1. **Паспорт Образовательной программы**

Уважаемые образовательные организации!

Вы можете преобразовать шаблон в обычный Word, удалив установленное нами закрепление полей, но сохранив общую структуру. Закрепление было сделано для удобства ОО, чтобы было понятно, где можно менять данные. А где – нет.

При внесении необходимых Вам изменений в данный шаблон, пожалуйста, сохраняйте предложенную нами структуру документа.

**«** Язык программирования С++ **»**

|  |  |
| --- | --- |
| **Версия программы** | **1** |
| **Дата Версии** | 12**.**10**.**2020 |

1. **Сведения о Провайдере**

|  |  |  |
| --- | --- | --- |
| 1.1 | Провайдер | Нижегородский государственный технический университет им. Р.Е. Алексеева |
| 1.2 | Логотип образовательной организации |  |
| 1.3 | Провайдер ИНН | 5260001439 |
| 1.4 | Ответственный за программу ФИО | Ляхманов Дмитрий Александрович |
| 1.5 | Ответственный должность | И.о. заведующего кафедрой "Информационная безопасность вычислительных систем и сетей" |
| 1.6 | Ответственный Телефон | +7-910-105-40-92 |
| 1.7 | Ответственный Е-mail | <lyahmanov@nntu.ru> |

1. **Основные Данные**

|  |  |  |
| --- | --- | --- |
| **№** | **Название** | **Описание** |
| 2.1 | Название программы | Язык программирования С++ |
| 2.2 | Ссылка на страницу программы | ссылка |
| 2.3 | Формат обучения | Онлайн |
| 2.4 | Подтверждение от ОО наличия возможности реализации образовательной программы с применением электронного обучения и (или) дистанционных образовательных технологий с возможностью передачи данных в форме элементов цифрового следа | Существует возможность реализации образовательной программы с применением электронного обучения |
| 2.5 | Уровень сложности | Базовый |
| 2.6 | Количество академических часов | **72** |
| 2.7 | Практикоориентированный характер образовательной программы: не менее 50 % трудоёмкости учебной деятельности отведено практическим занятиям и (или) выполнению практических заданий в режиме самостоятельной работы (кол-во академических часов) | Практикоориентированный характер образовательной программы: количество учебной деятельности отведено практическим занятиям и (или) выполнению практических заданий в режиме самостоятельной работы (44 академических часа) |
| 2.8 | Стоимость обучения одного обучающегося по образовательной программе, а также предоставление ссылок на 3 (три) аналогичные образовательные программы иных организаций, осуществляющих обучение, для оценки объективности стоимости или обоснование уникальности представленной образовательной программы в случае отсутствия аналогичных образовательных программ на рынке образовательных услуг | 25 000  <http://edu.bmstu.ru/napravleniya-obucheniya/programmirovanie/cpp> 48 часов - 25 290  [http://itcenter.itmo.ru/programmirovanie\_na\_micr - 30 000](http://itcenter.itmo.ru/programmirovanie_na_micr%20-%2030%20000)  <https://www.avalon.ru/Courses/Course/DEV-C-MP2-Modulnaya-programma-Obektno-orientirovannoe-programmirovanie-YAzyk-S/2294/#nav-price> - 32 000 |
| 2.9 | Минимальное количество человек на курсе | 25 |
| 2.10 | Максимальное количество человек на курсе | 200 |
| 2.11 | Данные о количестве слушателей, ранее успешно прошедших обучение по образовательной программе | 150 |
| 2.12 | Формы аттестации | Зачет |
| 2.13 | Указание на область реализации компетенций цифровой экономики, к которой в большей степени относится образовательная программа, в соответствии с Перечнем областей | Программирование и создание ИТ-продуктов |

1. **Аннотация программы**

Наиболее полное и содержательное описание программы, которое включает:

1) общую характеристику компетенций, качественное изменение которых осуществляется в результате обучения или которые формируются в результате освоения образовательной программы;

2) описание требований и рекомендаций для обучения по образовательной программе;

3) краткое описание результатов обучения в свободной форме, а также описание востребованности результатов обучения в профессиональной деятельности.

Ограничение по размеру: не менее 1000 символов -?

Язык программирования C++ — компилируемый строго типизированный язык программирования общего назначения. Поддерживает разные парадигмы программирования: процедурную, обобщённую, функциональную; наибольшее внимание уделено поддержке объектно-ориентированного программирования.

Целью преподавания дополнительной профессиональной программы «Язык программирования С++» является изучение

основных принципов построения программного обеспечения различного назначения на языке программирования С++. В рамках лабораторных занятий изучаются особенности используемых программных платформ, рассматриваются приемы построения программ серверного и пользовательского классов, закладываются основы микроконтроллерного программирования. Лабораторные занятия сопровождаются практикумом на ЭВМ, где обучающиеся самостоятельно выполняют поставленные перед ними задачи на разработку программных модулей на языке С++, что является важной составляющей эффективной реализации программного продукта.

