

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 № _____

ПРОГРАММИРОВАНИЕ НА ЯЗЫКАХ ВЫСОКОГО УРОВНЯ

**Примерная учебная программа по учебной дисциплине
для специальности**

6-05-0611-05 Компьютерная инженерия

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.М.Ковальчук, старший преподаватель кафедры электронных вычислительных машин учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»;

Д.Н.Марченков, системный архитектор иностранного общества с ограниченной ответственностью «ЭПАМ Системз»;

Д.Ю.Перцев, доцент кафедры электронных вычислительных машин учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат технических наук, доцент

РЕЦЕНЗЕНТЫ:

Кафедра «Физика» Белорусского национального технического университета (протокол № 10 от 19.03.2024);

В.В.Ганченко, старший научный сотрудник Государственного научного учреждения «Объединенный институт проблем информатики Национальной академии наук Беларуси», кандидат технических наук

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ В КАЧЕСТВЕ ПРИМЕРНОЙ:

Кафедрой электронных вычислительных машин учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № 16 от 08.04.2024);

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

Научно-методическим советом по разработке программного обеспечения и информационно-коммуникационным технологиям Учебно-методического объединения по образованию в области информатики и радиоэлектроники (протокол № 8 от 15.04.2024)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

ХАРАКТЕРИСТИКА УЧЕБНОЙ ДИСЦИПЛИНЫ

Примерная учебная программа по учебной дисциплине «Программирование на языках высокого уровня» разработана для студентов учреждений высшего образования, обучающихся по специальности 6-05-0611-05 «Компьютерная инженерия» в соответствии с требованиями образовательного стандарта общего высшего образования и примерного учебного плана вышеуказанной специальности.

Подготовка современного специалиста требует уверенного владения возможностями, предоставляемыми компьютерными технологиями. Освоение учебной дисциплины «Программирование на языках высокого уровня» обеспечивает подготовку специалиста, владеющего фундаментальными знаниями и практическими навыками в области объектно-ориентированного анализа и программирования, а также реализацией их на современных языках программирования C++ и Java. Знания и навыки, полученные при изучении учебной дисциплины, формируют базовый уровень знаний инженера-системотехника и необходимы для освоения других учебных дисциплин примерного учебного плана специальности 6-05-0611-05 «Компьютерная инженерия».

Воспитательное значение учебной дисциплины «Программирование на языках высокого уровня» заключается в формировании у обучающихся правильной культуры и стиля программирования 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.

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.

ЦЕЛИ, ЗАДАЧИ УЧЕБНОЙ ДИСЦИПЛИНЫ

Цели учебной дисциплины:

приобретение теоретических знаний и практических навыков в области современных технологий объектно-ориентированной разработки программ с использованием современных стандартов;

освоение основных принципов объектно-ориентированного проектирования и программирования программных продуктов, а также информации о современных методах и средствах программирования;

приобретение навыков практического применения полученных знаний для решения инженерных и научно-исследовательских задач.

Задачи учебной дисциплины:

- освоение теоретических основ современных стандартов в объектно-ориентированном программировании;
- изучение методов и приемов объектно-ориентированного программирования, языковых средств проектирования;
- изучение концепций объектно-ориентированных языков высокого уровня, таких как C++ и Java;
- изучение интегрированных средств компоновки и отладки сложных проектов;
- составление представления о проектировании средних и больших проектов.

Базовой учебной дисциплиной для учебной дисциплины «Программирование на языках высокого уровня» является «Основы алгоритмизации и программирования». В свою очередь учебная дисциплина «Программирование на языках высокого уровня» является базой для таких учебных дисциплин компонента учреждения образования, как «Операционные системы и системное программирование», «Жизненный цикл разработки программного обеспечения», «Функциональное программирование».

ТРЕБОВАНИЯ К УРОВНЮ ОСВОЕНИЯ СОДЕРЖАНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

В результате изучения учебной дисциплины «Программирование на языках высокого уровня» формируются следующие компетенции:

универсальные:

владеть основами исследовательской деятельности, осуществлять поиск, анализ и синтез информации;

быть способным к саморазвитию и совершенствованию в профессиональной деятельности;

проявлять инициативу и адаптироваться к изменениям в профессиональной деятельности;

базовая профессиональная: применять методы и средства программирования, объектно-ориентированный подход в программировании и конструировании программ с использованием языков высокого уровня.

