

УТВЕРЖДЕНА
приказом директора АНО «Фортуна»
от 23.08.2026 № 5-ОД

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА «Основы Java»

Направленность: техническая. Уровень: базовый. Адресат: взрослые (18+). Срок реализации: 18 недель (72 академических часа)

1. Пояснительная записка

1.1. Дополнительная общеобразовательная общеразвивающая программа «Основы Java» (далее — Программа) разработана в соответствии с Федеральным законом от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации», Порядком организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам, утверждённым приказом Минпросвещения России от 27.07.2022 № 629, и Правилами применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ, утверждёнными постановлением Правительства Российской Федерации от 11.10.2023 № 1678.

1.2. Направленность Программы — техническая. Программа не является образовательной программой профессионального обучения и не предусматривает присвоение квалификации.

1.3. **Актуальность.** Java остаётся одним из наиболее востребованных языков разработки серверных приложений в Российской Федерации. Программа даёт систематизированную основу языка и практику решения типовых задач, необходимую для дальнейшего профессионального развития обучающегося.

1.4. **Отличительные особенности.** Программа реализуется с применением исключительно электронного обучения и дистанционных образовательных технологий в электронной информационно-образовательной среде «Ломть LMS». Каждое практическое задание проверяется преподавателем индивидуально с содержательным комментарием к решению; количество попыток сдачи задания не ограничено в пределах срока обучения.

1.5. **Адресат Программы** — совершеннолетние лица (18 лет и старше), не имеющие требований к уровню образования. Предварительная подготовка в области программирования не требуется. Необходимы базовые навыки работы с персональным компьютером.

1.6. **Цель Программы** — формирование у обучающихся базовых знаний и практических умений разработки программ на языке Java, достаточных для самостоятельного решения типовых учебных и прикладных задач.

1.7. Задачи Программы:

- обучающие: сформировать знание синтаксиса и базовых конструкций языка Java, принципов объектно-ориентированного программирования, основных структур данных и алгоритмов, базовых возможностей платформы Spring Framework;
- развивающие: развить алгоритмическое мышление, умение декомпозировать задачу, находить и устранять ошибки в программном коде, читать чужой код;
- воспитательные: сформировать культуру оформления кода, ответственное отношение к срокам выполнения работ и навык самостоятельного поиска информации в документации.

1.8. **Объём и срок освоения:** 72 академических часа, 18 недель. Академический час — 45 минут. Рекомендуемая недельная нагрузка — 4 академических часа.

1.9. **Форма обучения** — очная с применением исключительно электронного обучения и дистанционных образовательных технологий. Обучение осуществляется асинхронно; занятия проводятся индивидуально

или в группах численностью не более 10 человек.

1.10. Особенности организации образовательного процесса. После документального оформления прав АНО «Фортуна» на использование ЭИОС и до начала обучения учебные материалы, задания и результаты проверки размещаются по адресу <https://lms.lomt.ru>. Взаимодействие с преподавателем осуществляется в ЭИОС и по электронной почте. Освоение по индивидуальному учебному плану допускается без сокращения объема Программы.

2. Планируемые результаты освоения Программы

2.1. Предметные результаты. По завершении освоения Программы обучающийся:

- знает синтаксис языка Java, систему типов, управляющие конструкции, правила работы с исключениями;
- знает принципы объектно-ориентированного программирования и умеет применять их при проектировании классов;
- знает базовые структуры данных и оценку алгоритмической сложности типовых операций;
- умеет разрабатывать, компилировать и запускать программы на языке Java, использовать коллекции и обрабатывать ошибки;
- умеет разрабатывать простое серверное приложение с использованием Spring Framework;
- владеет навыком чтения технической документации и разбора сообщений об ошибках.

2.2. Метапредметные результаты: умение планировать самостоятельную учебную деятельность и соблюдать сроки, декомпозировать задачу на подзадачи, аргументировать выбранное техническое решение.

2.3. Личностные результаты: ответственность за результат работы, готовность к самостоятельному поиску информации, аккуратность в оформлении результатов труда.