Задачи дисциплины:

- обучить принципам построения

программных продуктов и основным используемым технологиям;

- привить обучающимся навыки программирования на языке С++, используемые при создании серверного, пользовательского и микроконтроллерного ПО;

- дать опыт разработки пользовательских и серверных программ с использованием современных технологий и методов.

1. ШАБЛОН ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ (ДПО)

Титульный лист программы

Название организации

Дополнительная профессиональная программа повышения квалификации

« Язык программирования С++ »

72 час.

**ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ**

**1.Цель программы**

Эффективное использование языков и методов программирования для обработки информации при решении практических задач

**2.Планируемые результаты обучения:**

2.1.Знание (осведомленность в областях)

2.1.1. Этапы решения задач на ЭВМ, основные конструкции языков высокого уровня, структурные типы данных, основы алгоритмизации

2.1.2. Базовые особенности реализации принципов структурного и объектно-ориентированного программирования

2.1.3. Базовые методы и технологии разработки программного обеспечения

2.2. Умение (способность к деятельности)

2.2.1. Оптимизировать алгоритмы для решения прикладных задач,

2.2.2. Выбирать стандартные структуры данных для реализации программного обеспечения.

2.2.3. Разрабатывать специальные структуры данных и алгоритмы их обработки.

2.3.Навыки (использование конкретных инструментов)

2.3.1. Практические навыки реализации программного обеспечения средствами интегрированных сред разработки

2.3.2. Практические навыки владения инструментами по отладке программного обеспечения

2.3.3. Практические навыки владения методами тестирования программного обеспечения

**3.Категория слушателей** (возможно заполнение не всех полей)

* 1. Образование - Высшее
  2. Квалификация - бакалавр, магистр, специалист
  3. Наличие опыта профессиональной деятельности - базовые знания в области информатики и знакомство с принципами языков программирования
  4. Предварительное освоение иных дисциплин/курсов /модулей - Не требуется

**4.Учебный план программы «**Язык программирования С++**»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Модуль** | **Всего, час** | **Виды учебных занятий** | | |
| **лекции** | **практические занятия** | **самостоятельная работа** |
| 1 | Введение в C++ | 10 | 4 | 4 | 2  (в виде практических заданий) |
| 2 | Объектно-ориентированное программирование | 32 | 12 | 12 | 8  (в виде практических заданий) |
| 3 | Динамические структуры данных | 10 | 4 | 4 | 2  (в виде практических заданий) |
| 4 | Рекурсивные алгоритмы | 10 | 4 | 4 | 2  (в виде практических заданий) |
| 5 | Методы тестирования и отладки ПО | 10 | 4 | 4 | 2  (в виде практических заданий) |
| **Итоговая аттестация** | |  | **Указывается вид (экзамен, зачёт, реферат и т.д.)** | | |
|  | | 72 | Зачет | | |

**5.Календарный план-график реализации образовательной** программы

(дата начала обучения – дата завершения обучения) в текущем календарном году, указания на периодичность набора групп (не менее 1 группы в месяц)

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Наименование учебных модулей** | **Трудоёмкость (час)** | **Сроки обучения** |
| **1** | Введение в C++ | 10 | В период 01.11.2020-15.11.2020 |
| **2** | Объектно-ориентированное программирование | 32 | В период 01.11.2020-15.11.2020 |
| 3 | Динамические структуры данных | 10 | В период 01.11.2020-15.11.2020 |
| 4 | Рекурсивные алгоритмы | 10 | В период 01.11.2020-15.11.2020 |
| 5 | Методы тестирования и отладки ПО | 10 | В период 01.11.2020-15.11.2020 |
| **Всего:** | | 72 | 01.11.2020-15.11.2020 |

