

Белорусский государственный университет

УТВЕРЖДАЮ
Проректор по учебной работе и

Проректор по учебной работе и образовательным инновациям

0.1 : 1000000000

«05» июля 2023 г.

Регистрационный № - 1539/б.

РАЗРАБОТКА КРОСС-ПЛАТФОРМЕННЫХ ПРИЛОЖЕНИЙ

**Учебная программа учреждения высшего образования
по учебной дисциплине для специальности:**

6-05-0533-09 Прикладная математика

Учебная программа составлена на основе ОСВО 6-05-0533-09-2023 специальности 6-05-0533-09 «Прикладная математика», примерного учебного плана, регистрационный № 6-05-05-020/пр. от 20.12.2022, учебных планов БГУ: № 6-5.3-57/01 от 15.05 2023 г., № 6-5.3-57/02 от 15.05 2023 г., № 6-5.3-57/03 от 15.05 2023 г., № 6-5.3-57/04 от 15.05 2023 г.

СОСТАВИТЕЛИ:

Пазюра Е.В., старший преподаватель кафедры технологий программирования факультета прикладной математики и информатики Белорусского государственного университета.

Карпович Н.А., старший преподаватель кафедры технологий программирования факультета прикладной математики и информатики Белорусского государственного университета;

Зенько Т.А., старший преподаватель кафедры технологий программирования факультета прикладной математики и информатики Белорусского государственного университета.

РЕЦЕНЗЕНТЫ:

П. В. Гляков, профессор кафедры информационных технологий в культуре учреждения образования «Белорусский государственный университет культуры и искусств», кандидат физико-математических наук, доцент.

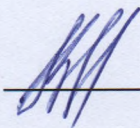
В. М. Котов, заведующий кафедрой дискретной математики и алгоритмики факультета прикладной математики и информатики Белорусского государственного университета доктор физико-математических наук, профессор.

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой технологий программирования БГУ
(протокол № 18 от 16.05.2023)

Научно-методическим советом БГУ
(протокол № 9 от 29.06.2023)

Заведующий кафедрой



А.Н. Курбацкий

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Цели и задачи учебной дисциплины

Цель учебной дисциплины «Разработка кросс-платформенных приложений» – теоретическая и практическая подготовка, обеспечивающая получение знаний по кросс-платформенному программированию, формирование практических навыков разработки кросс-платформенных решений, применения инструментальных средств разработки и отладки кросс-платформенного программного обеспечения, использования объектно-ориентированного и функционального подходов к решению практических задач.

Задачи учебной дисциплины

1. Подготовка специалиста, уверенно владеющего разнообразными возможностями, предоставляемыми современными компьютерными технологиями и грамотно ориентирующегося в них.
2. Обучение студентов работе со средой программирования, предназначенной для разработки кросс-платформенных решений.
3. Овладение методологией разработки программ на основе императивного и объектно-ориентированного подхода.
4. Формирование навыков разработки современного востребованного кросс-платформенного программного обеспечения.

Место учебной дисциплины. В системе подготовки специалиста с высшим образованием учебная дисциплина относится **к модулю «Программирование»** государственного компонента.

Учебная программа составлена с учетом межпредметных **связей** и программ по дисциплинам: «Основы и методологии программирования», «Машинно-ориентированное программирование», «Промышленное программирование», «Технологии программирования», которые являются базовыми при изучении всех дисциплин специальности, при выполнении курсовых и дипломных работ, а также используются как инструментарий для моделирования и компьютерного решения задач математических дисциплин.

Требования к компетенциям

Освоение учебной дисциплины «Разработка кросс-платформенных приложений» должно обеспечить формирование следующих компетенций:

Универсальные компетенции:

УК-2. Решать стандартные задачи профессиональной деятельности на основе применения информационно-коммуникационных технологий.

