

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н.И. ЛОБАЧЕВСКОГО»**

УТВЕРЖДЕНО
Решением
Ученого совета ННГУ
протокол №14
от «28» декабря 2022г.

**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ**

Информационные системы (профессиональное программирование)
565 часов

Нижний Новгород

1. ОБЛАСТЬ ПРИМЕНЕНИЯ

1.1. Цель программы

Совершенствование и формирование новых компетенций, необходимых для профессиональной деятельности в области алгоритмизации, программирования и проектирования программных систем, включая системы обработки данных (преимущественно на основе использования языка C++) для слушателей.

1.2. Нормативные документы для разработки программы профессиональной переподготовки:

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- Приказ Министерства образования и науки РФ от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;

- Приказ Министерства образования и науки РФ от 23 августа 2017 г. № 808 "Об утверждении федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии" (с изменениями и дополнениями), редакция с изменениями № 1456 от 26.11.2020;

- Приказ Министерства образования и науки РФ от 19 сентября 2017 г. № 926 "Об утверждении федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению подготовки 09.03.02 Информационные системы и технологии" (с изменениями и дополнениями), редакция с изменениями № 1456 от 26.11.2020

- Приказ Министерства образования и науки РФ от 19 сентября 2017 г. N 920 "Об утверждении федерального государственного образовательного стандарта высшего образования - бакалавриат по направлению подготовки 09.03.04 Программная инженерия" (с изменениями и дополнениями), редакция с изменениями N 1456 от 26.11.2020;

- Профессиональный стандарт 06.001 «Программист», утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 18 ноября 2013 г. № 679н (зарегистрирован Министерством юстиции Российской Федерации 18 декабря 2013 г., регистрационный № 30635), с изменениями, внесенным приказом Министерства труда и социальной защиты Российской Федерации от 12 декабря 2016 г. №727н (зарегистрирован Министерством юстиции Российской Федерации 13 января 2017 г., регистрационный № 45230);

- Профессиональный стандарт 06.004 "Специалист по тестированию в области информационных технологий", утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 11 апреля 2014 г. N 225н (зарегистрирован Министерством юстиции Российской Федерации 9 июня 2014 г., регистрационный N 32623), с изменением, внесенным приказом Министерства труда и социальной защиты Российской Федерации от 12 декабря 2016 г. N 727н (зарегистрирован Министерством юстиции Российской Федерации 13 января 2017 г., регистрационный N 45230);

- Профессиональный стандарт 06.011 "Администратор баз данных", утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 17 сентября 2014 г. N 647н (зарегистрирован Министерством юстиции Российской Федерации 24 ноября 2014 г., регистрационный N 34846), с изменением, внесенным приказом Министерства труда и социальной защиты Российской Федерации от 12 декабря 2016 г. N 727н (зарегистрирован Министерством юстиции Российской Федерации 13 января 2017 г., регистрационный N 45230);

- Образовательный стандарт ВО бакалавриата ННГУ по направлению подготовки 02.03.02 «Фундаментальная информатика и информационные технологии», утвержденный решением Ученого совета ННГУ (протокол от 16.06.2021 № 8), введенный приказом ННГУ от 21.06.2021 №349-ОД;

- Образовательный стандарт ВО бакалавриата ННГУ по направлению подготовки 09.03.02 «Информационные системы и технологии», утвержденный решением Ученого совета ННГУ (протокол от 16.06.2021 № 8), введенный приказом ННГУ от 21.06.2021 №349-ОД;

1.3. Категории слушателей на обучение которых рассчитана программа профессиональной переподготовки (далее – Программа):

Студенты старших курсов бакалавриата и специалитета, магистранты, аспиранты, работники научно-исследовательской сферы, государственные и муниципальные служащие,

административный и иной среднетехнический исполнительный персонал, работники сферы IT, индивидуальные предприниматели, сотрудники коммерческих компаний.

Программа предназначена как для слушателей, имеющих опыт программирования, так и для начинающих с нуля программистов без опыта программирования. Желательным (но не обязательным) является естественно-научный, инженерно-технический или же финансово-экономический профиль имеющегося образования.

1.4. Входные требования к обучающимся (в случае необходимости):

Как отмечено в п.1.3, по Программе могут обучаться слушатели, имеющие нулевой уровень подготовки в области программирования. Однако, в начале обучения слушатели должны, по крайней мере, обладать компетенциями на уровне квалифицированного пользователя персонального компьютера, быть способными самостоятельно устанавливать программное обеспечение, необходимое для обучения по Программе.

