

**ИТОГОВАЯ АТТЕСТАЦИЯ
ПО ПРОГРАММЕ
ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ
«Информационные системы (профессиональное программирование)»**

Итоговая аттестация проводится в форме экзамена и защиты итоговой аттестационной работы.

При прохождении итоговой аттестации в форме экзамена слушатель случайным образом выбирает один из предложенных билетов, в каждом из которых содержится по три вопроса из материала основных модулей 2–7. Список вопросов, включаемых в билеты на итоговой аттестации в форме экзамена, приведен ниже в п. 5.1, пример экзаменационных билетов – в п. 5.2.

Прохождение итоговой аттестации в форме защиты итоговой аттестационной работы предполагает разработку программного проекта и его защиту. Примеры тем итоговых аттестационных работы представлены в п. 5.3.

5.1. Перечень вопросов, выносимых на итоговую аттестацию в форме экзамена

I. Методы программирования на C++

1. Структура программы на C++. Процесс разработки программы. Исходный код и исполняемый файл программы. Препроцессор, компилятор, редактор связей. Среды разработки.

2. Встроенные числовые типы C++ и операции над ними. Представление целых чисел и чисел с плавающей точкой. Способы инициализации. Ввод и вывод чисел.

3. Логический тип и логические операторы. Логические выражения. Условный оператор. Переключатель. Примеры.

4. Циклы, включая цикл по контейнеру. Примеры.

5. Одномерные массивы. Объявление, инициализация, передача в функции. Примеры.

6. Контейнер данных типа vector. Преимущества работы с ним по сравнению с обычными массивами. Основные функции при работе с ним.

7. Многомерные массивы. Объявление, инициализация, передача в функции. Примеры.

8. Функции. Формальные параметры и фактические аргументы. Способы передачи аргументов в функцию. Возвращаемые значения. Прототипы. Примеры.

9. Рекурсивные функции. Локальные статические переменные. Примеры.

10. Перегрузка функций. Аргументы функций по умолчанию. Указатели на функции. Автоматическое определение типа возвращаемого значения. Примеры.

11. Символьный тип данных. Представление символьного типа данных. Терминальные строки. Строковые константы. Стандартные функции обработки строк. Примеры.

12. Тип данных string. Его отличие от терминальных строк. Основные методы для работы с данным типом данных.

13. Структуры. Объявление, инициализация, передача в функции. Примеры.

14. Перечисления, объединения, битовые поля. Примеры.

15. Препроцессор. Директивы препроцессора. Страж повторного включения.

16. Указатели. Указатели и массивы. Операции над указателями в пределах одного массива (адресная арифметика). Примеры.

17. Символьные указатели. Массивы указателей. Примеры.

18. Динамическое выделение и освобождение памяти для переменных, одномерных и двумерных массивов. Преимущества работы с данными, память под которые выделена динамически.

19. Ссылки. Передача и возвращение из функции. Ссылки на временные объекты.

20. Командная строка программы. Обработка параметров командной строки в функции main. Примеры.

21. Программа из нескольких файлов. Проекты. Модули. Заголовочные файлы.

22. Статические переменные модуля. Области действия имен. Пространства имен.

23. Понятие класса и объекта. Инкапсуляция. Управление доступом к членам класса. Статические поля и методы. Конструкторы и деструкторы. Примеры.

24. Перегрузка операторов. Перегрузка операторов в классах. Дружественные функции и операторы. Примеры.

25. Указатель на себя (this). Конструктор копирования и оператор присваивания. Конструктор копирования временных объектов. Примеры.

26. Поточковый ввод и вывод в C++. Перегрузка операторов ввода и вывода для пользовательских типов. Примеры.

27. Наследование классов, множественное наследование. Управление доступом при наследовании. Дружественные классы. Примеры.

28. Виртуальные методы. Абстрактные классы. Раннее и позднее связывание. Примеры.

29. Шаблоны функций и шаблоны классов. Примеры.

30. Обработка исключений. Генерация исключений. Примеры.

