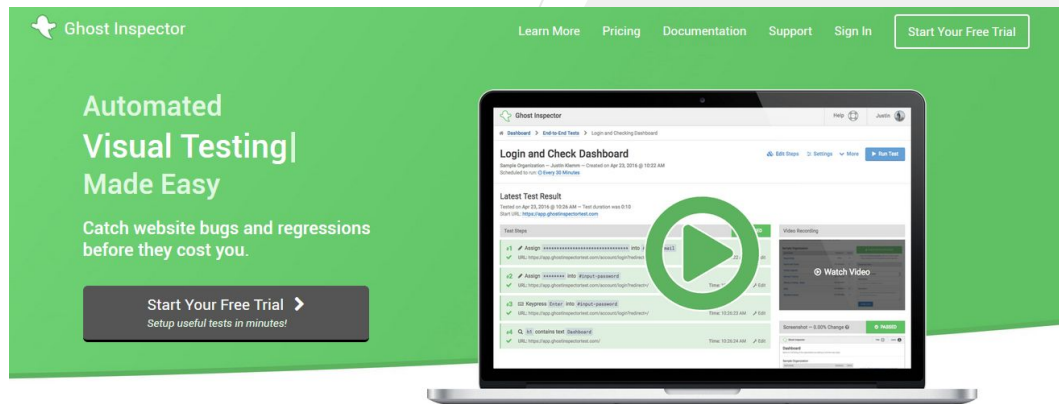
Предоставя безплатна версия.





# Инструменти тестване

## 3. Ghost Inspector (<https://ghostinspector.com/>)



✓ Test Recorder & Codeless Editor

✓ Video & Screenshot Comparison

✓ Multiple Browsers & Screen Sizes

✓ API & Third Party Integrations

✓ Advanced Test Scheduling

✓ Parallel Testing by Default

Empower your entire team to start testing.

Improve Efficiency. Deliver Faster. Ensure Quality.

# Инструменти тестване

## 4. JUnit (<https://junit.org/junit5/>)

Предоставя regression тестване на Java (напр. проверка на стари функционалности след update на software...). Той се използва за писане и изпълнение на unit тестове (юнит тестове), които помагат на разработчиците да гарантират, че техният код работи правилно.



# Инструменти тестване

## 5. Селениум (<https://selenium.dev/>)

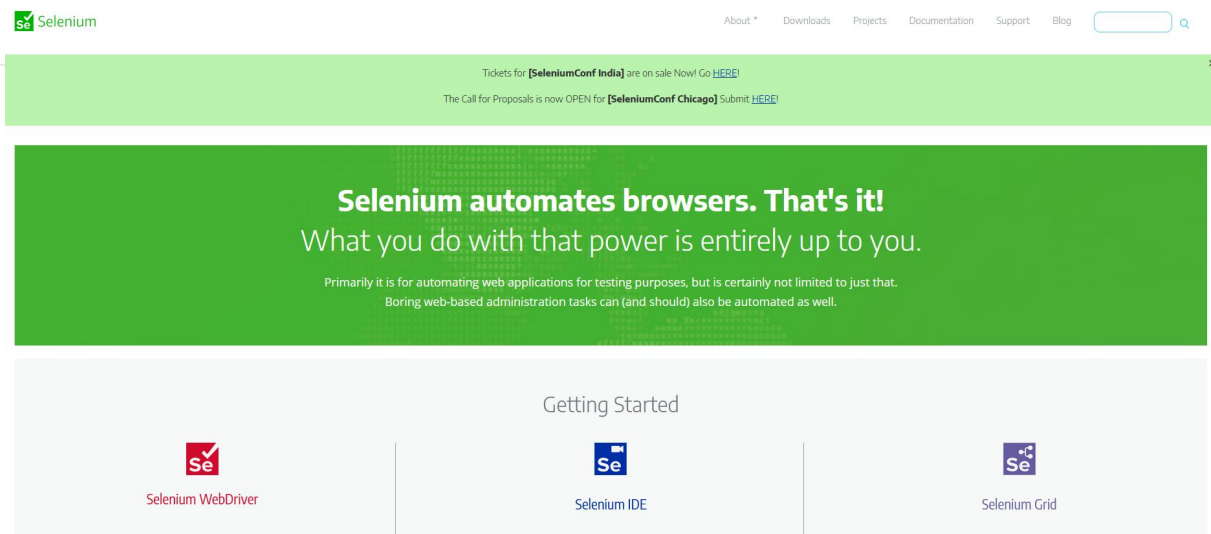
Предоставя група инструменти с възможност за поддръжка на автоматизирани тестове. Последните включват множество функции за тестване на различни видове уеб приложения.

