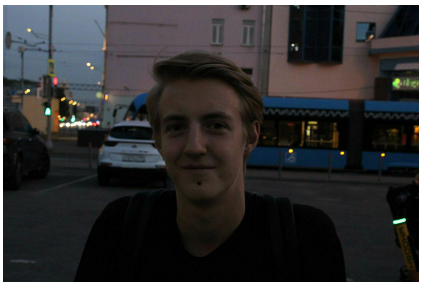
Vladimir Chursin



☎ +7(968)-951-70-38
✉ vladimirchursin18@gmail.com
✉ chursin.vv@phystech.edu
Amlola
@amlolaalready

Education

- **MIPT, Department of Radio Engineering and Computer Science** *2023 – Present*
  Applied Mathematics and Physics, 2nd year, GPA: 7.52
- **Introduction to Tensor Compilers course** *2025 – Present*
  Taken during 2nd year at MIPT
- **System Programming and Compiler Technologies course** *2023 – 2024*
  Completed during 1st year at MIPT
- **School 1501** *2021 – 2023*
  Grades 10-11, IT class, graduated with honors

Experience

- **Baikal Electronics (AI team)** *July - August 2024*
  Worked with internals of modern AI libraries.
  Developed high-performance C/C++ libraries. Wrote tests. Configured CI

Mentorship

- **MIPT, I.R. Dedinsky's course** *August 2024 - Present*
  Mentored three freshmen as a tutor. Conducted code reviews

Projects

- **Shift-Reduce-parser | https://github.com/Amlola/Shift-Reduce-parser** *March 2025*
  Toolset: C++, Flex, CMake, CI, Graphviz, GTest, dot, git
  – Built DFA
  – Tokenization using Flex
  – Implemented detailed algorithm logging and parse tree visualization
- **Language | https://github.com/Amlola/Language** *May 2024*
  Toolset: C/C++, x86_64 assembly, Graphviz, Make, dot, git
  – Developed AST standard
  – Implemented recursive descent and cross-translation with other languages
  – Created MiddleEnd with machine-independent optimizations (constant folding, math expression simplification)
  – Implemented IR for command optimization and translation to NASM
- **Hash Table | https://github.com/Amlola/HashTable** *April 2024*
  Toolset: C/C++, python, x86_64 assembly, Kcachegrind, objdump, Make, git
  – Implemented Hash Table data structure
  – Compared different hash functions by element distribution uniformity
  – Profiled the program and optimized element search function using minimal platform-dependent code
  – Used assembly and AVX instructions
- **MandelbrotSet | https://github.com/Amlola/MandelbrotSet** *April 2024*
  Toolset: C/C++, python, SFML, Make, git
  – Used SFML library for set visualization
  – Compared different rendering algorithms, one using SIMD instructions
  – Evaluated time measurement accuracy under CPU load
  – Researched timing precision using SFML library

Skills

Languages: C, C++, Python, x86_64 assembly, CUDA
Tools: git, CMake, Make, bash, LaTeX, Kcachegrind, perf, gdb, Matplotlib, CI/CD, objdump, readelf, Graphviz, dot

About Me

- **Soft skills:** teamwork, ability to work under pressure, time management, quick inclusion in the work, flexibility, perseverance, attention to details, accountability,
transparency, empathy and emotional intelligence
- **Hobbies:** Basketball and mountain hiking
- **Languages:** English - intermediate, Russian - native