31. Текстовые файлы. Потокные объекты C++ для работы с файлами. Открытие и закрытие файлов. Чтение и запись данных. Примеры.
32. Двоичные файлы. Потокные объекты C++ для работы с файлами. Открытие и закрытие файлов. Чтение и запись данных. Управление позиционированием. Примеры.
33. Динамические структуры данных. Основные понятия. Основная идея организации односвязного линейного списка, стека в виде односвязного списка, очереди в виде односвязного списка.
34. Динамические структуры данных. Основные понятия. Основная идея организации односвязного кольцевого списка, двусвязного кольцевого списка.
35. Динамические структуры данных. Бинарные деревья. Идеи прямого, обратного и поперечного обхода.
36. Стандартная библиотека шаблонов STL. Типовые контейнеры. Итераторы и их использование. Примеры.
37. Алгоритмы библиотеки STL. Применение типовых алгоритмов для обработки шаблонных контейнеров STL и массивов. Примеры.
38. Лямбда-выражения и анонимные функции. Примеры.
39. «Умные» указатели C++ на примере `shared_ptr`. Различия «умных» и обычных указателей. Примеры.

II. VBA и офисное программирование

1. Модульная структура программного проекта в MS Word и MS Excel. Типы модулей, назначение модулей, структура размещения кода в модулях. Понятие макроса.
2. Логический строковый и числовые типы в VBA. Особенности описания переменных, констант, статических массивов. Конвертация типов. Виды операторов ветвления. Цикл For, цикл по контейнеру. Примеры.
3. Общий ссылочный тип в VBA, строковый тип, тип для дат и времени, тип Variant. Цикл while и семейство циклов do – Loop. Примеры использования.
4. Динамические массивы в VBA и их использование, определение и изменение границ диапазонов индексов. Передача массивов в процедуры и функции.
5. Определение процедур и функций в VBA. Способы передачи параметров. Правила вызова. Локальные и статические переменные. Примеры.
6. Примеры характерных типов объектов в MS Word, коллекции и общие правила работы с ними. Доступ к документам, элементам документа (абзацам, символам, словам). Примеры изменения элементов документа.
7. Примеры специализированных типов объектов в MS Word. Работа с таблицами, ячейками таблиц. Извлечение и запись данных в ячейки.
8. Типы данных для описания закладок и областей документа MS Word. Коллекция закладок, программное использование закладок. События, связанные с документом. Примеры.
9. Характерные типы в объектной модели MS Excel. Программный доступ к книгам, листам и ячейкам рабочих листов. Примеры событий, связанных с изменением выделения и данных областей листа, добавлением и удалением листов.
10. Тип Range и объект Selection. Примеры полезных методов задания и изменения значений Range. Определение строкового представления адреса, связанного с Range, программная запись формул в ячейки. Программное использование буфера обмена.
11. Основные свойства и методы форм в VBA. Использование форм как модальных и немодальных диалогов. Характерные события для форм. Размещение элементов управления. Понятие родителя и его роль.
12. Принципы визуального проектирования интерфейса форм в VBA. Примеры элементов управления: кнопки, текстовый редактор, список; примеры характерных свойств и событий, связанных с этими элементами. Примеры заполнения списка и использования списка как таблицы.
13. Программное управление приложением MS Word из книги MS Excel. Программное создание нового документа Word на основе шаблона, пример его изменения и сохранения в текущей директории.
14. Взаимодействие с базами данных средствами VBA с использованием внешней библиотеки ADO. Объекты соединения и виртуальной таблицы. Загрузка данных в виртуальную таблицу. Доступ к полям и записям виртуальной таблицы; фильтрация виртуальной таблицы, сохранение изменений. Примеры.
15. Технология изменения ленты, программирование пользовательских элементов управления лентой на примере добавления вкладок, блоков с кнопками. Принцип обеспечения реакции на нажатие пользовательских кнопок.

