Last Updated on 2nd September 2021

NataliiaStulova

contact

nata.stulova@pm.me

links

web:// **s0nata.github.io**
LinkedIn:// **nata-stulova**

education

PhD in Software, Systems and Computing cum laude
2014–2018
Technical University of Madrid (UPM)

MSc in Artificial Intelligence
2012–2013
Technical University of Madrid (UPM)

BSc in Systems Analysis
2008–2012
National Technical University of Ukraine “Kyiv Polytechnic Institute” (NTUU “KPI”)

programming

Java, C++
Prolog
LaTeX
bash

languages

native
Ukrainian
Russian
proficient
English
Spanish
intermediate
German
beginner
French
Hebrew

experience

2020–now **Senior Researcher |** *University of Bern* Switzerland
My main research focus is software documentation: its quality analysis and developing tools for improving it automatically. I am also working on techniques for supporting requirement engineering tasks in software development.

2019–2020 **Scientist |** *Swiss Federal Institute of Technology in Lausanne (EPFL)* Switzerland
I have worked on formal languages-based techniques for source code automatic documentation and summarization, and explored NLP-based approaches to software documentation analysis.

2014–2018 **Research Assistant |** *IMDEA Software Institute* Spain
My research focus has been specification-based software verification: how to write formal specifications of program behavior, how to introduce non-trivial properties of programs, how to check them at run time, and how to do this efficiently.

2012–2013 **Research Intern |** *Technical University of Madrid (UPM)* Spain
I have designed and developed the initial graphical user interface (GUI) for a multi-agent airspace simulation system, prepared demos and documentaiton.

research

SCAM’21 Do Comments follow Commenting Conventions? A Case Study in Java and Python
code comments analysis • software documentation • coding style guidelines • coding standards

JSS’21 RepliComment: Identifying Clones in Code Comments
code comments analysis • natural language processing • code clones • code smells

LowCode’21 Interactive Behavior-driven Development: a Low-code Perspective
behavior-driven development • acceptance testing • low-code • no-code

SCAM’20 Towards Detecting Inconsistent Comments in Java Source Code Automatically
software documentation • natural language processing • software quality

PPDP’18 Static Performance Guarantees for Programs with Run-time Checks
resource usage analysis • assertions • declarative programming • run-time verification

PADL’18 Exploiting Term Hiding to Reduce Run-time Checking Overhead
declarative programming • module systems • assertions • abstract interpretation • run-time checking

SCP’17 Some Trade-offs in Reducing the Overhead of Assertion Run-time Checks via Static Analysis
abstract interpretation • assertions • run-time checking • logic programming • horn clauses

ICLP’15 Practical Run-time Checking via Unobtrusive Property Caching
assertions • property caching • memoization • run-time checking

PPDP’14 Assertion-based Debugging of Higher-Order (C)LP Programs
higher-order • assertions • run-time checking • declarative programming

other qualifications

2020–2021 **Lecturer** online and in-person teaching • lecture video recording • course design
Taught the Java Crash Course and Algorithms and Data Structures parts of the Software Skills Lab, a 5 ECTS course of the Joint Master in Computer Science program of the BeNeFri universities.

2020–now **Development Volunteer | Webmaster** requirements gathering • no-code web development
Consulting a Dnipro-based cultural space KsiProstir on website development and maintenance.

2017 **Workshop Chair | Organizer** web presence • talk scheduling • submission review
Co-organized CICLOPS’17 – 15th International Colloquium on Implementation of Constraint and LOgic Programming Systems.