Базовые профессиональные компетенции:

БПК-4. Применять навыки построения, анализа и тестирования алгоритмов и программ для решения типовых задач прикладной математики

БПК-5. Применять при проектировании приложений такие парадигмы программирования, как структурное, объектно-ориентированное и

функциональное программирование, а также иные парадигмы, разрабатывать программное обеспечение в интегрированных средах разработки.

В результате освоения учебной дисциплины студент должен:

знать:

- основные аспекты концепции кросс-платформенного программирования;
- архитектуру и компоненты кросс-платформенных решений;
- особенности программирования для кросс-платформенных решений;
- концепцию и особенности применения функционального программирования;
- особенности программирования с использованием основных компонентов фреймворка Qt;

уметь:

- разрабатывать программное обеспечение, используя кросс-платформенный подход;
- применять функциональный подход при разработке программного обеспечения;

владеть:

- навыками работы с объектно-ориентированными языками программирования;
- инструментарием разработки кросс-платформенных приложений.

Структура учебной дисциплины

Дисциплина изучается во 2-м семестре. Всего на изучение учебной дисциплины «Разработка кросс-платформенных приложений» отведено:

- для очной формы получения высшего образования отведено 216 часов, в том числе 102 аудиторных часа, из них: лекций – 42 часа, лабораторных занятий – 54 часа, управляемая самостоятельная работа – 6 часов.

Трудоемкость учебной дисциплины составляет 6 зачетных единиц.

Форма промежуточной аттестации – экзамен.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. Концепции и современные средства кросс-платформенного программирования

Тема 1.1. Понятие кроссплатформенного программирования.

Определение кросс-платформенности. Уровни кросс-платформенности: аппаратный, программный, компиляции, выполнения. Проблемы кросс-платформенной разработки.

Языки, реализующие кросс-платформенность на уровне компиляции: C, C++. Языки, реализующие кросс-платформенность на уровне выполнения: Java, JavaScript, PHP, C#. Кросс-платформенные среды разработки.

Подходы к кросс-платформенному программированию. Кросс-платформенный пользовательский интерфейс и проблемы его создания.

Интегрированная среда разработки QtCreator. Модульная архитектура и классы Qt. Особенности. Основные составляющие. Пространство имен.

Создание простого приложения с использованием Qt.

Тема 1.2. Объектно-ориентированное программирование – инструмент для кроссплатформенного программирования.

Класс как абстрактный тип, классы и объекты. Семантика перемещения (move-семантика) для оптимизации работы с объектами в C++.

Ссылки на объекты классов. Lvalue и Rvalue – значение. Rvalue-reference – ссылка. Макет базового класса с move- конструктором и move- operator=.

Тема 1.3. Работа со строковыми данными.

Особенности QT строк. Сравнение с классом string из библиотеки stl. Регулярные выражения в стандартном C++ и в фреймворке QT. Использование текстовых файлов. Codec для представления символов в разных кодовых таблицах.

Регулярные выражения в стандартном C++ и в фреймворке QT. Использование текстовых файлов.

Тема 1.4. Структуры данных и основные операции над ними.

Пример разработки комплекса дружественных классов «контейнер стек» и «итератор стека». Вложенные классы. Разработка класса «очередь». Сравнение классов «стек» и «очередь» в stl и в QT.

Обзор структур данных, лежащих в основе оптимальных алгоритмов работы с данными в stl.

Поисковое дерево, двоичная куча. Оценка трудоемкости методов построения и поиска.

Тема 1.5. Шаблоны как поддержка обобщенного программирования.

Множественная реализация функций в QT C++. Методы решения этой проблемы. Определение шаблонов функций. Параметры шаблонов функций. Выведение типа параметров шаблона по типам аргументов при вызове функции. Переопределение шаблонов функций.

Полиморфизм времени компиляции (параметрический полиморфизм).

Определение шаблонов классов. Описание и конкретизация шаблона. Требования к параметрам шаблона. Явное и неявное указание параметров при конкретизации шаблонов. Специализация шаблонов. Создание объектов по шаблонам.