В результате изучения учебной дисциплины обучающийся должен:

знать:

технологии объектно-ориентированного программирования (ООП);

основные средства языка C++ для работы с объектами;

механизмы наследования, инкапсуляции и полиморфизма;

обработку исключительных ситуаций;

основы применения шаблонов и стандартные библиотеки шаблонов;

проблемы кроссплатформенности в Java;

мультипоточность и асинхронные вычисления средствами Java;

методы работы с хранилищами данных средствами Java, облачные решения;

уметь:

применять объектно-ориентированный подход при разработке программного обеспечения;

использовать потоки ввода-вывода;

разрабатывать программы с генерацией и обработкой исключительных ситуаций;

разрабатывать и использовать классы шаблонов и стандартных библиотек шаблонов;

разрабатывать прикладное программное обеспечение, используя кроссплатформенный подход;

использовать фреймворки и библиотеки в web-разработке;

владеть:

методами и приемами объектно-ориентированного подхода при проектировании и разработке прикладного программного обеспечения;

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

инструментарием разработки кроссплатформенных приложений;

основами исследовательской деятельности, навыками осуществления поиска, анализа и синтеза информации.

Примерная учебная программа рассчитана на 336 учебных часов, из них – 144 аудиторных. Примерное распределение аудиторных часов по видам занятий: лекции – 80 часов, лабораторные занятия – 64 часа.

ПРИМЕРНЫЙ ТЕМАТИЧЕСКИЙ ПЛАН

Наименование раздела, темы	Всего аудиторных часов	Лекции	Лабораторные занятия
Раздел 1. Концепция и особенности объектно-ориентированного подхода	6	4	2
Тема 1. Концептуальные основы объектно-ориентированного подхода	2	2	-
Тема 2. Принципы объектно-ориентированного программирования	4	2	2
Раздел 2. Базовые абстракции объектно-ориентированного программирования	18	12	6
Тема 3. Классы и объекты	4	4	-
Тема 4. Статические и динамические объекты	4	2	2
Тема 5. Методы и механизмы инкапсуляции	4	2	2
Тема 6. Структурные элементы класса и методы взаимодействия объектов	6	4	2
Раздел 3. Методы и механизмы разработки объектно-ориентированных программ	44	24	20
Тема 7. Наследование в C++	10	6	4
Тема 8. Полиморфизм, механизмы его использования	10	6	4
Тема 9. Параметризация объектов в C++	8	4	4
Тема 10. Исключения и их обработка	8	4	4
Тема 11. Потоки ввода-вывода в C++. Организация работы с файлами	8	4	4
Раздел 4. Стандартная библиотека шаблонов	12	8	4
Тема 12. Организация библиотеки STL и стандартные контейнеры	6	4	2
Тема 13. Итераторы, алгоритмы и распределители памяти	4	2	2
Тема 14. Паттерны проектирования	2	2	
Раздел 5. Экосистема языка Java	24	12	12
Тема 15. Положение Java в иерархии языков и фреймворков	2	2	
Тема 16. Основные средства разработки в Java-экосистеме	6	2	4
Тема 17. Проблемы кроссплатформенности в Java: от JVM к web-приложениям	4	4	
Тема 18. Фреймворки и библиотеки. Использование Spring-фреймворка в web-разработке	12	4	8

Наименование раздела, темы	Всего аудиторных часов	Лекции	Лабораторные занятия
Раздел 6. Базовые концепции языка Java	24	8	16
Тема 19. Реализация принципов ООП в Java. Синтаксис и структуры данных	6	2	4
Тема 20. Отладка и обработка ошибок средствами Java	6	2	4
Тема 21. Мультипоточность и асинхронные вычисления средствами Java	6	2	4
Тема 22. Средства функционального программирования в Java	6	2	4
Раздел 7. Основы разработки программных продуктов средствами Java	16	12	4
Тема 23. «Вечнозелёные» принципы и шаблоны разработки в Java	2	2	
Тема 24. Работа с хранилищами данных средствами Java	8	4	4
Тема 25. Фронтенд-разработка в web- и мультиплатформенных приложениях	4	4	
Тема 26. Облачные решения	2	2	
Итого:	144	80	64

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. КОНЦЕПЦИЯ И ОСОБЕННОСТИ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПОДХОДА

Тема 1. КОНЦЕПТУАЛЬНЫЕ ОСНОВЫ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПОДХОДА

История создания языка C++, стандарты языка. Новые возможности языка C++. Основные направления в программировании. Парадигмы программирования.

