

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учебно-методическое объединение по образованию
в области информатики и радиоэлектроники

УТВЕРЖДАЮ

Первый заместитель Министра образования
Республики Беларусь

_____ А.Г. Баханович

Регистрационный № _____

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Примерная учебная программа по учебной дисциплине

для специальности

6-05-0611-04 Электронная экономика

СОГЛАСОВАНО

Председатель Учебно-методического
объединения по образованию в
области информатики и
радиоэлектроники

_____ В.А. Богуш

СОГЛАСОВАНО

Начальник Главного управления
профессионального образования
Министерства образования
Республики Беларусь

_____ С.Н. Пищов

СОГЛАСОВАНО

Проректор по научно-методической
работе Государственного учреждения
образования «Республиканский
институт высшей школы»

_____ И.В. Титович

Эксперт-нормоконтролер

Минск 2024

СОСТАВИТЕЛИ:

А.А.Ефремов, заведующий кафедрой экономической информатики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат экономических наук, доцент;

В.Н.Комличенко, доцент кафедры экономической информатики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат технических наук, доцент;

М.Н.Салапура, старший преподаватель кафедры экономической информатики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», магистр экономических наук;

Н.О.Петрович, старший преподаватель кафедры экономической информатики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», магистр экономических наук;

Д.А.Сторожев, старший преподаватель кафедры экономической информатики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», магистр экономических наук

РЕЦЕНЗЕНТЫ:

Кафедра информационных технологий в экологии и медицине учреждения образования «Международный государственный экологический институт имени А.Д.Сахарова» Белорусского государственного университета (протокол №7 от 22.02.2024);

Н.В.Шалькевич, директор общества с ограниченной ответственностью «Трисофт»

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ В КАЧЕСТВЕ ПРИМЕРНОЙ:

Кафедрой экономической информатики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № 9 от 21.02.2024);

Научно-методическим советом учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № ____ от _____);

Научно-методическим советом по прикладным информационным системам и технологиям Учебно-методического объединения по образованию в области информатики и радиоэлектроники (протокол № 8 от 21.02.2024)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

ХАРАКТЕРИСТИКА УЧЕБНОЙ ДИСЦИПЛИНЫ

Примерная учебная программа по учебной дисциплине «Технологии программирования» разработана для студентов учреждений высшего образования, обучающихся по специальности 6-05-0611-04 «Электронная экономика» в соответствии с требованиями образовательного стандарта общего высшего образования и примерного учебного плана вышеуказанной специальности.

Актуальность изучения учебной дисциплины «Технологии программирования» обоснована необходимостью подготовки специалистов, владеющих фундаментальными знаниями и практическими навыками в области объектно-ориентированного анализа, программирования и элементов проектирования при решении практических задач, создания сетевых приложений.

Воспитательное значение учебной дисциплины «Технологии программирования» заключается в формировании у обучающихся математической культуры и научного мировоззрения; развитии исследовательских умений, аналитических способностей, креативности, необходимых для решения научных и практических задач; развитии познавательных способностей и активности: творческой инициативы, самостоятельности, ответственности и организованности; формировании способностей к саморазвитию, самосовершенствованию и самореализации.

Изучение данной учебной дисциплины способствует созданию условий для формирования интеллектуально развитой личности обучающегося, которой присущи стремление к профессиональному совершенствованию, активному участию в экономической и социально-культурной жизни страны, гражданская ответственность и патриотизм.

ЦЕЛЬ, ЗАДАЧИ УЧЕБНОЙ ДИСЦИПЛИНЫ

Цель учебной дисциплины: приобретение базовых знаний по основам объектно-ориентированного программирования и созданию сетевых программ.

