

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
ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ**

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

6-05-0611-06 Системы и сети инфокоммуникаций

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 2025

СОСТАВИТЕЛИ:

Е.Г.Макейчик, старший преподаватель кафедры инфокоммуникационных технологий учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», магистр технических наук.

РЕЦЕНЗЕНТЫ:

Кафедра связи учреждения образования «Военная академия Республики Беларусь (протокол № 15 от 10.04.2024);

Д.А.Сикорский, заместитель генерального директора по научно-техническому развитию открытого акционерного общества «ПЕЛЕНГ», кандидат технических наук

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ В КАЧЕСТВЕ ПРИМЕРНОЙ:

Кафедрой инфокоммуникационных технологий учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № 17 от 28.05.2024);

Научно-методическим советом учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № 6 от 17.01.2025);

Научно-методическим советом по системам и сетям инфокоммуникаций Учебно-методического объединения по образованию в области информатики и радиоэлектроники (протокол № 4 от 10.06.2024).

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

ХАРАКТЕРИСТИКА УЧЕБНОЙ ДИСЦИПЛИНЫ

Примерная учебная программа по учебной дисциплине «Технологии программирования инфокоммуникационных систем» разработана для студентов учреждений высшего образования, обучающихся по специальности 6-05-0611-06 «Системы и сети инфокоммуникаций» в соответствии с требованиями образовательного стандарта общего высшего образования и примерного учебного плана вышеуказанной специальности.

Учебная дисциплина «Технологии программирования инфокоммуникационных систем» является одной из дисциплин, начинающих подготовку инженера по инфокоммуникациям. Учебная дисциплина предусматривает изучение основ объектно-ориентированного программирования и технологий программирования инфокоммуникационных систем.

Воспитательное значение учебной дисциплины «Технологии программирования инфокоммуникационных систем» 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 знаний об основных технологиях программирования инфокоммуникационных систем с использованием объектно-ориентированного подхода и применением инструментальных средств, языков программирования C++, Java.

Задачи учебной дисциплины:

освоение представления об объектно-ориентированном подходе к разработке программного обеспечения и средствах его реализации;

приобретение практических навыков по использованию объектно-ориентированных языков и соответствующих систем программирования для разработки программного обеспечения инфокоммуникационных систем;

освоение технологий программирования инфокоммуникационных систем.

Базовыми учебными дисциплинами по курсу «Технологии программирования инфокоммуникационных систем» являются «Основы алгоритмизации и программирования», «Веб-технологии в инфокоммуникациях».

В свою очередь учебная дисциплина «Технологии программирования инфокоммуникационных систем» является базой для таких учебных дисциплин компонента учреждения образования, как «Объектное проектирование сетевых приложений», «Автоматизация разработки программного обеспечения инфокоммуникаций», «Программное обеспечение сетевых смарт-устройств инфокоммуникаций».

ТРЕБОВАНИЯ К УРОВНЮ ОСВОЕНИЯ СОДЕРЖАНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

В результате изучения учебной дисциплины «Технологии программирования инфокоммуникационных систем» 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;

базовые профессиональные:

разрабатывать программные средства инфокоммуникаций с использованием языка C++;

разрабатывать программные средства инфокоммуникаций с использованием языка Java.

В результате изучения учебной дисциплины обучающийся должен:

знать:

концепцию, методы и технологии объектно-ориентированного программирования (ООП);

основные понятия, синтаксис и семантику конструкций языка программирования C++;

основные понятия, синтаксис и семантику конструкций языка программирования Java;

средства разработки объектно-ориентированных программ;

уметь:

реализовывать алгоритмы на языках программирования C++, Java;

использовать средства программирования для разработки и отладки программ на языке C++, Java.

иметь навык:

применения методов и технологий объектно-ориентированного программирования;

работы со средствами разработки и отладки программ на языках C++, Java.

Примерная учебная программа рассчитана на 222 учебных часа, из них – 98 аудиторных. Примерное распределение аудиторных часов по видам занятий: лекции – 16 часов, лабораторные занятия – 48 часов, практические занятия – 34 часа.