Особенности компиляции шаблонов. Возникновение ошибок компиляции.

Тема 1.6. Наследование и полиморфизм как базовые понятия в парадигме объектно-ориентированного программирования.

Основные принципы и правила наследования. Понятие производного класса. Базовый класс и атрибуты его доступа. Иерархия производных классов. Конструкторы производных классов. Реализация наследования в языке программирования C++. Параметризация и наследование.

Виды полиморфизма. Основные принципы и правила полиморфизма. Виртуальные функции. Виртуальные деструкторы. Чисто виртуальные функции, абстрактные классы. Иерархия классов. Использование указателя и ссылки базового класса для доступа к объектам потомков. Указатели объектов производного и базовых классов. Специальный полиморфизм через виртуальные функции производных классов, таблица виртуальных методов, позднее связывание,

Абстрактные классы. Виртуальные базовые классы. Множественное наследование.

Тема 1.7. Обработка файлов. Сериализация объектов.

Работа с файловыми потоками ввода/вывода. Использование текстовых файлов. Использование бинарных файлов. Особенности использования классов для работы с файлами.

Потоки ввода/вывода. Назначение класса QTextStream. Понятие сериализации объектов. Класс QDataStream как основа сериализации.

Тема 1.8. Библиотеки контейнерных классов.

Контейнерные классы stl. Итераторы.

Последовательные контейнеры вектор, массивы байтов, списки, стек, очередь. Соответствие классов stl и QT.

Ассоциативные контейнеры: словари, хэш-таблицы, множества. Соответствие классов stl и QT.

Алгоритмы: сортировка, поиск, сравнение, заполнение значениями, копирование значений элементов, подсчет значений.

Раздел 2. Разработка графического интерфейса для кроссплатформенных приложений

Тема 2.1. Разработка графического интерфейса средствами QT.

Разработка приложений, управляемых событиями. Философия объектной модели. Организация графического интерфейса. Механизм слотов и сигналов. Быстрая разработка прототипов в QtDesigner.

Базовые элементы пользовательского интерфейса, предоставляемые QT. Виджеты – элементы управления для организации графического интерфейса. Иерархия виджетов. Политики размера. Элементы отображения. Надписи. Индикатор выполнения. Кнопки, флажки и переключатели. Основные классы для организации графического интерфейса.

Тема 2.2. Управление автоматическим размещением элементов.

Управление автоматическим размещением элементов. Менеджеры компоновки. Горизонтальное и вертикальное размещение. Классы для управления элементами управления.

Элементы ввода. Однострочное текстовое поле. Редактор текста.

Элементы выбора как базовые классы архитектуры «модель-представление».

Тема 2.3. Диалоговые окна

Правила создания диалоговых окон. Классы для организации диалоговых окон. Модальные диалоговые окна. Немодальные диалоговые окна. Создание собственного диалогового окна.

Стандартные диалоговые окна: выбора файлов, цвета, шрифта, окно ввода.

Тема 2.4. Проектирование модель - отображение

Реализация шаблона проектирования модель-отображение для табличных данных и виджеты их отображения. Использование абстрактной модели для настройки данных, предназначенных для табличного отображения.

Тема 2.5 . Разработка однооконных (SDI) и многооконные (MDI) приложений.

Программирование приложений, основанных на обработке событий. Цикл обработки сообщений. Модель событий, класс события.

Сосуществование двух механизмов: «слоты и сигналы» и «события». Переопределение специализированных методов обработки событий, события клавиатуры, событие обновления контекста рисования. События мыши.

Класс главного окна. Класс действия. Панель инструментов. Строка состояния. Создание меню. «Анатомия» меню. Виды меню.