Задачи учебной дисциплины:

изучение принципов проектирования, создания, масштабирования объектно-ориентированных приложений и сетевых программ;

овладение методами, подходами, принципами создания объектно-ориентированных приложений и сетевых программ;

приобретение знаний и навыков проектирования и создания объектно-ориентированных приложений и сетевых программ;

освоение навыков программирования с использованием объектно-ориентированных подходов и сетевых программ;

приобретение навыков работы в интегрированной среде разработки программных продуктов;

освоение возможностей, предоставляемых современными компьютерными сетевыми технологиями;

изучение принципов проектирования, создания, масштабирования сетевых приложений;

овладение методами, подходам, принципами, протоколами создания приложений, реализующих сетевое взаимодействие;

приобретение знаний и навыков проектирования и создания сетевых приложений;

освоение навыков программирования при создании сетевого взаимодействия между программными компонентами.

Базовыми учебными дисциплинами по курсу «Технологии программирования» являются «Иностранный язык», «Линейная алгебра и аналитическая геометрия», «Математический анализ», «Дискретная математика», «Основы алгоритмизации и программирования». В свою очередь, учебная дисциплина «Технологии программирования» является базой для таких учебных дисциплин, как «Основы машинного обучения», «Веб-технологии и программирование».

ТРЕБОВАНИЯ К УРОВНЮ ОСВОЕНИЯ СОДЕРЖАНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

В результате изучения учебной дисциплины «Технологии программирования» формируется следующая базовая профессиональная компетенция: применять технологии программирования и использовать язык программирования в профессиональной деятельности.

В результате изучения учебной дисциплины обучающийся должен:

знать:

базовые понятия и синтаксис объектно-ориентированного языка программирования, технологии программирования и приемы разработки программ;

методы создания и использования основных объектов и конструкций объектно-ориентированного языка программирования;

технологии создания, организации и использования иерархии классов, предопределенных классов и типов данных, методы ограничения доступа и обработки исключительных ситуаций;

методы параметризации классов и их использование для решения практических задач;

методы применения шаблонов и контейнерных абстракций;

методы работы с потоками ввода/вывода и разработки интерактивных приложений;

основы и особенности применяемого языка программирования при разработке клиент-серверных приложений;

методы разработки и отладки клиент-серверного приложения;

технологии и подходы к созданию клиент-серверного взаимодействия и разработке интерфейса пользователя;

уметь:

определять абстракции, модули, строить иерархию классов для реализации программ;

использовать принципы типизации, инкапсуляции, наследования, полиморфизма для разработки программных продуктов;

использовать возможности стандартных библиотек объектно-ориентированного языка программирования;

использовать механизм исключений для создания устойчивых приложений;

создавать собственные и использовать предоставляемые стандартные библиотеки динамических структур данных;

использовать технологию программирования для разработки сложных и масштабируемых программ и систем;

создавать приложения удаленного вызова процедур и методов, приложения в архитектуре клиент-сервер с различными методами и процедурами серверной обработки запросов клиентов;

разрабатывать приложения с различными типами клиентских приложений и интерфейсов, используя современные технологии;

использовать для разработки приложений наиболее распространенные сетевые протоколы обмена данными и другие средства передачи данных в клиент-серверных архитектурах;

владеть:

методами, инструментальными средствами и системами разработки объектно-ориентированных программ;

техникой создания объектно-ориентированных программных компонент и организацией их взаимодействия в программных проектах;

базовыми принципами и технологиями проектирования и программирования сетевых приложений;

техникой компонентно-ориентированной разработки клиент-серверных приложений;

методами разработки приложений распределенной обработки данных и технологии удаленных вызовов;

языками и техниками программирования серверных приложений и интерфейсов пользователя;

программными средствами разработки распределенных информационных систем;

инструментальными средствами разработки сетевых приложений.

Примерная учебная программа рассчитана на 120 учебных часов, из них – 56 аудиторных. Примерное распределение аудиторных часов по видам занятий: лекции – 24 часа, лабораторные занятия – 32 часа.