Тема 2. ПРИНЦИПЫ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ

Базовые принципы ООП. Основные положения объектной модели, ее преимущества. Абстрагирование. Модульность. Иерархия. Типизация. Пространство имен и стандартные библиотеки языка, организация ввода-вывода.

Раздел 2. БАЗОВЫЕ АБСТРАКЦИИ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ

Тема 3. КЛАССЫ И ОБЪЕКТЫ

Базовые конструкции объектно-ориентированных программ: классы и объекты. Инициализация и разрушение объекта. Компоненты класса. Конструкторы и деструктор. Перегрузка и переопределение методов класса. Динамическое выделение памяти.

Тема 4. СТАТИЧЕСКИЕ И ДИНАМИЧЕСКИЕ ОБЪЕКТЫ

Указатели и ссылки. Операторы для динамического выделения и освобождения памяти. Статические объекты. Временные объекты.

Тема 5. МЕТОДЫ И МЕХАНИЗМЫ ИНКАПСУЛЯЦИИ

Принцип инкапсуляции. Область действия класса и доступ к компонентам класса. Управление доступом к компонентам класса.

Тема 6. СТРУКТУРНЫЕ ЭЛЕМЕНТЫ КЛАССА И МЕТОДЫ ВЗАИМОДЕЙСТВИЯ ОБЪЕКТОВ

Организация внешнего доступа к локальным компонентам класса. Интерфейсные (дружественные) методы. Статические и константные компоненты класса. Вложенные классы. Перегрузка операций. Преобразование типов.

Раздел 3. МЕТОДЫ И МЕХАНИЗМЫ РАЗРАБОТКИ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ ПРОГРАММ

Тема 7. НАСЛЕДОВАНИЕ В C++

Базовые и производные классы. Спецификаторы наследования. Простое и множественное наследование. Наследование членов класса и доступ к ним. Конструкторы и деструкторы при наследовании. Композиция и наследование.

Тема 8. ПОЛИМОРФИЗМ, МЕХАНИЗМЫ ЕГО ИСПОЛЬЗОВАНИЯ

Виртуальные методы класса. Виртуальные таблицы. Абстрактные классы, их назначение и свойства. Позднее и раннее связывание. Использование виртуального механизма для реализации принципа полиморфизма.

Тема 9. ПАРАМЕТРИЗАЦИЯ ОБЪЕКТОВ В C++

Параметризованные классы и методы, их свойства. Совместное использование параметризации и принципов наследования. Параметризованные классы и статические члены. Задание значений параметров класса по умолчанию.

Тема 10. ИСКЛЮЧЕНИЯ И ИХ ОБРАБОТКА

Основы обработки исключений. Генерация и обработка исключений. Повторная генерация исключений. Обработка неожиданных исключений. Генерация исключений в конструкторах. Развертывание стека. Спецификация исключений. Иерархия исключений стандартной библиотеки.

Тема 11. ПОТОКИ ВВОДА/ВЫВОДА В C++. ОРГАНИЗАЦИЯ РАБОТЫ С ФАЙЛАМИ

Понятие потока. Потоки ввода-вывода: библиотеки, классы и объекты потоков ввода-вывода, встроенные потоки. Состояние потоков ввода-вывода. Связывание классов с потоками ввода-вывода. Файловая система и ее связь с системой ввода-вывода. Текстовые, бинарные и файлы произвольного доступа. Контроль ошибок при работе с файлами.

Раздел 4. СТАНДАРТНАЯ БИБЛИОТЕКА ШАБЛОНОВ

Тема 12. ОРГАНИЗАЦИЯ БИБЛИОТЕКИ STL И СТАНДАРТНЫЕ КОНТЕЙНЕРЫ

Введение в стандартную библиотеку шаблонов, основные понятия. Организация стандартной библиотеки. Типы контейнерных классов. STL-контейнеры.

Тема 13. ИТЕРАТОРЫ, АЛГОРИТМЫ И РАСПРЕДЕЛИТЕЛИ ПАМЯТИ

Категории итераторов, операции с итераторами, итераторы потоков. Стандартный распределитель памяти. Распределители памяти, определяемые пользователем. Обзор алгоритмов стандартной библиотеки.