SDI- приложение, MDI-приложение. Создание SDI- и MDI-приложений.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Очная форма получения высшего образования с применением
дистанционных образовательных технологий (ДОТ)

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов		Количество часов УСП	Форма контроля знаний
		Лекции	Лабораторные занятия		
1	2	3	4	5	6
	Разработка кросс-платформенных приложений	42	54	6	
1.	Концепции и современные средства кросс-платформенного программирования	28	44	4	
1.1.	Понятие кроссплатформенного программирования	2			
1.2.	Объектно-ориентированное программирование – инструмент для кроссплатформенного программирования	4	8		Отчет по лабораторной работе
1.3.	Работа со строковыми данными.	4	8	2	Отчет по лабораторной работе, устный опрос
1.4.	Структуры данных и основные операции над ними.	4	8		Отчет по лабораторной работе
1.5.	Шаблоны как поддержка обобщенного программирования	4	6		Отчет по лабораторной работе
1.6.	Наследование и полиморфизм как базовые понятия в парадигме объектно-ориентированного программирования	4	8		Коллоквиум по темам 1.1-1.6
1.7.	Обработка файлов. Сериализация объектов.	4	4		Отчет по лабораторной работе

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов		Количество часов УСР	Форма контроля знаний
		Лекции	Лаборатор ные занятия		
1.8.	Библиотека контейнерных классов	2	2	2	Контрольная работа по разделу 1. Собеседование
2.	Разработка графического интерфейса для кроссплатформенных приложений	14	10	2	
2.1.	Создание графического интерфейса средствами QT	4	4		Устный опрос
2.2.	Управление автоматическим размещением элементов.	2		2	Устный опрос
2.3.	Диалоговые окна	2	2		Устный опрос
2.4.	Проектирование модель – отображение	4	2		Дискуссия
2.5.	Разработка однооконных (SDI) и многооконных (MDI)-приложений	2	2		Отчет по лабораторной работе

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Перечень основной литературы

1. Фримен, Э. HeadFirst. Паттерны проектирования / Эрик Фримен, Элизабет Робсон [при участии] Кэти Сьерра и Берта Бейтса ; [пер. с англ. Е. Матвеев]. – 2-е изд. – Санкт-Петербург ; Москва ; Минск : Питер, 2022. – 633 с.
2. Лафоре, Р. Объектно-ориентированное программирование в C++ / Р. Лафоре ; [пер. с англ.: А. Кузнецов, М. Назаров, В. Шрага]. – 4-е изд. – Санкт-Петербург ; Москва ; Минск : Питер, 2022. – 923 с. – URL: <https://ibooks.ru/reading.php?short=1&productid=376836>.
3. Прохоренок, Н. А. Qt 6. Разработка оконных приложений на C++ / Николай Прохоренок. – Санкт-Петербург : БХВ-Петербург, 2022. – 512 с. – URL: <https://ibooks.ru/bookshelf/386517>.
4. Тидвелл, Д. Разработка интерфейсов. Паттерны проектирования / Дж. Тидвелл, Ч. Брюэр, Э. Валенсия ; [пер. с англ. яз. С. Чирников]. – 3-е изд. – Санкт-Петербург ; Москва ; Минск : Питер, 2022. – 557 с. – URL: <https://ibooks.ru/bookshelf/386796>.
5. Пикус, Ф. Г. Идиомы и паттерны проектирования в современном C++ : применение современных паттернов проектирования к решению типичных задач на C++ для построения надежных приложений / Федор Г. Пикус ; [пер. с англ. А. А. Слинкина]. – Москва : ДМК Пресс, 2020. – 451 с.

Перечень дополнительной литературы

1. Паттерны проектирования на платформе .NET / Тепляков С. – СПб.: Питер, 2015. – 320 с.
2. Программирование на языке C++ в среде Qt Creator: Курс лекций / Е.Р. Алексеев, Г.Г. Злобин, Д.А. Костюк [и др.] — Москва : Интуит НОУ, 2016. — 715 с.
3. Qt 5.10. Профессиональное программирование на C++ / Шлее М. – СПб.: БХВ-Петербург, 2022. – 1072 с.
4. Создание пользовательских интерфейсов в программах на C++ с использованием библиотеки Qt / Иванова Г.С. – Учебное пособие. – М. – 2011. – 52 с.