3. Учебный план

№	Наименование модуля	Всего часов	Теория	Практика	Форма аттестации (контроля)
1	Основы языка Java	18	6	12	Практические задания модуля, итоговое практическое задание модуля
2	Объектно-ориентированное программирование на Java	16	5	11	Практические задания модуля, итоговое практическое задание модуля
3	Алгоритмы и структуры данных	14	4	10	Практические задания модуля, итоговое практическое задание модуля
4	Разработка приложений с использованием Spring Framework	16	5	11	Практические задания модуля, итоговое практическое задание модуля
5	Итоговая практическая работа	8	0	8	Итоговая аттестация: защита итоговой практической работы
	Итого	72	20	52	

4. Содержание Программы

Модуль 1. Основы языка Java (18 часов)

Теория (6 часов). Платформа Java: JDK, JRE, виртуальная машина. Структура программы, компиляция и запуск. Примитивные типы данных и ссылочные типы. Переменные, литералы, операторы. Управляющие конструкции: ветвления и циклы. Массивы. Методы: объявление, параметры, возвращаемые значения, перегрузка. Строки и их особенности. Исключения: механизм обработки, проверяемые и непроверяемые исключения.

Практика (12 часов). Установка и настройка среды разработки. Разработка программ с вводом и выводом данных. Задачи на ветвления и циклы. Обработка одномерных и двумерных массивов. Разбор и исправление ошибок компиляции и времени выполнения. Написание методов с обработкой некорректных входных данных.

Форма контроля: практические задания по темам модуля, итоговое практическое задание модуля.

Модуль 2. Объектно-ориентированное программирование на Java (16 часов)

Теория (5 часов). Классы и объекты, поля и методы, конструкторы. Инкапсуляция и модификаторы доступа. Наследование, переопределение методов, ключевое слово `super`. Полиморфизм. Абстрактные классы и интерфейсы. Перечисления. Статические члены класса. Пакеты и организация кода. Принципы проектирования классов.

Практика (11 часов). Проектирование иерархии классов для предметной области. Реализация интерфейсов. Разработка учебного приложения из нескольких взаимодействующих классов. Рефакторинг кода с устранением дублирования.

Форма контроля: практические задания по темам модуля, итоговое практическое задание модуля.

Модуль 3. Алгоритмы и структуры данных (14 часов)

Теория (4 часа). Оценка сложности алгоритмов. Коллекции Java: списки, множества, отображения; их внутреннее устройство и области применения. Итераторы. Сортировка и поиск. Рекурсия. Введение в потоковую обработку данных (Stream API).

Практика (10 часов). Решение задач с использованием коллекций. Реализация алгоритмов сортировки и поиска. Задачи на рекурсию. Обработка наборов данных с помощью Stream API. Сравнение решений по вычислительной сложности.

Форма контроля: практические задания по темам модуля, итоговое практическое задание модуля.

Модуль 4. Разработка приложений с использованием Spring Framework (16 часов)

Теория (5 часов). Назначение Spring Framework. Инверсия управления и внедрение зависимостей. Компоненты приложения и их жизненный цикл. Spring Boot: автоконфигурация, запуск приложения. Разработка веб-слоя: контроллеры, маршрутизация, обработка запросов. Работа с реляционной базой данных: сущности, репозитории. Конфигурация приложения и профили.

Практика (11 часов). Создание приложения на Spring Boot. Разработка контроллеров и обработка запросов. Подключение базы данных и реализация операций сохранения и чтения данных. Обработка ошибок. Написание простых автоматических тестов.

Форма контроля: практические задания по темам модуля, итоговое практическое задание модуля.

Модуль 5. Итоговая практическая работа (8 часов)

Практика (8 часов). Выбор темы и постановка задачи итоговой работы. Проектирование структуры приложения. Реализация. Подготовка к защите: описание принятых технических решений. Защита итоговой практической работы.

Форма контроля: итоговая аттестация — защита итоговой практической работы.