ПРИМЕРНЫЙ ТЕМАТИЧЕСКИЙ ПЛАН

Наименование раздела, темы	Всего аудиторных часов	Лекции	Лабораторные занятия
Раздел 1. Фундаментальные свойства объектно-ориентированного программирования в Java	14	6	8
Тема 1. Введение в разработку программных продуктов с использованием платформы Java. Лексика языка Java. Классы, объекты, поля, методы. Массивы в приложениях Java	6	2	4
Тема 2. Абстракция, инкапсуляция, наследование, полиморфизм. Использование пакетов и интерфейсов. Использование перечислений, автоупаковки, аннотаций	6	2	4
Тема 3. Обработка исключительных ситуаций	2	2	-
Раздел 2. Поточковая организация системы ввода-вывода в Java. Использование обобщений в Java	10	6	4
Тема 4. Потoki ввода-вывода байт и символов	8	4	4
Тема 5. Использование обобщений	2	2	-
Раздел 3. Многопоточное программирование. Основы программирования сетевых приложений	12	4	8
Тема 6. Распределенные системы и распределенная обработка данных. Основы построения сетей и межсетевого взаимодействия. Программирование сетевого взаимодействия на базе сокетов и дейтаграмм	6	2	4
Тема 7. Многопоточное программирование. Программирование параллельного многопоточного серверного обслуживания клиентов	6	2	4
Раздел 4. Использование Java Collection framework	8	4	4
Тема 8. Коллекции в приложениях Java. Интерфейсы и классы коллекций	6	2	4
Тема 9. Коллекции в приложениях Java. Отображения	2	2	-
Раздел 5. Создание графического пользовательского интерфейса. Организация взаимодействия с базой данных. Использование технологий программирования сетевых web-приложений	12	4	8
Тема 10. Обработка событий пользовательского интерфейса. Использование библиотек Swing и JavaFX для создания пользовательских интерфейсов	6	2	4
Тема 11. Программирование взаимодействия сетевых приложений с базами данных	6	2	4
Итого:	56	24	32

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. ФУНДАМЕНТАЛЬНЫЕ СВОЙСТВА ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ В JAVA

Тема 1. ВВЕДЕНИЕ В РАЗРАБОТКУ ПРОГРАММНЫХ ПРОДУКТОВ С ИСПОЛЬЗОВАНИЕМ ПЛАТФОРМЫ JAVA. ЛЕКСИКА ЯЗЫКА JAVA. КЛАССЫ, ОБЪЕКТЫ, ПОЛЯ, МЕТОДЫ. МАССИВЫ В ПРИЛОЖЕНИЯХ JAVA

Предмет курса и его содержание. Связь курса с другими учебными дисциплинами. Парадигмы программирования и проектирования, их особенности. Основные направления в программировании и проектировании программных продуктов. Возникновение объектно-ориентированного программирования. Эволюция платформы Java и ее основные характеристики. Основы разработки программ на Java. Взаимосвязь между Java и языками C, C++, C#. Переносимость программ, написанных на Java. Особенности создания и использования байт-кода. Типы данных, переменные, массивы.

Классы и объекты в объектно-ориентированном программировании. Основные сведения о классах. Общая форма определения класса. Порядок создания объектов классов. Переменные ссылочного типа. Присваивание объектов класса. Методы класса. Возврат значения из метода. Конструкторы класса. Конструкторы без параметров и параметризованные конструкторы. Использование оператора `new`. Сборка мусора и методы завершения. Модификаторы доступа и управление доступом к элементам класса и к классу. Понятие интерфейса в объектно-ориентированном программировании. Передача объектов методам. Способы передачи аргументов методу. Перегрузка методов и конструкторов класса. Использование рекурсии. Применение ключевого слова `static`. Вложенные и внутренние классы. Использование и перегрузка методов с переменным числом аргументов.

Одномерные, многомерные массивы. Инициализация и обработка элементов массива. Нерегулярные массивы. Альтернативный синтаксис объявления массивов. Присваивание ссылок на массивы. Применение переменной экземпляра `length`. Класс `Arrays`. Разновидность цикла `for-each`. Циклическое обращение к элементам массивов. Создание и использование строк. Массивы строк. Неизменяемость строк. Использование строк в операторе `switch`. Использование аргументов командной строки. Использование тернарного оператора.

