

Puchkov Kyryll

+7 985 699 45 77 | puchkov.k@phystech.edu | Moscow, Russia | Oct 17, 2000 | github:puchkovki

EDUCATION:

2015 — 2017	The Richelieu Lyceum	Advanced mathematics and physics
2017 — now	Moscow Institute of Physics and Technology	Applied mathematics and physics
2019 — now	Department of applied and theoretical informatics (Acronis company)	Applied and theoretical informatics



CAREER OBJECTIVE: To get a position where I will be able to obtain and utilize my experience, develop my technical and social skills.

QUALIFICATIONS SUMMARY:

- Have programming experience on C/C++, Golang, basic skills with Python, MySQL;
- Have experience of writing of research works on physics, mathematics, IT;
- Have good analytical/problem-solving skills, self-initiative and fast to learn new skills/technologies;
- Able to communicate in Russian (native), English (upper-intermediate) and German (pre-intermediate);
- Have 5 years' experience in organizational activities: Olympiad Fiztech, Winter Olympiad School actions, Start v nauku, MIPT Open doors day;
- Have 4 years' experience as text editor and designer in Vkontakte groups: Profkom MIPT, DCAM MIPT, Phystech Tour, MIPT Sport Club;
- Well-organized, self confidence, able to keep deadlines successfully, possess proactive approach, quality-oriented and enjoy learning new things.

PROJECTS:

2018 | **Command-line interpreter (Microshell)**

- Microshell provides most of commands, contained in bash
- Syntax is the same as in other implementations of UNIX shell, with regular expressions, pipes, input-output redirection and standard notation available
- Microshell interacts with operating system by a set of UNIX system calls and supports signals and multiprocessing.
- Backend is written fully in C++

2019 | **Website for programming contests (Judex)**

- Judex is a system for automatic testing for student contests
- Backend is written fully in Golang, using the standard libraries and MongoDB Server. Frontend is using JavaScript to interact with users
- Project is using multithreading with goroutines for faster results.

2020	Telegram bot for the campus domestic issues (Domestic bot) <ul style="list-style-type: none"> • This bot should simplify the applications' filling process for the domestic issues in the MIPT campus • Backend is written fully in Python, using the Google Tables. Frontend is using JavaScript to interact with users.
2020	Multithreaded list <ul style="list-style-type: none"> • Stack is implemented with two-way adding methods <i>push_front()</i> and <i>push_back()</i> and front deleting <i>pop_front()</i>. Due to the <i>Iterator</i> definition we could make range based loop for our list or output it with output function. To check ABA problem was implemented swap methods with sleep and yield realisations <i>swapSleep</i> and <i>swapYield</i> respectively. Deleting is implemented with <i>Hazardpointers</i> • Project is fully written in C++
2020	B-tree <ul style="list-style-type: none"> • B-tree realisation. This tree is a self-balancing tree data structure that maintains sorted data and allows searches, sequential access, insertions, and deletions in $O(\log(n))$ • Backend is written fully in C++
2020	Heat equation <ul style="list-style-type: none"> • During the course of Distributed systems had been made several tasks: integration using trapezoidal rule (on OpenMPI and OpenMP), solution of the heat equation (on OpenMPI and OpenMP) and bignum arithmetic (on OpenMPI) • Project is written on C and C++
2021	Model of a distributed system <ul style="list-style-type: none"> • Realisation of an infrastructure that allows to create models of distributed processes • Requirements: <ol style="list-style-type: none"> 1. Modeling of distributed processes that exchange messages 2. Simulation of synchronous and asynchronous operation mode 3. Simulation of message loss • Backend is written fully in Golang
2021	Garbage collector in search engines, based on bitmap-indexes <ul style="list-style-type: none"> • Thesis at the Department of applied and theoretical informatics, MIPT • Realisation of the effective garbage collection algorithm, made tests and comparison with other algorithms • Backend is written fully in Golang