5. Календарный учебный график

Период обучения	Модуль	Объём , ак. ч	Форма контроля
Недели 1–5 (20 ак. ч, из них 2 ак. ч резерв)	Модуль 1. Основы языка Java	18	Итоговое практическое задание модуля 1
Недели 6–9	Модуль 2. Объектно-ориентированное программирование на Java	16	Итоговое практическое задание модуля 2
Недели 10–12 (12 ак. ч) и первая половина недели 13	Модуль 3. Алгоритмы и структуры данных	14	Итоговое практическое задание модуля 3
Вторая половина недели 13 — неделя 16	Модуль 4. Разработка приложений с использованием Spring Framework	16	Итоговое практическое задание модуля 4
Недели 17–18	Модуль 5. Итоговая практическая работа	8	Итоговая аттестация

5.1. Продолжительность учебного года: обучение организуется в течение всего календарного года по мере комплектования групп. Конкретные даты начала и окончания обучения устанавливаются договором об образовании.

5.2. Каникулы не предусмотрены; при необходимости обучающийся вправе воспользоваться приостановлением образовательных отношений в порядке, установленном локальным нормативным актом.

5.3. Нерабочие праздничные дни, установленные законодательством Российской Федерации, не являются днями проверки работ; сроки проверки в эти дни не исчисляются.

6. Оценочные материалы и формы аттестации

6.1. Настоящий раздел является фондом оценочных средств Программы. Полные формулировки заданий, материалы уроков, поля для сдачи работы, результаты проверки и комментарии преподавателя размещены в электронной информационно-образовательной среде «Ломть LMS». Текущий контроль и промежуточная аттестация оцениваются по системе «зачтено» / «не зачтено». Работа засчитывается при выполнении всех перечисленных для неё критериев; при незачёте преподаватель указывает, что необходимо исправить, после чего обучающийся вправе повторно сдать работу в пределах срока обучения.

6.2. Задания текущего контроля по темам модулей

Модуль 1. Основы языка Java

Тема	Практическое задание и представляемый результат	Критерии «зачтено»
Введение в Java и первая программа	Создать <code>Main.java</code> , выводящий <code>Hello, World!</code> и не менее одной строки об обучающемся или его учебной цели; представить исходный код и пример вывода.	Есть <code>public class Main</code> и метод <code>main</code> ; программа компилируется и запускается; выведено не менее двух понятных строк.
Переменные, типы данных и ввод/вывод	С помощью <code>Scanner</code> запросить имя, возраст и город, сохранить значения в переменные подходящих типов и вывести связанную фразу; представить код и пример запуска.	Использованы <code>String</code> и числовой тип; ввод читается корректно; результат снабжён поясняющим текстом.
Операторы и выражения	Прочитать два числа и вывести сумму, разность, произведение, среднее и остаток от деления; представить код и пример запуска.	Среднее вычисляется как дробное число; результаты подписаны; исключено деление на ноль.
Условия <code>if/else</code> и <code>switch</code>	Прочитать оценку от 1 до 5 и вывести соответствующий комментарий, предусмотрев обработку значения вне диапазона; представить код и примеры допустимого и ошибочного ввода.	Для разных оценок получаются разные результаты; неверный ввод обработан; условия не дублируются без необходимости.
Циклы <code>for</code> , <code>while</code> , <code>do while</code>	Прочитать число от 1 до 10 и циклом вывести десять строк таблицы умножения; представить код и пример вывода.	Ровно десять строк; нет ручного повторения десяти команд вывода; неверный диапазон обработан.
Массивы и обработка данных	Для массива из 5–10 целых чисел вычислить минимум, максимум, сумму и среднее; представить код и вывод.	Все элементы обрабатываются циклом; границы массива не нарушены; среднее имеет дробный тип.
Методы	Реализовать методы <code>sum</code> , <code>average</code> и <code>max</code> , принимающие параметры и возвращающие результат, и вызвать каждый не менее двух раз; представить код и вывод.	Методы используют <code>return</code> ; основная логика вынесена из <code>main</code> ; для среднего используется <code>double</code> .