ПРИМЕРНЫЙ ТЕМАТИЧЕСКИЙ ПЛАН

Наименование раздела, темы	Всего аудиторных часов	Лекции	Лабораторные занятия	Практические занятия
Раздел 1. Технологии программирования C++	48	8	32	8
Тема 1. Введение в программирования на языке C++. Классы. Конструкторы и деструкторы классов	12	2	8	2
Тема 2. Динамическая память. Дружественные функции и дружественные классы. Перегрузка операторов	12	2	8	2
Тема 3. Наследование, полиморфизм, абстракция	12	2	8	2
Тема 4. Шаблоны функций и классов. Исключительные ситуации	12	2	8	2
Раздел 2. Технологии программирования Java	50	8	16	26
Тема 5. Введение в программирования на языке Java. Основы языка Java	10	2	4	4
Тема 6. Классы в языке Java	14	2	4	8
Тема 7. Наследование и интерфейсы. Обработка исключительных ситуаций	14	2	4	8
Тема 8. Универсальные типы. Коллекции. Потоки	12	2	4	6
Итого:	98	16	48	34

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

РАЗДЕЛ 1. ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ C++

Тема 1. ВВЕДЕНИЕ В ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C++. КЛАССЫ. КОНСТРУКТОРЫ И ДЕКТРУКТОРЫ КЛАССОВ

Современные технологии программирования. Преимущества ООП. Эволюция языков программирования. Среда разработки. Концепции ООП в языке C++: инкапсуляция, наследование, полиморфизм, абстракция.

Понятие класса. Объявление и определение элементов класса. Управление доступом к элементам класса. Создание объекта (экземпляра) класса. Указатель на объект класса.

Назначение конструкторов и деструкторов. Перегрузка конструкторов. Виды конструкторов (конструктор по умолчанию, конструктор с параметрами). Правила объявления и определения конструкторов и деструкторов при создании классов. Работа с динамической памятью.

Тема 2. ДИНАМИЧЕСКАЯ ПАМЯТЬ. ДРУЖЕСТВЕННЫЕ ФУНКЦИИ И ДРУЖЕСТВЕННЫЕ КЛАССЫ. ПЕРЕГРУЗКА ОПЕРАТОРОВ

Указатели и динамическая память. Динамические массивы. Назначение дружественных функций и дружественных классов. Правила объявления и определения дружественных функций и дружественных классов.

Возможность переопределения операторов в языке C++. Синтаксис объявления и определения операторов в классах. Создание функций операторов сложения, присваивания и других. Ограничения на перегрузку операторов.

Тема 3. НАСЛЕДОВАНИЕ, ПОЛИМОРФИЗМ, АБСТРАКЦИЯ

Понятие наследования, его назначение. Базовые и производные классы, синтаксис определения производных классов. Одиночное и множественное наследование. Особенности конструкторов производных классов.

Понятие полиморфизма. Виртуальные методы: назначение, правила использования. Понятие абстракции. Чисто виртуальные методы и абстрактные классы.

Тема 4. ШАБЛОНЫ ФУНКЦИЙ И КЛАССОВ. ИСКЛЮЧИТЕЛЬНЫЕ СИТУАЦИИ

Назначение шаблонов, понятие параметрического полиморфизма. Шаблоны функций: объявление шаблона, создание экземпляра шаблона.

Понятие исключительной ситуации. Понятие исключения. Средства языка C++ для обработки исключительных ситуаций.

Раздел 2. ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ JAVA

Тема 5. ВВЕДЕНИЕ В ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ JAVA. ОСНОВЫ ЯЗЫКА JAVA

Виртуальная машина Java. Среда разработки. Особенности объектно-ориентированного программирования на Java. Документация по классам Java.

Типы данных языка Java. Операторы языка Java. Вычисление выражений. Приведение встроенных типов. Массивы в Java.

Тема 6. КЛАССЫ В ЯЗЫКЕ JAVA

Общая форма класса. Объявление объектов. Конструкторы. Сборка мусора. Перегрузка методов. Перегрузка конструкторов. Классы-оболочки. Автоупаковка и автораспаковка. Строки и числа. Классы регулярных выражений в Java.

Тема 7. НАСЛЕДОВАНИЕ И ИНТЕРФЕЙСЫ. ОБРАБОТКА ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦИЙ

Наследование и доступ к членам класса. Конструкторы и наследование. Абстрактные классы. Класс Object. Интерфейсы. Пакеты и ограничение доступа. Внутренние (inner) классы. Вложенные (nested) классы. Анонимные (anonymous) классы.

Исключения в Java. Типы исключений. Неперехваченные исключения.

Тема 8. УНИВЕРСАЛЬНЫЕ ТИПЫ. КОЛЛЕКЦИИ. ПОТОКИ

Общие сведения об универсальных типах. Универсальные типы (дженерики). Лямбда выражения. Функциональный интерфейс. Коллекции. Потоки данных. Многопоточность. Синхронизация потоков. Межпоточковые коммуникации.

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

ЛИТЕРАТУРА

ОСНОВНАЯ

1. Шилдт, Г. Полное руководство. Java / Г. Шилдт – 12-е изд. – Москва : Диалектика, 2022. – 1344 с.
2. Макейчик, Е. Г. Объектно-ориентированное программирование [+ электр. вариант] : учебно-методическое пособие / Е. Г. Макейчик, В. В. Чепикова, О. Г. Шевчук. – Минск : БГУИР, 2020. – 80 с.
3. Эккель, Б. Философия Java / Б. Эккель. - 4-е полное изд. – Санкт-Петербург : Питер, 2018. – 1168 с.