Тема 14. ПАТТЕРНЫ ПРОЕКТИРОВАНИЯ

Назначение паттернов проектирования. Основные их виды. Решение задач проектирования с помощью паттернов.

Раздел 5. ЭКОСИСТЕМА ЯЗЫКА JAVA

Тема 15. ПОЛОЖЕНИЕ JAVA В ИЕРАРХИИ ЯЗЫКОВ И ФРЕЙМВОРКОВ

Классификация языков программирования. Классификация фреймворков и средств разработки программного обеспечения. Положение Java в общей

классификации. Экосистема языка Java. История развития языка Java. Версионность языка Java, стабильные версии языка, основные вендоры. Главные отличия языка Java. Организация работы программ на Java: JVM, JRE, JDK.

Тема 16. ОСНОВНЫЕ СРЕДСТВА РАЗРАБОТКИ В JAVA-ЭКОСИСТЕМЕ

Структура простейшей программы на Java. Сборка и запуск простейших программ на Java. Обзор IDE для разработки Java-программ. Структура типичных Java-приложений. Средства сборки Java-приложений: ant, maven, gradle. Средства запуска web-приложений: Tomcat, Jetty. Средства контроля версий кода: Git.

Тема 17. ПРОБЛЕМЫ КРОССПЛАТФОРМЕННОСТИ В JAVA: ОТ JVM К WEB-ПРИЛОЖЕНИЯМ

Виды «платформ» и проблема кроссплатформенности. Традиционный взгляд на решение проблемы кроссплатформенности: от виртуальных машин к койнтернеризации. Решение проблемы кроссплатформенности с помощью клиент-сервер архитектуры в web-приложениях. История развития Java как основного средства построения web-приложений. Web-сервисы, основные типы API в web-сервисах: SOAP, REST, GraphQL.

Тема 18. ФРЕЙМВОРКИ И БИБЛИОТЕКИ. ИСПОЛЬЗОВАНИЕ SPRING-ФРЕЙМВОРКА В WEB-РАЗРАБОТКЕ

Экосистема и набор средств Java для решения задач различной направленности. Обзор основных фреймворков и библиотек для построения web-приложений и сервисов в Java. Отличие фреймворка от библиотеки. Обзор основных принципов и средств фреймворка Spring: container, autowiring, AOP. Конфигурация Spring-приложения. Использование Spring MVC для построения web-сервисов. Вызов web-сервисов из Java приложений. Средства вызова и тестирования web-сервисов.

Раздел 6. БАЗОВЫЕ КОНЦЕПЦИИ ЯЗЫКА JAVA

Тема 19. РЕАЛИЗАЦИЯ ПРИНЦИПОВ ООП В JAVA. СИНТАКСИС И СТРУКТУРЫ ДАННЫХ

Классы и объекты в Java. Реализация основных принципов ООП в Java: инкапсуляция, наследование, полиморфизм, абстракция. Типы отношений: агрегация и композиция. Основные типы данных в Java. Классы-«обёртки» над основными типами данных. Работа с коллекциями в Java. Интерфейсы и абстрактные классы. Работа со строками. Сборщик мусора в Java.

Тема 20. ОТЛАДКА И ОБРАБОТКА ОШИБОК СРЕДСТВАМИ JAVA

Иерархия исключений в Java. Обработка исключений: try-catch-finally блок. Проверяемые и непроверяемые исключения. Средства логирования ошибок: Log4J, SLF4J, Logback, ELK стек. Тестирование приложений, пирамида тестирования. Автоматизированное тестирование средствами Java:

unit-тестирование, интеграционное тестирование. E2E-тестирование с помощью фреймворка Selenium. TDD-подход в написании программ.

Тема 21. МУЛЬТИПОТОЧНОСТЬ И АСИНХРОННЫЕ ВЫЧИСЛЕНИЯ СРЕДСТВАМИ JAVA

Мультипоточность и мультизадачность. Мультипоточность средствами Java. Решение проблемы одновременного доступа к ресурсам и потокобезопасность средствами Java: синхронизация, concurrent коллекции, пакет atomic. Организация асинхронных вызовов средствами Java.