Электронные ресурсы

1. Погружение в паттерны проектирования / Швец А. – Электронная книга о принципах и паттернах проектирования [Электронный ресурс]. – Режим доступа: <https://refactoring.guru/ru/design-patterns/book>
2. Все о кросс-платформенном программировании / Справочная документация по Qt (Выпуск OpenSource) [Электронный ресурс]. –

Режим доступа: <http://doc.crossplatform.ru/qt/4.3.2/index.html>

3. Руководство по Qt для начинающих / Князев А.А.[Электронный ресурс]. – Режим доступа: <http://knzsoft.ru/qt-bgr/>
4. Образовательный портал БГУ [Электронный ресурс]. – Режим доступа: <https://edufpmi.bsu.by/course/view.php?id=305>
5. Образовательная платформа InsightRunner [Электронный ресурс]. – Режим доступа: <https://acm.bsu.by>

Перечень рекомендуемых средств диагностики и методика формирования итоговой отметки

Объектом диагностики компетенций студентов являются знания, умения, полученные ими в результате изучения учебной дисциплины. Выявление учебных достижений студентов осуществляется с помощью мероприятий текущего контроля и промежуточной аттестации.

Для диагностики компетенций могут использоваться следующие средства текущего контроля:

- отчет по лабораторным работам (защита выполненных на лабораторных занятиях индивидуальных заданий);
- собеседование, устный опрос, дискуссия;
- контрольная работа, коллоквиум.

В качестве рекомендуемых технических средств диагностики используется обучение, организованное на платформе Moodle (<https://edufpmi.bsu.by>).

Формой промежуточной аттестации по дисциплине учебным планом предусмотрен **экзамен**.

При формировании итоговой отметки используется рейтинговая система оценки знаний студента, дающая возможность проследить и оценить динамику процесса достижения целей обучения. Рейтинговая система предусматривает использование весовых коэффициентов в ходе проведения контрольных мероприятий текущей аттестации.

Примерные весовые коэффициенты, определяющие вклад текущей аттестации в отметку при прохождении промежуточной аттестации:

Формирование отметки за текущую аттестацию:

- контрольная работа – 40 %;
- отчет по лабораторным работам – 40 %;
- коллоквиум – 20 %.

Итоговая отметка по дисциплине рассчитывается на основе отметки текущей аттестации (рейтинговой системы оценки знаний) – 40% и экзаменационной отметки – 60%.

Примерный перечень заданий для управляемой самостоятельной работы студентов

В качестве заданий для управляемой самостоятельной работы могут быть выданы задания для самостоятельного решения задач по следующим темам:

Занятие 1. Тема 1.3. Работа со строковыми данными: Классы QT для работы со строками. (2 ч.)

- привести примеры инициализации объекта QString с применением разных конструкторов;
- описать методы класса QString взаимного преобразования строковых и числовых данных;
- описать методы класса QString для поиска, вставки и замены подстроки в строке;

Форма контроля – устный опрос по изученной теме

Занятие 2. Тема 1.8. Библиотека контейнерных классов. Классы QT для работы с последовательными и абстрактными контейнерами. (2 ч.)

- описать классы для работы с последовательными контейнерами вектор, список, дек; сравнить их с классами библиотеки stl;
- описать классы для работы с контейнерами-адаптерами – стек и очередь;
- описать классы для работы с ассоциативными контейнерами, их реализацию с помощью древовидной структуры, указать трудоемкость операций добавления и извлечения элементов.

Форма контроля – собеседование.

Занятие 3. Тема 2.2. Управление автоматическим размещением элементов на форме (2 ч.).

