Approved 01.12.2025	
Name of training programme	Software developer Java
Training programme group	153135 Software and application development and analysis / Tarkvara ja rakenduste arendus ning analüüs
Learning objectives	By the end of the training, the student knows the basics of programming in Java and will be able to develop console and web applications, and be a candidate for the role of Junior developer
Knowledge and skills acquired through successful completion of the curriculum Learning outcomes to be achieved	At the end of the training, the student: - creates java applications using modern coding standards and the capabilities of the Java language (17); - applies the principles of object-oriented programming; - has knowledge and skills in working with Java Core, Collections - knows the maven build tool; - applies unit tests and integration tests in application development - knows the version control system: git / github - knows the basics of working with databases, SQL queries - understands the principles of CI/CD, Docker - knows the basics of the Spring framework and how to use it for building applications
Who this course is for, the target group	Professionals planning to work in IT (or already working) who want to gain the basic knowledge and skills needed to work as a junior Software engineer in Estonia and beyond.
Conditions imposed on the trainee to start the training, if they are a prerequisite for achieving the learning outcomes	Basic computer skills are required for participation in the training. The student must have a computer connected to the Internet.
The language of instruction	English Language
Total amount of training, including the proportion of classroom, practical and independent work	180 ac. hours
Independent work	120 ac. hours
Work in the classroom	60 ac. hours (30 online classes of 2 academic hours each)
Duration of training	6 months
The content of the training: Structure and scope of training	The training program was created based on the professional standard Noorem tarkvaraarendaja, tase 4 (Junior Software Developer, EstQF Level 4, Junior Software Developer, Level 4
	Block 1. Introduction to algorithms and programming with Java This module starts with a look at programming fundamentals and problem-solving approaches. Using Scratch as a starting point, they'll learn essential concepts like variables, logical operators, and loops. Next, we'll explore Java's structure, development tools, and the importance of version control with Git. Throughout the course, students progress from learning Java syntax to applying their skills in a hands-on project, allowing students to learn at their own pace. Coding practices, debugging techniques, and applying programming knowledge are emphasized throughout.
	Unit 2: Java core and testing fundamentals In this block, students will deepen their Java expertise, progressing from methods and abstraction to handling exceptions. The course continues with lessons on classes, objects, encapsulation, and generics, fostering a deeper understanding of object-oriented programming. With a Java practice session, you'll put everything you've learned into practice. Lessons further explore Java collections, inheritance, polymorphism, and introduce essential tools like Maven, Lombok, and the JUnit testing framework. Students learn about file operations, lambda expressions, and functional interfaces. All concepts are integrated into a comprehensive project, providing students with insight into software development lifecycles, teamwork, and project management.
	Unit 3: Introduction to Spring Boot framework and software engineering best practices In this module, students explore client-server architecture, REST API, and HTTP requests in Java. Lessons advance to CI/CD principles, Docker, Linux console usage, and SQL database management. Students will learn project planning, Spring Boot framework basics, and how to use Docker Compose and Flyway. The next sessions cover testing strategies, including unit tests with MockMvc and integration tests with test containers.
	Unit 4: Employment programme Preparing a CV and setting up a LinkedIn page Soft skills mock interview and best practices Motivational letter, search strategy, and discussion of the offer
	Final test The course will conclude with an online quiz to reinforce what you have learned and advice on further reading material, videos and articles.
Teaching methods:	**Classroom (online) work:** theoretical material (lectures and discussion of examples). **Practical work (online and independently):** completing tasks, conducting software development, preparing documentation. All practical tasks are completed in the Intellij IDEA environment, github and are checked by the teacher. Detailed feedback is provided in writing. **Practical exercises will include:** tasks for developing algorithms, using data types and structures, developing console and web applications, as well as mastering various tools for building and running applications **Independent work** includes reading additional material on the subject and watching training videos. In between lessons, there is discussion in a private group on Telegram
Description of the learning environment:	Classes take the form of online video conferences on Zoom and Google Documents. In the learning environment, theoretical and practical lessons are taught and homework is sent out. The student asks his/her own questions and receives feedback from the teacher. The number of students in one group ranges from 4 to 12. One academic hour is 45 minutes. Each lesson lasts 2 academic hours. Classes are held twice a week on weekdays, either in the evening.
List of training materials	Teaching material is made available to students electronically. - Lecture notes; - Project and documentation templates; - Training simulators.
Completion requirements, including assessment methods and assessment criteria	Doing homework, staying in online classes. The final test is used to assess learning outcomes.
Conditions for graduation and documents to be issued (Certificate or Certificate)	Successful completion of the course requires attendance in at least 80% of the academic and practical lessons and completion of all homework. Achievement of the learning outcomes is assessed through practical work and a final test. A certificate is issued to a learner who has attended at least 80% of the lessons, completed all practical work and passed the final test. A certificate of participation or completion is issued to a learner if the learning outcomes have not been achieved, but the learner has participated in the training. A certificate is issued according to the number of training hours attended, if the pupil has participated in at least half of the classes
A description of the training provider's qualifications, training or work experience required for the training	Ilya Nikolsky is an engineer with 5 years of software development experience. 1 year adult teaching experience.