Тема 2. АБСТРАКЦИЯ, ИНКАПСУЛЯЦИЯ, НАСЛЕДОВАНИЕ, ПОЛИМОРФИЗМ. ИСПОЛЬЗОВАНИЕ ПАКЕТОВ И ИНТЕРФЕЙСОВ. ИСПОЛЬЗОВАНИЕ ПЕРЕЧИСЛЕНИЙ, АВТОУПАКОВКИ, АННОТАЦИЙ

Основы наследования. Доступ к элементам класса и наследование. Использование ключевого слова `super`. Создание многоуровневой иерархии. Порядок вызова конструкторов. Переопределение методов. Динамическая диспетчеризация методов. Назначение и применение переопределенных методов. Применение абстрактных классов. Использование ключевого слова

final при наследовании и переопределении. Атрибуты доступа к компонентам класса при наследовании. Класс Object.

Пакеты как контейнеры классов. Интерфейсы как способ абстрагировать функционал класса от его реализации. Определение пакета, именование пакета, поиск пакетов и переменная окружения CLASSPATH. Защита доступа, импорт пакетов. Объявление интерфейса, реализация интерфейса, доступ к реализациям через ссылки на интерфейсы. Реализация интерфейса, частичные реализации, вложенные интерфейсы, применение интерфейсов. Переменные в интерфейсах, расширение интерфейсов, методы по умолчанию в интерфейсах. Вопросы множественного наследования. Применение статических методов в интерфейсе.

Основные положения о перечислениях. Перечисления как типы классов. Перечисления как наследники класса Enum. Оболочки типов. Оболочки числовых типов. Автоупаковка. Автоупаковка в выражениях.

Автоупаковка и автораспаковка как средство предотвращения ошибок. Аннотации (метаданные). Основы аннотирования программ. Правила удержания аннотаций. Получение аннотаций во время выполнения программы с помощью рефлексии. Использование значений по умолчанию для элементов типа аннотации. Аннотации-маркеры. Одноэлементные аннотации. Встроенные (предопределенные) аннотации.

Тема 3. ОБРАБОТКА ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦИЙ

Основы обработки исключений. Типы исключений. Необработанные исключения. Необрабатываемые исключения. Использование блоков операторов try, catch. Вывод описания исключения. Применение нескольких операторов catch. Вложенные операторы try. Операторы throw, throws, finally. Встроенные исключения. Создание собственных подклассов исключений. Цепочки исключений. Альтернативные средства обработки исключений. Применение исключений при разработке программ.

Раздел 2. ПОТОКОВАЯ ОРГАНИЗАЦИЯ СИСТЕМЫ ВВОДА-ВЫВОДА В JAVA. ИСПОЛЬЗОВАНИЕ ОБОБЩЕНИЙ В JAVA

Тема 4. ПОТОКИ ВВОДА-ВЫВОДА БАЙТ И СИМВОЛОВ

Использование потоков ввода-вывода. Потоки ввода-вывода байтов и символов. Классы потоков ввода-вывода байтов и символов. Предопределенные потоки ввода-вывода. Чтение данных, вводимых с консоли. Буферизированный ввод данных. Ввод данных с использованием класса Scanner. Чтение символов и символьных строк. Вывод данных на консоль. Класс PrintWriter. Чтение/запись данных из/в файлы (файлов). Автоматическое закрытие файла. Пакет java.io для ввода вывода. Классы и интерфейсы ввода-вывода. Класс File. Каталоги. Применение интерфейса FilenameFilter. Альтернативный метод listFiles(). Создание каталогов. Интерфейсы AutoCloseable, Closeable, Flushable. Исключения ввода-вывода. Классы потоков ввода-вывода. Буферизованные потоки ввода-вывода байтов.

Тема 5. ИСПОЛЬЗОВАНИЕ ОБОБЩЕНИЙ

Понятие и принципы использования обобщений. Типовая безопасность при использовании обобщений. Ограниченные типы. Применение метасимвольных аргументов (подстановочных типов). Ограниченные метасимвольные аргументы. Создание обобщенного метода и конструктора. Обобщенные интерфейсы. Базовые типы и унаследованный код. Иерархии обобщенных классов. Сравнение типов в обобщенной иерархии во время выполнения. Приведение обобщенных типов. Переопределение методов в обобщенном классе. Выведение типов и обобщения. Ограничения при использовании обобщений. Ограничения на обобщенные массивы и исключения.