1.5. Сфера применения слушателями полученных профессиональных компетенций, умений и знаний.

Сфера научных исследований и академического программирования (область образования и науки) – из ФГОС ВО 3++ направления бакалавриата «Фундаментальная информатика и информационные технологии (ФИИТ)».

Сфера разработки и тестирования программного обеспечения; создания, поддержки и администрирования информационно-коммуникационных систем и баз данных, управления информационными ресурсами в информационно-телекоммуникационной сети "Интернет" (область связи, информационные и коммуникационные технологии) – из ФГОС ВО 3++ направления бакалавриата «Фундаментальная информатика и информационные технологии».

Сфера организации и проведения научно-исследовательских и опытно-конструкторских работ в области информатики и вычислительной техники (область сквозных видов профессиональной деятельности в промышленности) – из профессиональных стандартов «Информационные системы и технологии», «Программная инженерия»

Сфера разработки компьютерного программного обеспечения (разработчики программного обеспечения, программисты приложений) – из профессионального стандарта «Программист».

Сфера деятельности, связанная с использованием вычислительной техники и информационных технологий, прочая (разработчики и аналитики программного обеспечения и приложений, не входящие в другие группы) – из профессионального стандарта «Специалист по тестированию в области информационных технологий»

Выпускники могут осуществлять профессиональную деятельность в других областях профессиональной деятельности и (или) сферах профессиональной деятельности при условии соответствия уровня их образования и полученных компетенций требованиям к квалификации работника.

2. ХАРАКТЕРИСТИКА ПОДГОТОВКИ ПО ПРОГРАММЕ

2.1. Нормативный срок освоения программы 565 ак. часов.

2.2. Срок обучения – 39 недель (включая рождественские каникулы).

2.3. Общая трудоемкость – 15,69 ЗЕ¹.

2.4. Режим обучения: первые 13 недель (первый семестр) – в среднем по 12.0 ак. часов в неделю; 1 неделя – рождественские каникулы; последние 25 недель (второй семестр) – в среднем по 13.9 ак. часа в неделю.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Учебная программа включает 7 курсов (модулей), промежуточные и итоговые аттестации, консультации перед экзаменом итоговой аттестации (ИА) и консультации перед защитой итоговой аттестационной работы (ИАР) (итогового программного проекта), часы самостоятельной работы для подготовки к промежуточным и итоговым аттестациям. Объем программы — составляет 565 академических часов.

В обязательную часть программы входят: алгоритмизация и программирование на базовой части C/C++, объектно-ориентированное программирование на C++, Web-технологии, программирование офисных приложений на Visual Basic for Application, проектирование и разработка баз

¹ 1 ЗЕ = 36 ак. часов

данных, визуальная разработка GUI-приложений на базе C++ Qt Widget, VBA, а также C#, C++/CLI платформы .Net. Все курсы предусматривают теоретическую часть, проведение практических занятий в компьютерных классах, выполнение домашних заданий. Выполняется текущий контроль усвоения материала модулей – промежуточная аттестация (в форме зачетов), а также итоговая аттестация в форме сдачи экзамена и подготовки под руководством преподавателей и последующей защиты итоговой аттестационной работы (программного проекта).

Ниже, в п.3.1 приведены описания профессиональных компетенций в рамках имеющейся квалификации, приобретение и (или) качественное изменение которых осуществляется в результате обучения по Программе.

Слушатель, освоивший программу, должен:

3.1. обладать профессиональными компетенциями, включающими в себя способность:

ПК-1: Способен к применению применять в профессиональной деятельности современные языки программирования и методы обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии (научно-исследовательский тип профессиональной деятельности) — ОС ННГУ по направлению ФИИТ;

ПК-2. Способен разрабатывать программное обеспечение систем цифровой обработки данных в различных областях профессиональной деятельности (научно-исследовательский тип проф. деятельности) — ОС ННГУ по направлению «Информационные системы и технологии»;

ПК-3. Способен проектировать программное обеспечение (производственно-технологический тип профессиональной деятельности) — ОС ННГУ по направлению ФИИТ;

ПК-4. Способен использовать современные инструментальные и вычислительные средства информационных технологий (производственно-технологический тип профессиональной деятельности) — ОС ННГУ по направлению ФИИТ;