Модуль 2. Объектно-ориентированное программирование на Java

Тема	Практическое задание и представляемый результат	Критерии «зачтено»
Класс, объект, состояние и поведение	Описать объект учебной области (студент, курс, урок или задание): указать класс, не менее трёх полей и двух действий; представить текстовое описание или код.	Класс отличён от экземпляра; отдельно названы данные и действия; пример относится к заявленной предметной области.
Классы, поля, методы и конструкторы	Создать класс 'Student' с полями 'name', 'age', конструктором и 'printInfo'; создать в 'main' два объекта и вывести их данные.	Объекты создаются через 'new'; конструктор заполняет поля; метод использует состояние конкретного объекта.
Инкапсуляция	Сделать поля 'Student' закрытыми, добавить методы доступа и запретить возраст меньше 1; представить код и пояснение обработки ошибки.	Нет прямого внешнего доступа к полям; проверка действует и при создании объекта; неверное состояние не сохраняется.
Взаимодействие объектов	Создать классы 'Student' и 'Course', передать студенту объект курса и вывести совместные сведения; представить код и вывод.	Использованы два отдельных класса; 'Student' хранит ссылку на 'Course'; обязанности классов разделены.
Наследование	Создать базовый 'User' и наследников 'Student', 'Teacher', вывести общие данные в базовый класс и создать объекты наследников.	Применены 'extends' и 'super'; общий код не дублируется; различия находятся в наследниках.
Полиморфизм	Добавить 'describe()' в 'User', переопределить в наследниках и вызвать через переменные или массив типа 'User'; представить код и вывод.	Есть '@Override'; вызов выполнен через общий тип; результат зависит от фактического типа объекта без ручной проверки типа.
Интерфейсы и абстрактные типы	Создать интерфейс 'Printable' или 'Graded', реализовать его классом и вызвать метод через переменную интерфейсного типа.	Интерфейс задаёт поведение; класс использует 'implements'; вызов проходит через тип интерфейса.

Модуль 3. Алгоритмы и структуры данных

Тема	Практическое задание и представляемый результат	Критерии «зачтено»
Оценка сложности	Реализовать линейный и бинарный поиск, указать их асимптотическую сложность и привести результаты для трёх размеров данных.	Бинарный поиск выполняется по отсортированному массиву; оценки сложности верны; различие результатов объяснено.
'List', 'Set', 'Map'	Создать каталог студентов на основе 'Map': добавление, поиск, удаление, вывод уникальных групп, обработка повтора идентификатора и отсутствующей записи.	Операции реализованы; уникальные группы получены структурой без повторов; граничные случаи обработаны.
Итераторы	Сформировать список 'Task', удалить просроченные элементы через 'Iterator', вывести оставшиеся и объяснить риск удаления внутри 'for-each'.	Использован 'iterator.remove()'; состав результата верен; объяснён 'ConcurrentModificationException'.
Сортировка и поиск	Отсортировать студентов по баллу по убыванию и фамилии по возрастанию, выполнить бинарный поиск идентификатора, проверить пустой список и отсутствие значения.	Компаратор задаёт оба ключа; поиск получает отсортированные данные; граничные случаи проверены.
Рекурсия	Реализовать рекурсивную и итеративную сумму массива, проверить пустой массив и один элемент, сравнить ограничения решений.	Есть базовый случай; обе реализации дают одинаковый результат; отмечено ограничение стека вызовов.

Модуль 4. Разработка приложений с использованием Spring Framework

Тема	Практическое задание и представляемый результат	Критерии «зачтено»
Контейнер и компоненты	Реализовать минимальную структуру из контроллера, сервиса и репозитория и описать ответственность каждого слоя.	Классы зарегистрированы как Spring-компоненты; зависимости и обязанности слоёв не смешаны.
Внедрение зависимостей	Создать `NotificationSender` с двумя реализациями и сервисом, получающим зависимость через конструктор; настроить выбор реализации и тестовую подмену.	Использовано конструкторное внедрение; выбор реализации воспроизводим; сервис протестирован без реального отправителя.
Spring Boot и автотесты	Создать приложение с диагностической конечной точкой, вынести имя приложения в конфигурацию и добавить тест; представить структуру и пример ответа.	Приложение запускается; настройка читается из конфигурации; тест подтверждает ответ конечной точки.
Контроллеры и маршрутизация	Реализовать REST API создания, чтения и закрытия учебной задачи с DTO, валидацией и единым обработчиком ошибок.	Контракты отделены от сущностей; корректны успешные статусы; отсутствующая запись и неверный запрос обработаны.
JPA и репозитории	Добавить сущность `Task`, репозиторий, транзакцию создания, ограничение обязательного названия, поиск по статусу и интеграционный тест.	Ограничение действует в модели и БД; операция транзакционна; тест работает с тестовой базой.
Конфигурация, ошибки и тесты	Добавить профиль `test`, глобальный обработчик и тесты успешного запроса, неверных данных и отсутствующей записи; вынести секрет наружу.	Не менее трёх тестов проходят; секрет отсутствует в коде и журнале; ошибки имеют предсказуемый формат.