Раздел 3. МНОГОПОТОЧНОЕ ПРОГРАММИРОВАНИЕ. ОСНОВЫ ПРОГРАММИРОВАНИЯ СЕТЕВЫХ ПРИЛОЖЕНИЙ

Тема 6. РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ И РАСПРЕДЕЛЕННАЯ ОБРАБОТКА ДАННЫХ. ОСНОВЫ ПОСТРОЕНИЯ СЕТЕЙ И МЕЖСЕТЕВОГО ВЗАИМОДЕЙСТВИЯ. ПРОГРАММИРОВАНИЕ СЕТЕВОГО ВЗАИМОДЕЙСТВИЯ НА БАЗЕ СОКЕТОВ И ДЕЙТАГРАММ

Централизованные системы и системы распределенной обработки данных. Вычислительные сети. Преимущества и недостатки распределенной обработки данных. Методы и средства организации распределенной обработки информации. Понятие сетей. Классификация сетей. Глобальные и локальные сети. Основные технологии локальных сетей. Понятие протокола. Особенности протоколов TCP и UDP. Другие протоколы и стандарты организации взаимодействия в сетях.

Понятия сервера и клиента. Виды серверов и области их применения. Архитектура «клиент-сервер». Сокеты TCP серверов и клиентов. Дейтаграммы. Основные классы и интерфейсы реализации сетевого взаимодействия. Программирование последовательного серверного взаимодействия с клиентами на базе протоколов TCP и UDP. Пересылка объектов по сети. Механизмы сериализации объектов. Фабричные методы. Клиентские сокеты по протоколу TCP/IP. Серверные сокеты по протоколу TCP/IP. Дейтаграммы.

Тема 7. МНОГОПОТОЧНОЕ ПРОГРАММИРОВАНИЕ. ПРОГРАММИРОВАНИЕ ПАРАЛЛЕЛЬНОГО МНОГОПОТОЧНОГО СЕРВЕРНОГО ОБСЛУЖИВАНИЯ КЛИЕНТОВ

Модель потоков исполнения в Java. Приоритеты потоков. Синхронизация исполнения потоков. Класс Thread и интерфейс Runnable. Главный поток исполнения. Реализация интерфейса Runnable. Расширение класса Thread. Выбор способа создания потоков исполнения. Создание нескольких потоков исполнения. Синхронизация выполнения потоков. Применение синхронизированных методов. Оператор synchronized. Взаимодействие потоков исполнения. Взаимная блокировка. Приостановка, возобновление и остановка потоков исполнения. Получение состояния потока исполнения. Применение многопоточности. Основные термины и определения многопоточности. Механизмы реализации

потоков. Программирование многопоточного сервера, взаимодействующего с клиентами по протоколам TCP и UDP.

Раздел 4. ИСПОЛЬЗОВАНИЕ JAVA COLLECTION FRAMEWORK

Тема 8. КОЛЛЕКЦИИ В ПРИЛОЖЕНИЯХ JAVA. ИНТЕРФЕЙСЫ И КЛАССЫ КОЛЛЕКЦИЙ

Обзор коллекций. Изменения каркаса коллекций в версиях JDK. Обобщения и каркас коллекций. Использование примитивных типов данных в средствах автоматической упаковки. Цикл for в стиле for each. Интерфейсы коллекций. Интерфейсы Collection, List, Set, NavigableSet, Queue, Deque. Класс ArrayList. Получение массива из коллекции типа ArrayList. Класс LinkedList. Различия между классами коллекций. Классы HashSet, LinkedHashSet, TreeSet, PriorityQueue, ArrayDeque, EnumSet. Доступ к коллекциям через итератор. Применение интерфейса Iterator. Цикл for в стиле for each как альтернатива итераторам. Сохранение объектов пользовательских классов в коллекциях. Массивы. Унаследованные классы и интерфейсы.

Тема 9. КОЛЛЕКЦИИ В ПРИЛОЖЕНИЯХ JAVA. ОТОБРАЖЕНИЯ

Каркас коллекций. Интерфейсы отображений. Классы отображений. Компараторы.

