АННОТАЦИЯ

Дисциплины

«ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ»

Специальность – 10.05.03 «Информационная безопасность автоматизированных систем»

Квалификация (степень) выпускника – специалист по защите информации

Специализация – «Информационная безопасность автоматизированных систем на транспорте»

**1. Место дисциплины в структуре основной профессиональной образовательной программы**

Дисциплина «Функциональное программирование» (Б1.В.ДВ.5.2) относится к вариативной части и является дисциплиной по выбору обучающегося.

**2. Цель и задачи дисциплины**

Целью изучения дисциплины «Функциональное программирование» (Б1.В.ДВ.5.2) является получение студентами знаний и навыков, необходимых для разработки кроссплатформенного прикладного программного обеспечения, а также программного обеспечения для встраиваемых систем и мобильных платформ.

Для достижения поставленной цели решаются следующие задачи:

* обучение студентов принципам построения и сопровождения программных продуктов, моделям программирования и паттернам проектирования.
* обучение студентов практическому применению технологий и средств прикладного программирования, предоставляемых современными языками программирования и средами разработки.
* обучение студентов разработке кроссплатформенного программного обеспечения с графическим интерфейсом пользователя.
* обучение студентов основам разработки программного обеспечения для встраиваемых систем и мобильных платформ.

**3. Перечень планируемых результатов обучения по дисциплине**

Планируемыми результатами обучения по дисциплине являются: приобретение знаний, умений, навыков и/или опыта деятельности.

В результате освоения дисциплины обучающийся должен:

**ЗНАТЬ**:

* приемы объектно-ориентированного проектирования;
* различия при решении поставленной задачи для разных операционных систем;
* возможности, предоставляемые языками C++ и Python, для решения задач прикладного программирования;
* виды задач, решаемых с помощью набора библиотек Qt;
* возможности основных классов и модулей набора библиотек Qt, применяемые при разработке;
* правила оформления исходного кода программы;
* часто встречающиеся и возможные ошибки, а также методы их поиска при разработке прикладных программ;
* средства разработки прикладных программ;
* средства отладки прикладных программ.

**УМЕТЬ**:

* составлять требования к разрабатываемой программе;
* принимать обоснованное решение относительно выбора инструментария и технологий для решения поставленных задач;
* определять оптимальный вариант решения задачи при наличии альтернатив;
* решать поставленные задачи по другим изучаемым курсам с применением технологий и средств прикладного программирования.

**ВЛАДЕТЬ**:

* навыками проектирования архитектуры программного обеспечения;
* навыками разработки программного обеспечения с применением современных технологий и инструментальных средств;
* навыками разработки интерфейса пользователя;
* навыками тестирования прикладных программ;
* навыками составления документации в соответствии с современными требованиями.

Изучение дисциплины направлено на формирование следующих **общепрофессиональных компетенций (ОПК)**:

* способностью применять языки, системы и инструментальные средства программирования в профессиональной деятельности (ОПК-3);
* способностью применять методы научных исследований в профессиональной деятельности, в том числе в работе над междисциплинарными и инновационными проектами (ОПК-5);
* способностью к освоению новых образцов программных, технических средств и информационных технологий (ОПК-8).

Область профессиональной деятельности обучающихся, освоивших данную дисциплину, приведена в п. 2.1 ОПОП.

Объекты профессиональной деятельности обучающихся, освоивших данную дисциплину, приведены в п. 2.2 ОПОП.

**4. Содержание и структура дисциплины**

1. Введение в паттерны проектирования

Паттерны проектирования. Описание паттернов. Порождающие паттерны. Структурные паттерны. Паттерны поведения. Применение паттернов при разработке прикладного программного обеспечения.

2. Общие сведения о языке С++

Основные элементы языка С++: алфавит, переменные, константы, типы данных, основные операции, стандартные функции, структура программы и средства ввода-вывода данных. Операторы управления. Статические и динамические массивы и матрицы.

3. Язык Python и его особенности

Основные элементы языка Python: алфавит, переменные, константы, типы данных, основные операции, стандартные функции, структура программы и средства ввода-вывода данных. Операторы управления. Статические и динамические массивы и матрицы. Особенности языка Python.

4. Знакомство с Qt. Подготовка к работе

Основные сведения о наборе библиотек Qt. Использование библиотек Qt для разработки прикладных программ на языке С++. Особенности и реализация PyQt для языка Python. Ознакомление с процессом подготовки окружения для разработки. Описание среды разработки Qt Creator и вспомогательного инструментария разработчика.

5. Создание графического интерфейса средствами Qt

Изучение виджетов - визуальных элементов, из которых состоит графический интерфейс пользователя, их компоновка, политика размеров, сигнально-слотовые соединения, элементы графического интерфейса и их использование.

6. Собственные классы в Qt. Создание элементов графического интерфейса

Изучение базового объекта QObject. Управления памятью. Иерархия объектов. События и их обработка. Фильтры и распространение событий. Создание собственного элемента интерфейса.

7. Разработка Web-приложений

Обсуждается одна из основных сфер прикладного программирования: разработка распределенных сетевых приложений с Web-интерфейсом. Рассматриваются технологии CGI и Servlet, вопросы безопасности сетевых приложений.

8. Работа с базой данных

Вопросы проектирования структуры баз данных. Обработка хранимых в базе данных. Взаимодействие с базой данных через высокоуровневые интерфейсы языков программирования. Основы языка SQL.

9. Многопоточные вычисления

Построение многопоточных приложений. Распараллеливание обработки данных. Модель конкурентного взаимодействия потоков.

10. Создание приложений с графическим интерфейсом пользователя

Основы проектирования графических интерфейсов пользователя. Рассмотрение архитектур графических подсистем. Сравнение графических библиотек.

11. Обработка текстов. Регулярные выражения. Unicode

Обработка текстовой информации. Алгоритмы над строками. Синтаксис и семантика регулярных выражений. Использование Unicode.

12. Общие сведения о библиотеке MathGL

Рассмотрение вопросов визуализации данных и построения графических зависимостей. Использование кроссплатформенной библиотеки MathGL.

13. Элементы функционального программирования

Рассмотрение вопросов построения приложений с использованием функциональной парадигмы программирования. Функциональные и рекурсивные алгоритмы.

**5. Объем дисциплины и виды учебной работы**

Объем дисциплины – 5 зачетные единицы (180 час.), в том числе:

лекции – 34 час.

практические занятия – 16 час.

лабораторные занятия – 34 час.

самостоятельная работа – 51 час.

Форма контроля знаний – экзамен, курсовая работа в 6-ом семестре.