6.3. Задания промежуточной аттестации

Модуль	Итоговое практическое задание модуля	Представляемые материалы и критерии «зачтено»
1. Основы языка Java	Реализовать консольный калькулятор с меню либо программу учёта оценок с массивом, статистикой и комментарием по среднему баллу.	Исходный код, описание варианта, список не менее трёх собственных методов и пример запуска; используются ввод/вывод, условия, циклы и методы; ошибочная ситуация обработана; программа запускается и решает задачу.
2. ООП на Java	Реализовать мини-систему учёта студентов, курсов и оценок.	Код, описание классов и связей, пример запуска; не менее трёх классов; применены закрытые поля и конструкторы; объекты взаимодействуют; используется наследование, интерфейс или переопределение.
3. Алгоритмы и структуры данных	По списку учебных результатов оставить зачтённые работы, сгруппировать их по модулю, вычислить средний балл и вывести три лучших результата.	Код и пояснение сложности; применены коллекции и Stream API; пустые группы обработаны; группировка, средние значения и тройка результатов вычислены корректно.
4. Spring Framework	Разработать Spring Boot приложение для учёта учебных проектов: REST API, реляционное хранение, валидация, обработка ошибок и тесты.	Исходный код и `README` с запуском и примерами запросов; приложение воспроизводимо запускается; слои разделены; данные сохраняются; тесты покрывают успешные и ошибочные сценарии.

6.4. Итоговая практическая работа и её защита

Итоговая работа подтверждает способность применить материалы всех модулей. Тема согласуется с преподавателем, объём рассчитан на 8 академических часов. Обучающийся последовательно выполняет и сдаёт в ЭИОС следующие задания:

Этап	Задание	Представляемый результат
1. Паспорт проекта	Определить тему, цель, пользователей, 5–10 функциональных требований, ограничения, критерии готовности и план; указать применяемые темы Java, ООП, алгоритмов и Spring.	Паспорт проекта с проверяемыми требованиями и планом.

Этап	Задание	Представляемый результат
2. Проектное решение	Подготовить схему компонентов, перечень классов, модель данных и контракты API либо пользовательского взаимодействия; для двух ключевых решений назвать обоснование и альтернативу.	Схема и текстовое описание архитектуры и модели данных.
3. Готовое приложение	Реализовать проект, тесты и 'README'; показать чистую сборку и запуск, не менее трёх функциональных и двух ошибочных сценариев, назвать ограничения.	Исходный код или архив/ссылка на него, результаты тестов, 'README' и демонстрационные примеры.
4. Защита	Продемонстрировать результат и ответить, какую задачу решает приложение, как устроена модель данных, какие алгоритмы и средства Spring применены и как проверялась корректность.	Демонстрационный сценарий и устная защита по видеоконференции либо письменная защита в ЭИОС по выбору обучающегося.

6.5. Критерии оценки итоговой практической работы:

Критерий	Показатель оценки «зачтено»
Функциональная корректность	Выполнены заявленные функции, демонстрационные сценарии воспроизводятся, некорректные входные данные обработаны.
Применение изученного материала	Обоснованно применены конструкции Java, принципы ООП, структуры данных или алгоритмы и средства Spring, предусмотренные учебным планом.
Структура и читаемость кода	Код разделён на классы, методы и слои с понятными именами; дублирование основной логики отсутствует.
Данные, тесты и документация	Хранение данных соответствует задаче; критическая логика покрыта тестами; 'README' позволяет собрать и запустить проект.
Обоснование решений и защита	Обучающийся объясняет модель данных и архитектуру, отвечает на вопросы и связывает решения с требованиями проекта.