ДОПОЛНИТЕЛЬНАЯ

5. Портал о программировании [Электронный ресурс]. – Режим доступа: <https://code-live.ru/tag/cpp/> – Дата доступа: 01.04.2024.
6. Самоучитель по Основам Java [Электронный ресурс]. – Режим доступа: <https://javarush.ru/> – Дата доступа: 01.04.2024.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

При изучении учебной дисциплины рекомендуется использовать следующие формы самостоятельной работы:

- выполнение лабораторных работ;
- выполнение курсового проекта;
- изучение дополнительного материала;
- решение задач к практическим занятиям;
- повторение пройденного теоретического материала.

ПЕРЕЧЕНЬ РЕКОМЕНДУЕМЫХ СРЕДСТВ ДИАГНОСТИКИ КОМПЕТЕНЦИЙ ОБУЧАЮЩИХСЯ

Примерным учебным планом по специальности 6-05-0611-06 «Системы и сети инфокоммуникаций» в качестве формы промежуточной аттестации по учебной дисциплине «Технологии программирования инфокоммуникационных систем» рекомендуется зачет и экзамен. Оценка учебных достижений обучающихся производится по системе «зачтено/не зачтено» и по десятибалльной шкале.

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

- отчеты по лабораторным работам с их устной защитой;
- тестирование;
- проведение текущих контрольных опросов по определенным темам;
- выполнение индивидуальных заданий в рамках курсового проектирования.

РЕКОМЕНДУЕМЫЕ МЕТОДЫ (ТЕХНОЛОГИИ) ОБУЧЕНИЯ

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

объяснительно-иллюстративное обучение, реализуемое на лекциях с использованием мультимедийного оборудования;

учебно-исследовательская деятельность и творческий подход, реализуемые на лабораторных занятиях, при курсовом проектировании.

РЕКОМЕНДАЦИИ ПО КУРСОВОМУ ПРОЕКТИРОВАНИЮ

Курсовой проект является элементом учебной дисциплины «Технологии программирования инфокоммуникационных систем» и выполняется с целью проверки полученных знаний обучающимися и диагностики их способности применять знания на практике.

Основная цель курсового проекта состоит в изучении технологий программирования на основе объектно-ориентированной методологии, приобретении навыков разработки программных средств на языке C++. Примерный объем текста пояснительной записки курсового проекта 30-45 страниц. Итогом курсового проектирования является программное средство, разработанное с использованием методологии ООП.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ КУРСОВЫХ ПРОЕКТОВ

Разработка программного средства (ПС) с использованием методологии объектно-ориентированного программирования:

ПС для складского учета;

ПС для конвертации валют;

ПС для подсчета калорий;

ПС для решения конкретной математической задачи;

ПС для подсчета количества операций за указанный период времени;

ПС для подсчета стоимости товара за указанный период;

ПС для подсчета стоимости товара с учетом скидки;

ПС для подсчета дней до определенной даты.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

1. Введение в тип данных Класс. Создание приложения с использованием Qt.

2. Использование конструктора и деструктора. Работа с динамической памятью. Основы работы с GUI.

3. Массивы объектов. Работа с динамической памятью. Слоты, сигналы и события.

4. Двумерный динамический массив. Дружественные функции. Чекбоксы и радиобаттоны в Qt.

5. Динамическая строка. Перегрузка операторов. База данных в Qt.
6. Наследование и механизм виртуальных функций. Клиент-серверное приложение Qt Основы языка Java.
7. Шаблоны. Соединение классов в Qt.
8. Исключительные ситуации. Запуск Qt приложений .exe.
9. Универсальные типы. Коллекции.
10. Основы языка Java.
11. Классы Java.
12. Наследование и интерфейсы. Обработка исключительных ситуаций.
13. Универсальные типы. Коллекции. Потоки.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

1. IDE Visual Studio. Фреймворк Qt. IDE Qt Creator.
2. Структура программ. Отладка программ. UML-диаграммы классов. Блок-схемы.
3. Введение в Java. Основы языка. Решение задач.
4. Классы Java. Решение задач.
5. Наследование и интерфейсы. Обработка исключительных ситуаций. Решение задач.
6. Универсальные типы. Коллекции. Потоки. Решение задач.

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

1. Microsoft Visual Studio.
2. Фреймворк Qt.
3. Qt Creator.
4. IntelliJ IDEA.
5. Java Development Kit.
6. Microsoft Office.