- описать класс для горизонтального размещения управляющих элементов на форме QHBoxLayout;
- описать класс для вертикального размещения управляющих элементов на форме QVBoxLayout;
- описать класс для вертикального размещения управляющих элементов на форме QGridLayout.

Форма контроля – устный опрос.

Примерная тематика лабораторных занятий

Занятие №1. Объектно-ориентированное программирование. Разработка классов

Занятие №2. Работа со строками. Регулярные выражения.

Занятие №3. Разработка шаблонного класса.

Занятие №4. Разработка контейнерного класса.

Занятие №5. Файлы, потоки ввода/вывода, сериализация.

Занятие №6. Создание оконных, SDI- и MDI-приложений .

Примерная тематика контрольной работы и коллоквиума

Контрольная работа «Решение задач с использованием библиотеки классов STL и QT» по **разделу 1. Концепции и современные средства кросс-платформенного программирования.** (2 ч)

Коллоквиум «Разработка пользовательских классов с реализацией наследования и полиморфизма».

Описание инновационных подходов и методов к преподаванию учебной дисциплины

При организации образовательного процесса используются следующие методы:

– **метод группового обучения**, который представляет собой форму организации учебно-познавательной деятельности обучающихся, предполагающую функционирование разных типов малых групп, работающих как над общими, так и специфическими учебными заданиями.

В качестве технических средств для организации работы в рамках учебной дисциплины рекомендуется использовать Образовательный портал БГУ (<https://edufpmi.bsu.by>) – инструмент с эффективной функциональностью контроля, тренинга и самостоятельной работы;

– **практико-ориентированный подход**, который предполагает освоение содержания образования через решения практических задач; приобретение навыков эффективного выполнения разных видов профессиональной деятельности; ориентацию на генерирование идей, реализацию групповых студенческих проектов; использование процедур, способов оценивания, фиксирующих профессиональные компетенции;

– **метод учебной дискуссии**, который предполагает участие студентов в конце курса в целенаправленном обмене мнениями, идеями для предъявления и/или согласования существующих позиций по определенной проблеме. Использование метода обеспечивает появление нового уровня понимания изучаемой темы, применение знаний (теорий, концепций) при решении проблем, определение способов их решения.

Методические рекомендации по организации самостоятельной работы обучающихся

Для организации самостоятельной работы студентов по учебной дисциплине следует использовать современные информационные ресурсы: разместить на образовательном портале комплекс учебных и учебно-методических материалов (учебно-программные материалы, учебное издание для теоретического изучения дисциплины, методические указания к лабораторным занятиям, материалы текущего контроля и промежуточной

аттестации, позволяющие определить соответствие учебной деятельности обучающихся требованиям образовательного стандарта высшего образования и учебно-программной документации, в т.ч. вопросы для подготовки к экзамену, задания, тесты, вопросы для самоконтроля и др., список рекомендуемой литературы, информационных ресурсов и др.).

При составлении заданий по учебной дисциплине необходимо предусмотреть возрастание их сложности: от заданий, формирующих достаточные знания по изученному учебному материалу на уровне узнавания, к заданиям, формирующим компетенции на уровне воспроизведения, и далее к заданиям, формирующим компетенции на уровне применения полученных знаний.

Задания на уровне узнавания

Задание 1. В чем особенность специализации шаблонов для типа `char*`?

Задание 2. В чем различие между последовательным и ассоциативным контейнером. Приведите примеры.

Задание 3. Назначение модели событий в программах Qt. Как работает цикл обработки событий?

Задания на уровне воспроизведения

Задание 1. Решить задачу нахождения суммы элементов массива с использованием шаблонной функции.

Задание 2. Реализовать класс контейнерного типа. Структуры реализуются на базе динамического массива. К структуре данных применить шаблон итератор, реализовав его через операции.

Задание 3. Сформировать нестандартный список, обеспечив доступ к середине списка. Организовать операции добавления и удаления, поддерживая неизменность указателя на середину.

