think-cell

C++ developer

HITCONTRACT

hitcontract.lt

Vilnius, Lietuva

Aprašymas

We are looking for smart, creative **C++ developers** with a solid theoretical background to join our client's - **think-cell** team. Developers hired by our client in the past mostly hold an exceptional master's degree in computer science or even a doctorate.

About the company:

The company was established in 2002 and now has over 40 employees from 17 countries. think-cell is the leading data visualization software for business presentations. Their challenge is to offer the most intuitive user interface for generating complex data-driven charts and slides, while at the same time ensuring consistency, accuracy, and seamless integration with Microsoft Office. They save time for their customers by automating many tasks for them, like chart labeling or slide layout. More than 750,000 users worldwide (such as American Express, Coca-Cola, Deloitte Consulting, Ernst & Young, Google, Hewlett-Packard, Nokia, Porsche Consulting, etc.) rely on their software for their daily business. They work on challenging visualization problems, reverse engineering of Microsoft's code, and reinventing the user interface. And they do this all based on their own pioneering C++ library, which they have the liberty to perfect along with the rest of their code. think-cell is the only German company funding a C++ ISO committee delegation, so there is a good chance that components they invent will find their way into the standard. think-cell is a kind of company developers truly like. In fact, of our now 20 full-time developers, in

17 years of think-cell, only two ever quit their jobs. They are highly profitable, so they can give you the time and resources to write beautiful code. There are no meetings. All management (the two co-founders) are computer science PhDs, so no demands from people who do not understand the trade.

About this job:

You will work largely independently and will be responsible for the whole range of activities when implementing a new feature. You should be able to look at a problem from the user's perspective and discuss abstract concepts with fellow developers. Company's developers do architecture, design, implementation, customer feedback and bug fixing, rather than splitting these activities between several people. Think-cell put everyone in control of their own work.

At think-cell, there are no deadlines and no scheduled meetings. A feature is ready to be released when you are convinced that you have implemented the best possible solution. Meetings take place as needed, with only the people that are actually involved being required to be present. At any time in the process, ideas, suggestions, and criticism from anybody in the hierarchy are welcome and are seriously considered. Your ideas are welcome, even if they mean that they have to change a lot of code to make things better.

They have published several scientific articles in the areas of Artificial Intelligence and Computer Graphics and they will encourage you to do the same. They sponsor visits to conferences and have close relationships with universities and research institutes in the U.S. and Germany.

Job requirements:

- Language: fluent English is mandatory, German is a plus
- Working experience: any • Education: EU citizens/residence holders – any; others have to have a University degree (starting from Bachelor) so they can sponsor the work permit

* They accept different profiles for C++ developer position, disregarding the former specialization and years of experience. For think-cell, it is more important to evaluate the programming abilities rather than a nice resume and past projects. As all their recruitment steps are task-based, the selection of the candidates based only on the results demonstrated during the

tests. They actually have several developers who joined think-cell right after university graduation without any previous working experience.

Company offers:

- A wide array of extremely challenging C++ development tasks
- An international team of brilliant minds
- A working environment that makes this team stay and grow
- Enough time to make sure that every detail of your solution is perfect
- A flat organization and plenty of room for your ideas
- No scheduled meetings
- Family-friendly working hours, no deadlines, no overtime
- Support for relocation
- A competitive salary from the start and a raise to EUR 120,000 annually after only one year

Additional details:

<u>About their software</u>

Their focus is on business slides (as opposed to more artful applications) because they offer great potential for automation of layout tasks that are traditionally performed by PowerPoint users themselves. Challenges are plenty: from a solid understanding of what makes a good layout and which guidelines are followed by humans who do manual layout, to algorithms that produce an acceptable output fast enough for interactive slide design, to a graphical user interface that supports their new, original approach to slide layout in a way that is easy to understand yet unobtrusive, to solid technical solutions for automatic bug reporting and automatic updates, to compatibility with third-party software on the computers of half a million users.

Here are some highlights of what they have done.

<u>Language</u>

- Everything they do is C++. Even their customer portal is written in C++. There is some Assembler glue code where it is necessary, and their build scripts are written in Python, but other than that think-cell is all about C++
- Naturally, they use C++11 features like lambdas and rvalue references throughout their codebase, and have switched to C++14 where their

compilers support it

- they fund the working group for programming languages of the German Institute for Standardization (DIN). Some of their employees are members of this committee and vote in the international standardization process of ISO/IEC C++
- they sponsor the Standard C++ Foundation helping them to promote the understanding and use of modern Standard C++ on all compilers and platforms

Library

- they use Boost throughout their code, e.g., Boost.Spirit for parsing
- they have their own range library, in the same spirit as Boost.Range or Eric Niebler's range-v3, but going further, for example, by unifying internal and external iteration. We gave a talk about it, and most of the code is public
- they develop their own cross-platform library to support Mac and Windows with a single code base
- they have tehir own reference-counting and persistence libraries to save and restore whole object trees
- they have an extensive bug reporting infrastructure. Assertions and error checks stay in the release code, and tehir software automatically reports bugs to their server. The server analyzes the bug, categorizes it, and files it in a database that all developers can access. If an update fixes the bug, the user can download the update directly from a bug response web page

Algorithms

- think-cell was founded on the idea for an algorithm for automatic slide layout, and they are still on an exciting journey towards that ambitious vision. You can see their most recent release in action!
- they developed a new algorithm for automatic point cloud labeling that allows labels to be positioned away from the actual points
- they developed a new algorithm for automatic column chart labeling
- they are working with John Forrest – author of the linear solver CLP – to make his simplex code faster on their kind of problems

- they developed many generic data structures that are not in C++ or Boost, for example partitions
- their software not only produces charts, it is also able read them back from paper. For our chart recognition tool, they rely on OpenCV and the Leptonica Image Processing Library

Reverse Engineering

- they do lots of reverse engineering with the disassembler IDA from Hex-Rays, in order to achieve things that are not possible via the documented Microsoft Office API
- they wrote probably the best function hooking engine out there. On each start of their software, they patch the Microsoft Office executables in memory. Rather than hard-coding patch addresses, they search for small chunks of assembly code to be robust against minor changes in the executables

think-cell

Required Skills

PROGRAMAVIMAS
C++ 3-4 metai
UŽSIENIO KALBOS
Anglų Independent User (B1/B2)