ПК- 5. Способен проводить проектирование программного обеспечения информационных систем в рамках проектов малой и средней сложности (проектный тип деятельности) — ОС ННГУ по направлению «Информационные системы и технологии»

3.2. знать:

1. Языки, утилиты и среды программирования (C++, VBA, C++/CLI, частично C#; IDE Visual Studio, редактор VBA, Qt Creator)

2. Специальные знания по работе с установленной БД (на примере СУБД MS Access);

3. Методы и средства сборки модулей и компонентов компьютерного программного обеспечения;

4. Методы и средства проверки работоспособности выпусков программных продуктов;

5. Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке компьютерного программного обеспечения;

6. Методы и средства проектирования компьютерного программного обеспечения;

7. Методы и средства проектирования баз данных;

8. Методы и средства проектирования программных интерфейсов и графических интерфейсов;

9. Методологии разработки компьютерного программного обеспечения и технологии программирования (включая принципы разработки на платформе .NET, принципы организации взаимодействия управляемого кода на C# с неуправляемым кодом на C++, использования управляемой модификации C++/CLI).

3.3. уметь:

1. Применять языки программирования для написания программного кода;

2. Использовать выбранную среду программирования для разработки процедур интеграции программных модулей;

3. Применять методы и средства сборки модулей и компонентов компьютерного программного обеспечения, разработки процедур для развертывания компьютерного программного обеспечения, миграции и преобразования данных, создания программных интерфейсов;

4. Выполнять процедуры сборки программных модулей и компонентов в программный продукт;

5. Проводить проверку работоспособности программного продукта;

6. Создавать резервные копии программ и данных, выполнять восстановление, обеспечивать целостность программного продукта и данных;
7. Работать с системами хранения и обработки информации;
8. Отличать штатное состояние базы данных (БД) от работы БД в нештатном режиме;
9. Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов;
10. Вырабатывать варианты реализации требований к компьютерному программному обеспечению.

3.4. владеть:

1. Навыками процедурного, модульного, объектно-ориентированного программирования (преимущественно на основе языка C++);
2. Навыками разработки, отладки, тестирования приложений с использованием IDE MS Visual Studio, Qt Creator, а также редактора VBA;
3. Навыками проектирования графических интерфейсов пользователя с использованием средств визуального программирования по технологии Qt Widgets, а также (частично) на основе форм VBA и технологии C# MS Forms .NET Framework;
4. Навыками проектирования баз данных невысокой сложности, использования языка запросов SQL.

3.5. Характеристика нового вида профессиональной деятельности, новой квалификации:

- к областям профессиональной деятельности относятся: — образование и наука; — связь, информационные и коммуникационные технологии (в указанных в п. 1.5 сферах применения); — сквозные виды профессиональной деятельности в промышленности (см. п. 1.5);
- объекты профессиональной деятельности: — алгоритмические модели, программы, программные системы и комплексы, методы их проектирования и реализации, способы производства, сопровождения, эксплуатации и администрирования;
- виды профессиональной деятельности включают: — научно-исследовательский; — производственно-технологический, — проектный; задачи профессиональной деятельности включают: — разработку и отладку программного кода, — проверку работоспособности и рефакторинг кода программного обеспечения, — интеграцию программных модулей и компонентов и проверку работоспособности выпусков программного продукта, — разработку требований и проектирование программного обеспечения;
- результаты обучения, в соответствии с профессиональным стандартом «Программист», позволяют выполнять трудовые функции на различных уровнях от 3 до 6 включительно.

4. ТРЕБОВАНИЯ К СТРУКТУРЕ ПРОГРАММЫ

Программа предусматривает изучение следующих основных модулей:

- Web – технологии;
- Основы алгоритмизации и программирования на основе языка C/C++;
- Объектно-ориентированное программирование на базе C++;
- Программирование офисных приложений на языке VBA;
- Проектирование и разработка баз данных;
- Разработка GUI- приложений на C++ в Qt Creator;
- Разработка GUI-приложений на C++/CLI, C# в MS Visual Studio на платформе MS .Net Framework.

Учебный план программы профессиональной переподготовки представлен в Приложении №1 к программе профессиональной переподготовки.

Календарный учебный график программы профессиональной переподготовки — в Приложении №2 к программе профессиональной переподготовки.

