Mihai – Alexandru Ciorobea

|  |
| --- |
| mihai.ciorobea@gmail.com ∙ 1Mai Street, Bucharest, Romania ∙ 0040730432772 |

**EDUCATION**

|  |
| --- |
|  |

**Polytechnic University of Bucharest** 2013-present

* Pursuing a Master’s degree in Computer Science (1st year)
* Expected graduation year: 2015

**Polytechnic University of Bucharest** 2009-2013

* Obtained a Bachelor’s degree in Computer Science
* Final GPA: 9.24 out of 10
* Final thesis: 10 out of 10

**WORK EXPERIENCE**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| **Adobe**  Bucharest, Romania  May 2013 – present | **Software Developer**  I am currently working at Digital Publishing Suite. In the analytics zone and on a search product.  **Used:**   * Java & Spring * Jax-RS & Jersey * Maven for building * Amazon Web Service for infrastructure * Git & Jenkins * CoralUI & AngularJs |
| **Teacher Assistant**  Polytechnic University of Bucharest  February 2013 – May 2013 | **Algorithms and Design Analysis**  I was a teacher assistant, where I had to teach the basic algorithms for Data and Design analysis. |
| **Adobe**  Bucharest, Romania  October 2011 – May 2013 | **Co-op Student**  I worked in Business Catalyst team. I worked from server monitoring to UI components.  **Used:**   * C# for the server side * IIS for web server * Cacti for the monitor script |
| **Adobe**  Bucharest, Romania  July – October 2011 | **Software Engineer Intern**  I worked at Adobe Translation Center. I wrote code for both the frontend and the backend.  **Used:**   * Java (Maven project) for the server side * GWT for the frontend part |

**PERSONAL PROJECTS**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| Intelligent Code Generation 2013 | I have implemented a solution for identifying and extracting structural patterns. This project is used for a big software solution where patterns can be easily found.  **Used:**   * C# * Partial compiler for C#   AI and Pattern Detection for extracting the template |
| **COMPILATOR**  2013 | I wrote an compiler with all the 4 components   * Parser * Lexer * Optimizations * Code generation |
| **Quantum tic-tac-toe**  2013 | Ai engine for Quantum tic-tac-toe player using reward-learning strategies.  **Used:**   * Java * Reward-Learning algorithms |
| **Auction application**  2013 | This was a client-server application for billing and auctions.  **Used:**   * Java * Swing |
| AI for car driving 2011 | I have implemented an AI engine together with three classmates. The team won the First-Place of the Faculty Engine Championship.  **Used:**  Java, algorithms and data structures (BFS, Bresenham’s line algorithm, simulating the track) |