Тема 22. СРЕДСТВА ФУНКЦИОНАЛЬНОГО ПРОГРАММИРОВАНИЯ В JAVA

Основные парадигмы и подходы к построению программ: императивный, процедурный, функциональный. Примеры и значение функциональных языков. Основные средства функционального подхода в Java: lambda-функции, функциональные интерфейсы, optional. Обзор Stream API и основных операторов.

Раздел 7. ОСНОВЫ РАЗРАБОТКИ ПРОГРАММНЫХ ПРОДУКТОВ СРЕДСТВАМИ JAVA

Тема 23. «ВЕЧНОЗЕЛЁНЫЕ» ПРИНЦИПЫ И ШАБЛОНЫ РАЗРАБОТКИ В JAVA

Основные «вечнозелёные» принципы при построении программного обеспечения. SOLID-принципы. Основные Java-паттерны для построения программ, классификация паттернов согласно GoF. Обзор основных шаблонов: стратегия, фабрика, прокси, MVC. Базовые архитектурные стили. Основы непрерывной разработки кода: Agile-методологии, CI/CD. Подходы и средства улучшения качества кода: ревью кода, тестирование, статические анализаторы.

Тема 24. РАБОТА С ХРАНИЛИЩАМИ ДАННЫХ СРЕДСТВАМИ JAVA

Понятие данных. Основные типы хранилищ. Реляционные базы данных: основные принципы. Язык SQL, основные команды. Нереляционные хранилища данных: основные принципы, CAP-теорема. Виды и примеры нереляционных хранилищ: MongoDB, Redis. Средства работы с базами данных в Java: ODBC/JDBC, ORM-фреймворки, Hibernate, MyBatis. Средства работы с нереляционными хранилищами на примере MongoDB и Redis.

Тема 25. ФРОНТЕНД-РАЗРАБОТКА В WEB- И МУЛЬТИПЛАТФОРМЕННЫХ ПРИЛОЖЕНИЯХ

Виды фронтенда в клиент-сервер архитектуре, «толстый» и «тонкий» клиенты. Средства Java для построения десктоп приложений: Swing, AWT, JavaFx. Использование GWT/GXT для построения фронтенда в web-приложениях. Основные технологии и фреймворки для построения клиентов web-приложений: HTML, CSS, JavaScript, Ajax, JavaScript-фреймворки. Современные принципы

построения фронтенда web-приложений: адаптивная вёрстка, single-page приложения, микро-фронтенды, SEO. Мультиплатформенная разработка для мобильных устройств: гибридные приложения, кросс-платформенные приложения, PWA.

Тема 26. ОБЛАЧНЫЕ РЕШЕНИЯ

Основные модели облачных решений: SaaS, PaaS, IaaS. Основные поставщики облачных решений. Облачные сервисы на примере AWS. Основные подходы построения приложений в облачных сервисах: cloud native, serverless.

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

1. Страуструп, Б. Язык программирования С++ : 4-е изд. / Б. Страуструп ; пер. с англ. – Москва : Бином, 2023. – 1216 с.
2. Страуструп, Б. Программирование : принципы и практика использования С++ / Б. Страуструп ; пер. с англ. Д. А. Ключина. – Москва ; Санкт-Петербург : Диалектика ; Вильямс, 2015. – 1223 с.
3. Шилдт, Г. С++ : базовый курс / Г. Шилдт. – 3-е изд. – Москва ; Санкт-Петербург : Диалектика, 2019. – 620 с.
4. Буч, Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++ / Г. Буч. – 3-е изд. – Москва : Вильямс, 2020. – 720 с.
5. Дейтел, Х. Как программировать на С++ / Х. Дейтел, П. Дейтел ; пер. с англ. – 8-е изд. – Москва : Бином, 2021. – 1032 с.
6. Таненбаум, Э. Современные операционные системы / Э. Таненбаум, Х. Бос. – 4-е изд. – Санкт-Петербург : Питер, 2021. – 1120 с.
7. Лафоре, Р. Структуры данных и алгоритмы Java / Р. Лафоре. – 2-е изд. – Санкт-Петербург : Питер, 2018. – 704 с.
8. Васильев, А. Н. Java. Объектно-ориентированное программирование / А. Н. Васильев. – Санкт-Петербург : Питер, 2013. – 400 с.
9. Программируем на Java / М. Лой, П. Нимайер, Д. Лук. – 5-е межд. изд. – Санкт-Петербург: Питер, 2023. – 544 с.
10. Java from ERAM : учебно-методическое пособие / И. Н. Блинов, В. С. Романчик. – Минск : Четыре четверти, 2020. – 560 с.
11. Java: эффективное программирование / Дж. Блох – 3-е изд. – пер. с англ. – Санкт-Петербург : Диалектика, 2019. – 464 с.
12. Spring в действии / У. Уоллс. – 6-е изд. – пер. с англ. А. Н. Киселева. – Москва : ДМК Пресс, 2022. – 544 с.
13. Spring. Все паттерны проектирования. / Р. Динеш. – Санкт-Петербург : Питер, 2019. – 320 с.