Задания на уровне применения полученных знаний

Задание 1. Реализовать класс `Stack` / `Dequeue` / `Queue` / `Vector` / `List` с использованием шаблонов.

Задание 2. Реализовать базовый класс прямоугольник и производные классы: вписанный овал, вписанный ромб.

Задание 3. Написать программу для построения графика функции вида $y=f(x)$.

Примерный перечень вопросов к экзамену

1. Многократная реализация функций в C++. Методы решения этой проблемы. Определение шаблона. Особенности компиляции шаблонов. Возникновение ошибок компиляции.

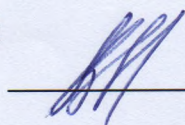
2. Описание и конкретизация шаблона. Требования к параметрам шаблона. Явное и неявное указание параметров при конкретизации шаблонов. Специализация шаблонов.

3. Семантика перемещения. Перемещающий конструктор и оператор присваивания.
4. Явное приведение типов в языке C++. Операторы приведения. Сведения о типах времени выполнения. Класс `type_info`.
5. Оператор приведения `dynamic_cast`: необходимость и использование. Динамическое приведение ссылки.
6. Операторы приведения типов `static_cast`, `const_cast`, `reinterpret_cast`. Краткая характеристика. Особенности использования.
7. Понятие кросс-платформенности. Кросс-платформенные языки программирования. Сценарии и интерпретируемые языки.
8. Подходы к кросс-платформенному программированию. Кросс-платформенные среды исполнения.
9. Структура проекта Qt. Характеристика проектного файла. Компиляция проекта.
10. Последовательные контейнеры Qt.
11. Ассоциативные контейнеры Qt.
12. Контейнер стек. Основные операции. Реализация на массиве.
13. Контейнер стек. Основные операции. Реализация на списковой структуре..
14. Контейнер очередь. Основные операции. Реализация на массиве.
15. Контейнер очередь. Основные операции. Реализация на списковой структуре.
16. Поисковое дерево. Операции добавления и поиска.
17. Бинарная куча. Основное свойство кучи. Реализация кучи на массиве. Операции добавления элемента и извлечения самого приоритетного элемента.
18. Особенности реализации алгоритмов в Qt. Примеры алгоритмов.
19. Работа с текстовыми строками в Qt. Класс `QString`. Примеры операций над строками.
20. Потоки ввода-вывода. Работа с текстовыми файлами.
21. Потоки ввода-вывода. Работа с бинарными файлами.
22. Понятие сериализации объекта. Класс `QDataStream` как основа сериализации.
23. Программирование приложений, управляемых событиями. Цикл обработки сообщений.
24. Философия объектной модели Qt. Класс `QObject`. Понятие свойств класса `QObject`.
25. Механизм сигналов и слотов. Реализация соединения объектов сигнала и слота. Метод `connect()`. Разъединение объектов.
26. Создание графического интерфейса средствами Qt. Виджеты, основные типы виджетов.

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Машинно-ориентированное программирование	Информационных систем управления	Нет	Изменения не требуются (протокол № 18 от 16.05.2023)
Операционные системы	Информационных систем управления	Нет	Изменения не требуются (протокол № 18 от 16.05.2023)
Промышленное программирование	Информационных систем управления	Нет	Изменения не требуются (протокол № 18 от 16.05.2023)

Заведующий кафедрой
д.т.н, профессор



____ А.Н. Курбацкий

16.05.2023

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ ПО ИЗУЧАЕМОЙ УЧЕБНОЙ ДИСЦИПЛИНЕ

на ____ / ____ учебный год

№№ Пп	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры информационных систем управления (протокол № _____)

Заведующий кафедрой
____ профессор д.т.н.____
(ученая степень, звание)

(подпись) ____ А.Н. Курбацкий
(И. О. Фамилия)

УТВЕРЖДАЮ
Декан факультета

(ученая степень, звание)

(подпись) _____
(И. О. Фамилия)