Позволява сравняването на резултата от теста с настоящото поведение на приложението. Selenium има възможност за изпълнение на тестовете върху множество браузър платформи. Притежава собствен специфичен език (Selenese), на който могат да се пишат тестове на програмни езици като Java, C#, Groovy, Perl, PHP, Python и Ruby. Той работи върху Windows, Linux и Macintosh OS. Освен това е open-source софтуер и може да се използва безплатно.




# Инструменти тестване

## 5. Селениум (<https://selenium.dev/>)



The screenshot shows the Selenium website homepage. At the top, there is a navigation bar with links for 'About', 'Downloads', 'Projects', 'Documentation', 'Support', and 'Blog', along with a search bar. Below the navigation bar, a green banner contains text about SeleniumConf India and Chicago. The main content area features a large green box with the text 'Selenium automates browsers. That's it! What you do with that power is entirely up to you.' and a subtext explaining its primary use for automating web applications for testing purposes. At the bottom, there is a 'Getting Started' section with three columns: 'Selenium WebDriver', 'Selenium IDE', and 'Selenium Grid', each with its respective logo.

 Selenium

About \* Downloads Projects Documentation Support Blog


Tickets for **SeleniumConf India** are on sale Now! Go [HERE](#)


The Call for Proposals is now OPEN for **SeleniumConf Chicago** Submit [HERE](#)


**Selenium automates browsers. That's it!**  
What you do with that power is entirely up to you.

Primarily it is for automating web applications for testing purposes, but is certainly not limited to just that.  
Boring web-based administration tasks can (and should) also be automated as well.

Getting Started

 Selenium WebDriver

 Selenium IDE

 Selenium Grid

# Инструменти тестване

## 6. Browserstack (<https://www.browserstack.com/responsive>)

Една отлична облачна платформа за паралелно тестване на уеб апликации на десктоп и мобилните устройства, която дава възможност на разработчиците да проверят своите уеб сайтове и мобилни приложения в браузъри, операционни системи и реални мобилни устройства, без да се изисква от потребителите да инсталират или поддържат физически. Той включва четири основни продукта - Live, App Live, Automate и App Automate.



Live

Features

Mobile Features

Browsers &amp; Devices

# World's leading cross browser testing tool

Interactive web-based testing on 2000+ browsers and real devices instantly. Say goodbye to your lab of devices and virtual machines.

[Get started free](#)

Test your websites

**Live**

Interactive cross browser testing

**Automate**

Selenium testing at scale

Test your mobile apps

**App Live**

Interactive native &amp; hybrid app testing

**App Automate**

Test automation for native &amp; hybrid mobile apps

Use BrowserStack with your favourite products. See our Integrations →

For Teams

Enterprise

Tools

Screenshots

Responsive

Responsive

# Responsive Design Testing on Real Devices

Check website responsiveness on multiple iOS & Android devices

 [Invite your team](#) to join BrowserStack and test collaboratively. ✕

<https://mw-medianetworks.eu/>

Check

[Enable Local Testing](#) ▾

[Learn more about local testing](#)



**iPhone X**  
375x812  
🍏 11.0



**Galaxy Note 10**  
412x869  
🌐 9.0



**iPhone 8 Plus**  
414x736  
🍏 12.0



**Galaxy S9 Plus**  
412x740  
🌐 8.0



**iPhone SE**  
320x568  
🍏 11.0



**Pixel 3 XL**  
412x846  
🌐 9.0



**iPad Mini 2019**  
768x1024  
🍏 12.0



**Galaxy S7**  
360x640  
🌐 6.0



**iPad Pro**  
1024x1366  
🍏 12.0



**Kindle Fire HDX 7**  
600x960  
🌐 4.3



**Galaxy Tab S4**  
1134x712  
🌐 8.1



**Windows**  
1280x1024  
# 7



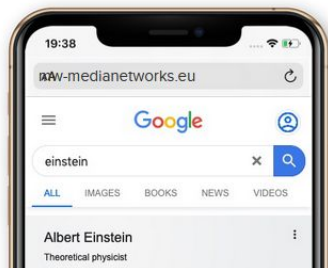
**Mac OSX**  
1920x1080  
🍏 Yosemite

Try Live Interactive Testing on 2000+ real browsers and devices. [Get Started Free](#)