Раздел 5. СОЗДАНИЕ ГРАФИЧЕСКОГО ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА. ОРГАНИЗАЦИЯ ВЗАИМОДЕЙСТВИЯ С БАЗОЙ ДАННЫХ. ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИЙ ПРОГРАММИРОВАНИЯ СЕТЕВЫХ WEB-ПРИЛОЖЕНИЙ

Тема 10. ОБРАБОТКА СОБЫТИЙ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА. ИСПОЛЬЗОВАНИЕ БИБЛИОТЕК SWING И JAVAFX ДЛЯ СОЗДАНИЯ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ

Модель делегирования событий. События, источники событий, приёмники событий. Классы событий. Интерфейсы приемников событий. Применение модели делегирования событий.

Введение в библиотеку Swing. Построение библиотеки Swing. Главные особенности библиотеки Swing. Легковесные компоненты Swing. Подключаемый стиль оформления. Классы и интерфейсы библиотеки Swing.

Основные свойства библиотеки JavaFX. Основные понятия JavaFX. Классы для создания подмостков, сцены. Узлы и графы сцены. Использование компоновки. Класс приложения и методы его жизненного цикла. Запуск JavaFX-приложения. Скелет JavaFX-приложения. Компиляция и выполнение JavaFX-приложения. Поток исполнения приложения. Основы обработки событий в JavaFX. Элементы управления. Обработка событий. Рисование на холсте. Эффекты и преобразования.

Тема 11. ПРОГРАММИРОВАНИЕ ВЗАИМОДЕЙСТВИЯ СЕТЕВЫХ ПРИЛОЖЕНИЙ С БАЗАМИ ДАННЫХ

Обзор основных особенностей проектирования баз данных. Понятия драйвера, соединения и запроса к базам данных, их особенности. Программирование взаимодействия приложения с конкретной СУБД. Использование специализированных фреймворков для взаимодействия с базами данных. Понятие трехзвенной архитектуры. Проектирование трехзвенной архитектуры. Интеграция сетевых приложений и баз данных.

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ**ЛИТЕРАТУРА****ОСНОВНАЯ**

1. Шилдт, Г. Полный справочник по Java : Java SE 6 Edition / Г. Шилдт ; пер. с англ. – 7-е изд. – Москва : Вильямс, 2009. – 1040 с.
2. Эккель, Б. Философия Java / Б. Эккель. – 4-е полное изд. – Санкт-Петербург : Питер, 2018. – 1168 с.
3. Чан, Д. Java : быстрый старт / Д. Чан. – Санкт-Петербург : Питер, 2021. – 272 с.
4. Бэзинс, Б. Java для начинающих. Объектно-ориентированный подход / Б. Бэзинс, Э. Бэкил. – Санкт-Петербург : Питер, 2018. – 688 с.
5. Хорстманн, К. С. Java 2 : в 2 т. Т. 1 : Основы / К. С. Хорстманн, Г. Корнелл ; пер. с англ. – 7-е изд. – Москва : Вильямс, 2008. – 896 с.
6. Хорстманн, К. С. Java 2 : в 2 т. Т. 2 : Тонкости программирования / К. С. Хорстманн, Г. Корнелл. – 8-е изд. – Москва : Вильямс, 2009. – 992 с.
7. Основы разработки веб-приложений на языке программирования Java : учебно-методическое пособие / О. Г. Смолякова. – Минск : БГУИР, 2019. – 131 с.
8. Лафоре, Р. Структуры данных и алгоритмы Java / Р. Лафоре. – 2-е изд. – Санкт-Петербург : Питер, 2018. – 704 с.
9. Седжвик, Р. Computer Science : основы программирования на Java, ООП, алгоритмы и структуры данных / Р. Седжвик. – Санкт-Петербург : Питер, 2018. – 1072 с.
10. Курняван, Б. Программирование web-приложений на языке Java / Б. Курняван. – Москва : Лори, 2014. – 880 с.
11. Java. Руководство для программиста : 75 рекомендаций по написанию надежных и защищенных программ / Ф. Лонг [и др.]. – Москва : Вильямс, 2014. – 256 с.
12. Васильев, А. Н. Java. Объектно-ориентированное программирование : для магистров и бакалавров / А. Н. Васильев. – Санкт-Петербург : Питер, 2013. – 400 с.
13. Флэнаган, Д. Java в примерах : Справочник / Д. Флэнаган. – Санкт-Петербург : Символ, 2016. – 664 с.