III. Разработка баз данных

1. Основные понятия баз данных. СУБД и их функции. Проектирование базы данных.
2. Архитектура систем баз данных. Виды архитектур. Преимущества и недостатки.
3. Информационная модель данных «сущность – связь». Ключи и их классификация. Примеры.
4. Модели СУБД (сетевая, иерархическая, многомерная).
5. Реляционная модель данных. Отношения. Реляционная алгебра.
6. Нормализация отношений.
7. Язык SQL. Работа с таблицами: создание, удаление, добавление.

8. Язык SQL. Отбор данных из одной таблицей. Инструкция SELECT. Вычисляемые поля, условие отбора, агрегатные функции, группировка, сортировка.

9. Язык SQL. Отбор данных из нескольких таблиц. Типы объединений JOIN.

IV. Основы разработки GUI-приложений в Qt Creator и на платформе MS .Net в Visual Studio

1. Структура простого Qt Widgets – проекта с формой. Роль ui-файлов, структура C++ файла *_ui.h, порождаемого ui-компилятором. Назначение базовых классов QWidget, QDialog, QMainWindow при создании приложения. Концепция сигналов и слотов. Пример приложения, использующего «вручную» подключаемый слот для обработки сигнала.

2. Разработка пользовательских диалоговых окон в Qt Creator. Создание пользовательского сигнала (с параметрами), посылаемого диалоговым окном, а также пользовательского слота для приема сигнала в главном окне. Соединение. Пример реализации.

3. Контейнерные классы Qt: класс строк QString (перегруженные операторы и полезные методы), список строк QStringList. Контейнеры QByteArray и QVariant. Примеры использования.

4. Контейнерные классы Qt: контейнеры QByteArray и QVariant. Правила регистрации пользовательского типа в классе QVariant. Примеры.

5. Система файлового ввода-вывода в Qt. Использование QFile и класса потоков QTextStream для форматированного текстового ввода-вывода.

6. Система файлового ввода-вывода в Qt. Использование QFile и класса двоичных потоков QDataStream для ввода-вывода в двоичном формате. Использование версия формата данных при двоичном чтении-записи.

7. Использование типовых диалогов в Qt на примере файлового диалога открытия файла. Пример загрузки текстовой информации в список строк QStringList из файла, выбранного с использованием файлового диалога.

8. Структура файла ресурсов для «хранения» изображений в приложениях Qt Widgets. Программная загрузка изображений из файла ресурсов на примере загрузки в QLabel и на пиксельную карту QPixmap. Методы копирования изображений с пиксельной карты на QWidget. Примеры.

9. Разработка главного меню и панелей инструментов для главного окна в приложениях Qt Widgets. Понятие действия, их настройка. Создание интерфейсных элементов со взаимозависимой фиксацией. Пример использования.

10. Организация обработки событий в Qt. События от мыши, клавиатуры, события перерисовки. Переопределение обработчиков событий. Примеры использования событий.

11. Использование «удобных» представлений на примере QTableWidgetItem. Извлечение и программное изменение данных в элементе таблицы. Понятия объектов «элемент ячейки» и «делегат». Пример.

12. Техника замены делегата в виджете QTableWidgetItem для нестандартного отображения данных в элементе представления. Техника встраивания виджетов в элементы таблиц. Примеры.

13. Основы графической системы Qt: кодирование цвета, основные инструменты рисования – перо, кисть. Рисовальщик, примеры графических примитивов. Примеры рисования на QWidget.

14. Графические контейнеры на примере пиксельной карты QPixmap. Методы загрузки изображения из файла и сохранения в файл. Примеры методов преобразования изображений: смещение, сжатие, поворот. Создание маски для заданного фоновой цвета. Копирование изображений из контейнера на виджет с прозрачным фоном.

15. Управляемая куча в .Net. Управляемые указатели и ссылки в C++/CLI. Размерные и ссылочные типы. Размещение объектов в управляемой куче. Характерные изменения в языке C++/CLI по сравнению с C++ на примере использования шаблонных контейнеров array<T>.

16. Принципы работы сборщика мусора в C++/CLI. Деструкторы и финализаторы при использовании в классах динамической памяти управляемого и неуправляемого типа. Простые примеры.