ДОПОЛНИТЕЛЬНАЯ

14. Рао, С. Освой самостоятельно С++ за 21 день / С. Рао. – 7-е изд. – Москва : Вильямс, 2013. – 688 с.
15. Прата, С. Язык программирования С++. Лекции и упражнения / С. Прата. – 6-е изд. – Москва : Вильямс, 2012. – 1248 с.
16. Хортон, А. Visual С++ 2010 : полный курс / А. Хортон. – Москва : Диалектика, 2010. – 1216 с.
17. Страуструп, Б. Дизайн и эволюция С++ / Б. Страуструп. – Санкт-Петербург : Питер, 2007. – 445 с.
18. Шилдт, Г. Самоучитель С++ / Г. Шилдт. – 3-е изд. – Санкт-Петербург : БХВ-Петербург, 2006. – 688 с.

19. Шилдт, Г. Искусство программирования на С++ / Г. Шилдт. – Санкт-Петербург : БХВ-Петербург, 2005. – 496 с.
20. Климова, Л. С++. Практическое программирование. Решение типовых задач / Л. Климова. – Москва : Кудиц-образ, 2001. – 174 с.
21. Вайсфельд, М. Объектно-ориентированное мышление / М. Вайсфельд. – пер. с англ. – Санкт-Петербург : Питер, 2014. – 304 с.
22. Сьерра, К. Паттерны проектирования / К. Сьерра, Э. Фримен ; пер. с англ. – Санкт-Петербург : Питер, 2014. – 656 с.
23. Вербург, М. Java. Новое поколение разработки / М. Вербург, Б. Эванс пер. с англ. – Санкт-Петербург : Питер, 2014. – 560 с.
24. Хорстман, К. Java 2. Библиотека профессионала / К. Хорстман, Г. Корнелл ; пер. с англ. – Москва : Вильямс, 2012. – Т. 1. – 816 с.
25. Java. Библиотека профессионала / К. Хорстманн, Г. Корнелл. – 9-е изд. – пер. с англ. – Москва : Вильямс, 2014. – Т. 1. Основы. – 864 с.
26. Spring Boot по-быстрому / М. Хеклер. – Санкт-Петербург : Питер, 2022. – 352 с.
27. Java. Полное руководство / Г. Шилдт. – 10-е изд. – пер. с англ. – Санкт-Петербург : Альфа книга, 2018. – 1488 с.
28. Микросервисы. Паттерны разработки и рефакторинга. / К. Ричардсон. – Санкт-Петербург : Питер, 2019. – 544 с.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

При изучении учебной дисциплины рекомендуется использовать следующие формы самостоятельной работы:

- работа с учебно-методическими пособиями;
- изучение рекомендуемой литературы;
- подготовка лабораторных работ по индивидуальным, в том числе разноуровневым заданиям;
- самостоятельное изучение тем и проблем, не выносимых на лекции и лабораторные занятия.

ПЕРЕЧЕНЬ РЕКОМЕНДУЕМЫХ СРЕДСТВ ДИАГНОСТИКИ КОМПЕТЕНЦИЙ ОБУЧАЮЩИХСЯ

Примерным учебным планом по специальности 6-05-0611-05 «Компьютерная инженерия» в качестве формы промежуточной аттестации по учебной дисциплине «Программирование на языках высокого уровня» рекомендуется зачет, экзамен и курсовой проект. Оценка учебных достижений обучающихся производится по системе «зачтено/не зачтено» и десятибалльной шкале.

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

- отчеты по лабораторным работам с их устной защитой;
- устный опрос.