ДОПОЛНИТЕЛЬНАЯ

14. Лафоре, Р. Структуры данных и алгоритмы Java / Р. Лафоре. – 2-е изд. – Санкт-Петербург : Питер, 2018. – 704 с.
15. Мартин, Р. Чистый код. Создание, анализ и рефакторинг / Р. Мартин. – Санкт-Петербург : Питер, 2014. – 464 с.
16. Парлог, Н. Система модулей Java / Н. Парлог ; пер. с англ. А. Павлова. – Санкт-Петербург : Питер, 2021. – 464 с.
17. Гупта, А. Java EE 7. Основы / А. Гупта ; пер. с англ. А. Г. Гузикевича. – Москва : Вильямс, 2014. – 336 с.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

При изучении учебной дисциплины рекомендуется использовать следующие формы самостоятельной работы:

самостоятельная работа в виде решения индивидуальных заданий в аудитории во время проведения лабораторных занятий под контролем преподавателя;

работа с основной и дополнительной литературой, интернет-ресурсами, документацией, рекомендованными преподавателем.

ПЕРЕЧЕНЬ РЕКОМЕНДУЕМЫХ СРЕДСТВ ДИАГНОСТИКИ КОМПЕТЕНЦИЙ ОБУЧАЮЩИХСЯ

Примерным учебным планом по специальности 6-05-0611-04 «Электронная экономика» в качестве формы промежуточной аттестации по учебной дисциплине «Технологии программирования» рекомендуется экзамен. Оценка учебных достижений обучающихся производится по десятибалльной шкале.

Для текущего контроля по учебной дисциплине и диагностики компетенций могут использоваться следующие формы:

электронные отчеты по лабораторным работам с их устной защитой;

контрольные работы;

устный опрос на лабораторных занятиях и лекциях;

электронные тесты на лекциях;

доклады на конференциях.

РЕКОМЕНДУЕМЫЕ МЕТОДЫ (ТЕХНОЛОГИИ) ОБУЧЕНИЯ

Основные рекомендуемые методы (технологии) обучения, отвечающие целям и задачам учебной дисциплины:

проблемное обучение (проблемное изложение, вариативное изложение, частично-поисковый метод, проектные подходы), реализуемое на лекционных занятиях;

учебно-исследовательская деятельность, творческий подход, реализуемые на лабораторных занятиях.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

1. Настройка рабочего окружения. Изучение синтаксиса и типов данных платформы Java.
2. Создание собственных классов и объектов.
3. Использование массивов для хранения набора данных.
4. Наследование. Переопределение методов.
5. Интерфейсы. Абстрактные классы и методы.
6. Потoki ввода/вывода. Преобразование типов.

7. Потоки ввода/вывода. Работа с файлами. Работа со строками.
8. Использование обобщений для создания параметризованных типов данных.
9. Многопоточное программирование.
10. Работа в сети. Протокол TCP/IP.
11. Работа в сети. Протокол UDP.
12. Коллекции в приложениях Java. Классы коллекций.
13. Использование библиотеки Swing.
14. Создание приложения на JavaFX.
15. Организация взаимодействия с базой данных.
16. Создание динамических web-приложений с использованием сервлетов и JSP.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ КОМПЬЮТЕРНЫХ ПРОГРАММ
(необходимого оборудования, наглядных пособий и др.)

1. Операционная система Microsoft Windows 7 или выше.
2. Инструментарий разработчика JDK 21 и выше.
3. Интегрированная среда разработки на объектно-ориентированном языке программирования (IntelliJ IDEA версии 2023.1 и выше или аналогичные IDE).
4. Enterprise Architect 7.1 или выше.