17. Принципы взаимодействия в CLR-сборках на C++/CLI управляемых и неуправляемых частей кода. Техника передачи в неуправляемую функцию C++ данных из управляемого массива в C++/CLI с использованием прикрепляющих указателей pin_ptr<>. Примеры.

18. Разработка на C++/CLI управляемых библиотек классов, их взаимодействие со сборками C#. Установка связей между сборками, определение порядка компиляции. Пример простого приложения с использованием C# Windows Forms.

19. Техника взаимодействия CLR-сборок на C# с неуправляемыми dll-библиотеками на обычном C++. Принципы использования механизма маршallingа на примере обработки числовых данных. Примеры.

5.2. Примеры экзаменационных билетов на итоговой аттестации в форме экзамена

Наследование классов, множественное наследование. Управление доступом при наследовании. Дружественные классы. Примеры.

Язык SQL. Работа с таблицами: создание, удаление, добавление.

Основы графической системы Qt: кодирование цвета, основные инструменты рисования – перо, кисть. Рисовальщик, примеры графических примитивов. Примеры рисования на QWidget.

Руководитель программы _____

Виртуальные методы. Абстрактные классы. Раннее и позднее связывание. Примеры.

Технология изменения ленты, программирование пользовательских элементов управления ленты на примере добавления вкладок, блоков с кнопками. Принцип обеспечения реакции на нажатие пользовательских кнопок.

Разработка пользовательских диалоговых окон в Qt Creator. Создание пользовательского сигнала (с параметрами), посылаемого диалоговым окном, а также пользовательского слота для приема сигнала в главном окне. Соединение. Пример реализации.

Руководитель программы _____

Шаблоны функций и шаблоны классов. Примеры.

Основные свойства и методы форм в VBA. Использование форм как модальных и немодальных диалогов.

Характерные события для форм. Размещение элементов управления. Понятие родителя и его роль.

Управляемая куча в .Net. Управляемые указатели и ссылки в C++/CLI. Размерные и ссылочные типы.

Размещение объектов в управляемой куче. Характерные изменения в языке C++/CLI по сравнению с C++ на примере использования шаблонных контейнеров `ataguy<T>`.

Руководитель программы _____

5.3. Примеры тем итоговых аттестационных работ (программных проектов)

1. Реализация на языке программирования C++ обмена данными с внешним хранилищем по протоколу FTP (язык программирования C++, среда разработки MS Visual Studio).

2. Qt-приложение для проведения численных расчетов (язык программирования C++, среда разработки Qt Creator).

3. Разработка в Qt программы для автоматизированного расчета смет (язык программирования C++, среда разработки Qt Creator).

4. Электронный органайзер (средства разработки: html, CSS, Java Script, библиотека JQuery, фреймворк Electron.js).

5. Репетитор английских слов (язык программирования C++, база данных SQLite, среда разработки Qt Creator).

6. Загрузка, обработка и анализ открытых сведений ИФНС с помощью разработанного АРМ «Open Data» версии 0.0.07 (язык программирования C++, база данных SQLite, среда разработки Qt Creator).
7. Разработка информационного веб-ресурса о снукере (выполнена с использованием PHP).
8. Разработка программного комплекса для игры «Пасьянс из домино» (язык программирования C++, среда разработки MS Visual Studio).
9. Программа на Qt: «Панель управление системой автономного электроснабжения» (язык программирования C++, среда разработки Qt Creator).
10. Разработка программного комплекса для графического моделирования физических процессов основных задач школьной физики (язык программирования C++, среда разработки C++Builder).
11. Использование среды Qt Creator при программировании игр (язык программирования C++, среда разработки Qt Creator).
12. Разработка приложения в среде C# с использованием СУБД MS Access для обучения в игровой форме (язык программирования C#, среда разработки MS Visual Studio, с использованием СУБД MS Access).
13. Разработка программа расчёта диаграммы направленности фазированной антенной решётки (язык программирования C++, среда разработки Qt Creator, с использованием стороннего виджета QCustomPlot).
14. Разработка базы данных для транспортной компании ООО «Лотос» (язык программирования VBA, использование базы данных MS Access).
15. Программа для решения прямой задачи магниторазведки для двухмерного тела произвольной формы (язык программирования C++, среда разработки Qt Creator).