5. ТРЕБОВАНИЯ К ОЦЕНКЕ КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ 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. Среда 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. Поточный 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. Поточные объекты C++ для работы с файлами. Открытие и закрытие файлов. Чтение и запись данных. Примеры.

32. Двоичные файлы. Поточные 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 ленты 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. Инфологическая 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 таблицы. 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 фиксации. 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 элементах 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 файле. 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.	Экзамен/устный опрос 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.	Защита итоговой аттестационной работы/защита проекта 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.

6. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ ОБЕСПЕЧЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Руководитель программы профессиональной переподготовки:

Городецкий Станислав Юрьевич, к.ф.-м.н., звание – доцент, доцент кафедры теории управления и динамики систем Института информационных технологий, математики и механики ННГУ.

Разработчики программы профессиональной переподготовки:

1. Городецкий Станислав Юрьевич, к.ф.-м.н., звание – доцент, доцент кафедры теории управления и динамики систем (ТУиДС) Института информационных технологий, математики и механики ННГУ;

2. Капитанов Денис Владимирович, к.ф.-м.н., доцент кафедры теоретической, компьютерной и экспериментальной механики (ТКЭМ) Института информационных технологий, математики и механики ННГУ.

Составители учебно-тематического плана программы профессиональной переподготовки:

1. Городецкий Станислав Юрьевич, к.ф.-м.н., звание – доцент, доцент кафедры теории управления и динамики систем Института информационных технологий, математики и механики ННГУ;

2. Капитанов Денис Владимирович, к.ф.-м.н., доцент кафедры теоретической, компьютерной и экспериментальной механики Института информационных технологий, математики и механики ННГУ.

3. Бурова Маргарита Сергеевна, преподаватель кафедры информационных технологий и инструментальных методов в экономике Института экономики и предпринимательства ННГУ.

Сведения о педагогических (научно-педагогических) работниках, участвующих в реализации программы профессиональной переподготовки, и лицах, привлекаемых к реализации дополнительной образовательной программы на иных условиях, представлены в табл. 2.

Таблица 2.

Преподаватели программы профессиональной переподготовки
«Информационные системы (профессиональное программирование)»

№ п/п	Наименование модулей (тем, разделов)	Фамилия, имя, отчество,	Ученая степень, ученое звание	Основное место работы, должность	Место работы и должность по совместительству (если есть)
1.	Web - технологии	Бурова Маргарита Сергеевна	нет	Кафедра информационных технологий и инструментальных методов в экономике ИЭП ННГУ, преподаватель	Центр непрерывного повышения профессионального мастерства педагогических работников ФПКИПП ННГУ, преподаватель дополнительного образования
2.	Основы алгоритмизации и программирования на основе языка C/C++	Капитанов Денис Владимирович	к.ф.-м.н.	кафедра ТКЭМ ИИТММ ННГУ, доцент	нет
		Городецкий Станислав Юрьевич	к.ф.-м.н., доцент	кафедра ТУиДС ИИТММ ННГУ, доцент	нет
3.	Объектно-ориентированное программирование на базе C++	Капитанов Денис Владимирович	к.ф.-м.н.	кафедра ТКЭМ ИИТММ ННГУ, доцент	нет
		Городецкий Станислав Юрьевич	к.ф.-м.н., доцент	кафедра ТУиДС ИИТММ ННГУ, доцент	нет
4	Программирование офисных приложений на языке VBA	Городецкий Станислав Юрьевич	к.ф.-м.н., доцент	кафедра ТУиДС ИИТММ ННГУ, доцент	нет
5	Проектирование и разработка баз данных	Капитанов Денис Владимирович	к.ф.-м.н.	кафедра ТКЭМ ИИТММ ННГУ, доцент	нет
6	Разработка GUI-приложений на C++ в Qt Creator	Городецкий Станислав Юрьевич	к.ф.-м.н., доцент	кафедра ТУиДС ИИТММ ННГУ, доцент	нет
7	Разработка GUI-приложений на C++/CLI, C# в MS Visual Studio на платформе MS .Net Framework	Городецкий Станислав Юрьевич	к.ф.-м.н., доцент	кафедра ТУиДС ИИТММ ННГУ, доцент	нет

Учебно-методическое и информационное обеспечение программы, а также материально-технические условия реализации модулей программы представлены в приложениях №3.1–№3.7 к программе профессиональной переподготовки, содержащих рабочие программы модулей, указанных в табл. 2.