6.6. Итоговая оценка «зачтено» выставляется при соответствии работы всем пяти критериям пункта 6.5 и успешной защите. Допускается не более двух повторных попыток прохождения итоговой аттестации в пределах срока обучения.

6.7. Лицам, успешно прошедшим итоговую аттестацию, выдаётся сертификат об освоении дополнительной общеобразовательной общеразвивающей программы «Основы Java» объёмом 72 академических часа.

7. Организационно-педагогические условия реализации Программы

7.1. Кадровое обеспечение. До подачи заявления Организация оформляет кадровое основание участия педагогического работника и проверяет документы о квалификации. Планируемый преподаватель — Овчаренко Андрей Александрович: диплом магистра по направлению 09.04.03 «Прикладная информатика» и диплом о профессиональной переподготовке с присвоением квалификации «преподаватель технических дисциплин высшей школы» подтверждены первичными сканами; конкретное назначение оформляется отдельным кадровым документом.

7.2. Материально-техническое обеспечение. До подачи заявления Организация документально оформляет право использования административного рабочего места, ЭИОС, домена, серверной инфраструктуры, программного обеспечения и средств связи. Программа реализуется исключительно с применением электронного обучения и дистанционных образовательных технологий; специальные учебные помещения не заявляются. Настоящий раздел устанавливает требование и не заменяет первичные документы о правах на ресурсы.

7.3. Требования к оборудованию обучающегося: персональный компьютер, доступ к сети «Интернет» со скоростью не менее 5 Мбит/с, современный веб-браузер, установленный комплект разработчика Java (JDK) и среда разработки; для итоговой аттестации в форме собеседования — веб-камера и микрофон.

7.4. Информационное обеспечение. До начала обучения в ЭИОС размещаются электронный курс, внутренние учебные материалы, практические задания, критерии оценивания и электронная библиотека. Обучающимся предоставляется персональный доступ; внешние ссылки используются только как дополнительные справочные ресурсы.

7.5. Методическое обеспечение. Основные методы обучения: объяснительно-иллюстративный (изучение учебных материалов и разбор примеров кода), репродуктивный (выполнение заданий по образцу), частично-поисковый и проектный (итоговая практическая работа). Формы организации: самостоятельная работа обучающегося, индивидуальная консультация, письменная обратная связь по решению.

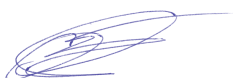
7.6. Список литературы и информационных источников:

- официальная документация платформы Java (Java SE Documentation, Oracle);
- официальная документация Spring Framework и Spring Boot;
- Хорстманн К. Java. Библиотека профессионала;
- Блох Дж. Java. Эффективное программирование;
- учебные материалы и практические задания, разработанные преподавателем и размещённые в электронной информационно-образовательной среде «Ломть LMS».

8. Карта размещения материалов в ЭИОС

Модуль программы	Курс ЭИОС	Урок ов	Обязательное доказательство освоения
1. Основы языка Java	Основы языка Java	8	Теория и задания по урокам 1–8; итоговая мини-программа
2. Объектно-ориентированное программирование	Объектно-ориентированное программирование на Java	8	Теория и задания по урокам 9–16; итоговый ООП-проект
3. Алгоритмы и структуры данных	Алгоритмы и структуры данных	6	Теория и задания по урокам 34–39; итоговое задание модуля
4. Spring Framework	Spring Framework	7	Теория и задания по урокам 40–46; Spring Boot приложение
5. Итоговая практическая работа	Итоговая практическая работа	4	Уроки 47–50: критерии, проектирование, реализация, тестирование, документация и защита

По каждому уроку в ЭИОС до начала реализации Программы размещаются внутренний теоретический материал, практическое задание, критерии оценивания и средство фиксации обратной связи. Электронная библиотека связана с разделами Программы, но внешние ссылки используются только как дополнительные справочные материалы. Полный маршрут проверяется по протоколу готовности ЭИОС до подачи заявления и повторно перед началом обучения.



Директор АНО «Фортуна»

А.А. Овчаренко

(должность)

(подпись)

(инициалы, фамилия)