Формы и методы контроля, а также способы оценивания промежуточных результатов освоения модулей представлены в Рабочих программах самих модулей (приложения №3.1-3.9).

Формы и методы контроля и оценки результатов итоговой аттестации представлены в табл. 1. Время прохождения каждой из форм итоговой аттестации слушателем – 2 ак. часа.

Таблица 1. Формы и методы контроля и оценки результатов итоговой аттестации

№ п/п	Наименование процедуры	Основные показатели оценки	Формы и методы контроля и оценки
1.	Итоговая аттестация – сдача итогового экзамена	1. Полнота и правильность ответа на три основных вопроса билета. 2. Степень полноты и правильности ответов на дополнительные вопросы по билету. 3. Оценка кругозора – правильность ответа на вопросы, выходящие за рамки билета.	Оценка каждого члена комиссии носит экспертный характер, итоговая оценка – результат коллективного обсуждения. Оценка ставится по пятибалльной шкале. Формализованные шкалы оценивания не применяются ввиду разнообразия и сложности материала.
2.	Итоговая аттестация – защита итоговой аттестационной работы (программного проекта)	1. Техническая сложность разработки, степень завершенности. 2. Оригинальность темы проекта и способов его реализации. 3. Уровень самостоятельности при разработке, глубина понимания представленного кода. 4. Грамотность оформления и представления проекта комиссии.	Оценка каждого члена комиссии носит экспертный характер, итоговая оценка – результат коллективного обсуждения. Оценка ставится по пятибалльной шкале. Формализованные шкалы оценивания не применяются ввиду разнообразия тематики программных проектов и применяемых средств разработки.

Критерии оценки итоговой аттестации в форме экзамена

Отлично	дает точное определение основных понятий, демонстрирует самостоятельность и аргументированность суждений, умеет установить связь между содержательными элементами курса
---------	---

Хорошо	удовлетворяет названным выше требованиям, но допускает одну негрубую ошибку или не более двух недочетов и может их исправить самостоятельно или при помощи небольшой помощи преподавателя
Удовлетворительно	при ответе обнаруживает отдельные пробелы в усвоении существенных вопросов курса, испытывает затруднения в применении знаний при объяснении или в подтверждении конкретных примеров практического применения теории; отвечает неполно на вопросы преподавателя; недостаточно понимает отдельные важные положения курса, допуская одну - две грубые ошибки
Не удовлетворительно	Не знает и не понимает значительную или основную часть материала в пределах поставленных вопросов; имеет слабо сформированные и неполные знания; при ответе (на один вопрос) допускает более двух грубых ошибок, которые не может исправить даже при помощи преподавателя.

Критерии оценки итоговой аттестационной работы

"5" (отлично) – работа отвечает структуре, имеет последовательное изложение материала с соответствующими выводами, результаты соответствуют поставленным задачам, инструменты применены правильно. При защите работы слушатель показывает глубокие знания практических инструментов и технические особенности их применения, грамотно отвечает на поставленные вопросы.

"4" (хорошо) – работа соответствует структуре, однако имеет некоторые не вполне точно выполненные задания. При защите слушатель показывает знания инструментов, оперирует характеристиками инструментария, без особых затруднений отвечает на поставленные вопросы.

"3" (удовлетворительно) – работа отвечает структуре, базируется на практическом материале, но имеет поверхностный характер выполнения поставленных задач. При защите слушатель проявляет неуверенность, показывает недостаточное знание практических инструментов, не всегда дает аргументированные ответы на заданные вопросы.

"2" (неудовлетворительно) – выставляется за работу, которая не отвечает требованиям заданной структуры. При защите слушатель не отвечает на поставленные вопросы.