РЕКОМЕНДУЕМЫЕ МЕТОДЫ (ТЕХНОЛОГИИ) ОБУЧЕНИЯ

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

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

элементы контролируемого обучения, реализуемые на лабораторных занятиях, а также в ходе самостоятельной работы обучающихся;

элементы учебно-исследовательской деятельности, творческого подхода, реализуемые на лабораторных занятиях и при подготовке курсового проекта;

проектные технологии, используемые при проектировании конкретного объекта, реализуемые при выполнении курсового проекта.

РЕКОМЕНДАЦИИ ПО КУРСОВОМУ ПРОЕКТИРОВАНИЮ

Целями курсового проекта является:

овладение практическими навыками проектирования и разработки законченного, отлаженного и протестированного программного продукта с использованием методик объектно-ориентированного проектирования и языка программирования высокого уровня;

закрепление и углубление теоретических знаний, полученных при изучении курса «Основы алгоритмизации и программирования», а также данного курса;

углубление знаний по оформлению документации на программный продукт.

Курсовой проект представляет собой задачу по проектированию прикладного программного обеспечения из следующих основных типов:

задачи, связанные с расчетами;

задачи автоматизации действий;

задачи планирования;

задачи моделирования.

Курсовой проект выполняется индивидуально. По согласованию с преподавателем обучающийся может выбрать иную тему курсового проекта, отвечающую вышеперечисленным целям и требованиям.

Для курсового проекта желательна нацеленность результата на требования конкретной группы потребителей, а также реализация в программном обеспечении собственных уникальных функций и алгоритмов обработки данных.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ КУРСОВЫХ ПРОЕКТОВ

1. Разработка программы вычисления по математическим формулам или методам (например, архиватор, калькулятор, построитель графиков, формирователь отчетов).

2. Разработка программы исследования и распознавания объектов или ситуаций (номеров, символов, машин, людей и т.п.).

3. Разработка менеджера организации данных (файлов, коллекций фотографий, музыки и т.п.).
4. Разработка программы просмотра или редактирования данных (текста, графики, видео и т.п.).
5. Разработка программы коммуникации данных (текста, графики, аудио- и видео-данных).
6. Разработка информационно-обучающей программы.
7. Разработка информационно-справочной программы (для различных областей и объектов хозяйства, включая задачи планирования).
8. Разработка программы моделирования (построение моделей физических, поведенческих и других процессов, включая задачи прогнозирования и искусственного интеллекта).
9. Разработка программы управления информационной системой.
10. Разработка программы, организующей работу организации.
11. Реализация справочно-информационной системы.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

1. Классы и объекты. Организация ввода-вывода. Динамическое выделение памяти.
2. Дружественные функции. Перегрузка функций и операторов.
3. Наследование. Простое и множественное наследование.
4. Принцип полиморфизма. Виртуальные функции и абстрактные классы.
5. Шаблоны функций и классов.
6. Генерация и обработка исключительных ситуаций.
7. Поток ввода/вывода. Работа с файлами.
8. Классы-контейнеры, классы-итераторы, алгоритмы. STL-контейнеры.
9. Стандартные средства разработки в Java-экосистеме для построения web-сервисов.
10. Стандартные фреймворки на примере Spring для разработки программ на Java.
11. Конфигурация web -сервиса с помощью Spring MVC фреймворка.
12. Отладка и обработка ошибок средствами Java.
13. Основные структуры данных в Java.
14. Мультипоточность и синхронизация доступа к ресурсам средствами Java.
15. Средства функционального программирования в Java.
16. Работа с хранилищами данных средствами Java.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ КОМПЬЮТЕРНЫХ ПРОГРАММ

1. Операционные системы Microsoft Windows (8 и выше) и Linux.
2. Компилятор для языка программирования C++ (рекомендуемая среда разработки MS Visual Studio 2017(или новее).
3. Пакет Java SE Development Kit 8.
4. Среда разработки IntelliJ IDEA Community Edition или Eclipse IDE for Java Developers.

5. Средства сборки проекта Apache Maven.
6. Средства тестирования Apache JMeter.
7. Электронный ресурс по учебной дисциплине «Программирование на языках высокого уровня».
8. Класс современных персональных ЭВМ
9. Проектор (монитор) для демонстрации презентаций лекций и примеров кода программ.