◀ **iPhone X** Galaxy Note 10 ▶

Viewport: **375x812** | Screen size: **5.8"** | Monitor size: **15.4"**

📱 **Portrait** 🖼️ Landscape 🖼️



# Най-добри инструменти за автоматизирано тестване за 2024/25 година

Selenium

Postman

Tricentis NeoLoad

ZAPTEST

JUnit

<https://www.guru99.com/bg/best-it-automation-tools.html>



# Още няколко инструмента за софтуерното тестване

<https://www.guru99.com/bg/top-6-api-testing-tool.html>

The image features a large white circle centered on a black background. To the left of the white circle, there is a series of overlapping gray circles of varying shades, with the number '2' in white. To the right of the white circle, there is a series of concentric white circles. The overall design is minimalist and geometric.

2

**Въпросчета...**

**Имате 100 бѹга за 10 дни, но в даден момент ви казват, че имате 5 дни, какво ще направите?**

*Верен отговор би бил:* ще приоритизират най-важните бѹгове.

**Кои test cases бихте автоматизирали – критичните или тези, които се изпълняват най-често?**

*Верен отговор би бил:* Тези, които се изпълняват най-често, за да ни спестят време, а за критичните отделяме време и внимание!

**Каква е разликата между Re-testing и Regression testing?**

*Верен отговор би бил:* re-testing – след като се открие бѹг, го поправяме и проверяваме отново, за премахнат. Тук трябва да се повтори същият test case, при който е намерен бѹга.

**Regression testing** – проверяваме дали след като сме открили и отстранили бѹга, не сме променили друга функционалност. Това най-често са автоматизирани тестове. Те трябва да са документирани и добре описани. Имаме четири нива на тестване: тестване за дефекти, тестване на променени пътища, тест на нова функционалност и накрая – пълно тестване.

**Кога е удачна автоматизацията?**

*Верен отговор би бил:* При regression testing.

**Кой е най-интересния бѹг, който сте откривали?**

Разкажете какво сте открили, какво бихте променили и как бихте установили дали бѹга е отстранен.

### Какво точно представя **severity** (тежест) и **priority** (приоритет) на дефекта/бъга?

*Верен отговор би бил:* Severity показва какво е въздействието на нашия бърк върху нашата система.

### Какво тестваме при **Black Box Testing** и **White Box Testing**?

*Верен отговор би бил:* При Black Box Testing-а тестваме само интерфейса на софтуера, без познаваме неговата Функционалност, а при White Box Testing проверяваме качеството на кода, без без да ни интересува интерфейса.

### Какво съдържа един тест план?

*Верен отговор би бил:* **test scope** – дефинира какво ще се тества в дадения период и кои точно модули.

**test Objectives** – дефинициите, използвани тест плана като бързина на сайт, тестовите резултати, които се очаква да бъдат се покриват и други.

**assumptions** – какво се очаква от тестерите, бюджет за тестване, начална и крайна дата и инструменти;

**risk Analysis** – дефинират се по-важни неща, които може да провалят тестването;

**test design** – какви са етапите на тестване, каква е последователността, каква е времевата рамка;

**roles & Responsibilities** – кой какво прави и за колко време го прави, приоритети върху задачите;

**test data management** – събират се проекти, които да се обновяват;

**test environment** – с каква среда разполагаме, с каква машина/сървър разполагаме, къде ще тестваме;


**communication approach** – кога да има срещи с екипа и т. Н.;

**test tools** – какви инструменти ще се използват за различните дейности.

# Литература

<http://forum.dev.bg/index.php/220/%D0%BA%D0%BE%D1%8F-bug-tracking-software-%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0-%D0%B8%D0%B7%D0%BF%D0%BE%D0%BB%D0%B7%D0%B2%D0%B0%D1%82%D0%B5-%D0%B8-%D0%B7%D0%B0%D1%89%D0%BE>

<https://kvasilev.com/sqa/test-plan-za-software/>

A circular frame containing a photograph of a desk. On the desk, there is a potted plant in a white pot with a lace-like top, a large black letter 'A', and a wooden block with the letter 'S'. The background shows a window with light coming through. The text "Благодаря за Вашето внимание!" is overlaid on the bottom right of the circle.

**Благодаря за  
Вашето внимание!**