**6.Учебно-тематический план программы «**  Язык программирования С++ **»**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Модуль / Тема** | **Всего, час** | **Виды учебных занятий** | | | **Формы контроля** |
| **лекции** | **практические занятия** | **самостоятельная работа** |
| 1 | Введение в C++ | 10 | 4 | 4 | 2  (в виде практических заданий) | Практическое задание |
| 1.1 | Операторы языка С++ | 2 | 1 | 1 |  |  |
| 1.2 | Операторы управления | 2 | 1 | 1 |  |  |
| 1.3 | Массивы и указатели | 3 | 1 | 1 | 1 |  |
| 1.4 | Работа с файлами | 3 | 1 | 1 | 1 |  |
| 2 | Объектно-ориентированное программирование | 32 | 12 | 12 | 8  (в виде практических заданий) | Практическое задание |
| 2.1 | Введение в ООП | 5 | 2 | 2 | 1 |  |
| 2.2 | Наследование | 5 | 2 | 2 | 1 |  |
| 2.3 | Полиморфизм и виртуальные функции | 6 | 2 | 2 | 2 |  |
| 2.4 | Обработка исключительных ситуаций | 6 | 2 | 2 | 2 |  |
| 2.5 | Шаблоны и библиотека STL | 10 | 4 | 4 | 2 |  |
| 3 | Динамические структуры данных | 10 | 4 | 4 | 2  (в виде практических заданий) | Практическое задание |
| 3.1 | Однонаправленные списки, двунаправленные списки, круговые списки | 5 | 2 | 2 | 1 |  |
| 3.2 | Стеки, очереди, деки | 5 | 2 | 2 | 1 |  |
| 4 | Рекурсивные алгоритмы | 10 | 4 | 4 | 2  (в виде практических заданий) | Практическое задание |
| 4.1 | Методы решения задач с помощью рекурсивных алгоритмов | 5 | 2 | 2 | 1 |  |
| 4.2 | Обработка древовидных структур данных с помощью рекурсивных алгоритмов | 5 | 2 | 2 | 1 |  |
| 5 | Методы тестирования и отладки ПО | 10 | 4 | 4 | 2  (в виде практических заданий) | Практическое задание |
| 5.1 | Методы автоматизированного тестирования ПО | 5 | 2 | 2 | 1 |  |
| 5.2 | Построение отладочной инфраструктуры | 5 | 2 | 2 | 1 |  |

**7. Учебная (рабочая) программа повышения квалификации «** Язык программирования С++ **»**

Модуль 1. Введение в C++ ( \_10\_\_ час.)

Тема 1.1 Операторы языка С++ ( \_2\_\_ час.)

Основные типы данных. Операции над основными типами данных.

Тема 1.2 Операторы управления ( \_2\_\_ час.)

Операторы if-else, for, switch, while

Тема 1.3 Массивы и указатели ( \_3\_\_ час.)

Ссылки и указатели, передача указателей в функцию, функции управления памятью

Тема 1.4 Работа с файлами ( \_3\_\_ час.)

Типы файлов и методы доступа к элементам. Операции над файлами, стандартные процедуры и функции обработки файлов. Работа с типизированными, текстовыми и безтиповыми файлами

Модуль 2. Объектно-ориентированное программирование ( \_32\_\_ час.)

Тема 2.1 Введение в ООП ( \_5\_\_ час.)

Классы. Описание класса. Поля класса. Методы класса. Хранение полей и методов класса. Доступ к элементам класса (private, public, protected). Конструкторы. Деструкторы.

Тема 2.2 Наследование ( \_\_5\_ час.)

Композиция . Базовые классы и производные классы Управление доступом к элементам базовых классов. Переопределение элементов базового класса в производном классе. Использование конструкторов и деструкторов в производных классах.

Тема 2.3 Полиморфизм и виртуальные функции ( \_6\_\_ час.)

Виртуальные функции и полиморфизм. Виртуальные функции. Позднее связывание. Механизм вызова виртуальных функций. Абстрактные классы и конкретные классы. Виртуальные функции и деструкторы. Отсутствие переопределения.

Тема 2.4 Обработка исключительных ситуаций ( \_6\_\_ час.)

Синтаксис исключений. Основная идея механизма обработки исключений. Генерация исключений. Перехват исключений блоком catch. Использование обработчика catch(…). Исключения, связанные с ошибкой выделения памяти. Конструкторы, деструкторы и исключения. Спецификация исключений. Алгоритм обработки исключительной ситуации.

Тема 2.5 Шаблоны и библиотека STL ( \_10\_\_ час.)

Определение шаблонов классов и функций. Наследование шаблонов и дружественность. Виртуализация шаблонов.

Модуль 3. Динамические структуры данных ( \_10\_\_ час.)

Тема 3.1 Однонаправленные списки, двунаправленные списки, круговые списки ( \_5\_\_ час.)

Операции вставки, удаления для однонаправленные, двунаправленных и круговых списков

Тема 3.2 Стеки, очереди, деки ( \_5\_\_ час.)

Операции вставки, удаления для стеков, очередей, деков

Модуль 4. Рекурсивные алгоритмы( \_\_10\_ час.)

Тема 4.1 Методы решения задач с помощью рекурсивных алгоритмов ( \_5\_\_ час.)

Виды рекурсий. Память и время при реализации рекурсивных алгоритмов, понятие стека памяти.

Тема 4.2 Обработка древовидных структур данных с помощью рекурсивных алгоритмов ( \_5\_\_ час.)

Модуль 5. Методы тестирования и отладки ПО ( \_10\_\_ час.)

Тема 5.1 Методы автоматизированного тестирования ПО ( \_5\_\_ час.)

Типы тестирования ПО. Тестирование чёрного и белого ящиков. Классификация ошибок. Автоматизированное тестирование.

Тема 5.2 Построение отладочной инфраструктуры ( \_5\_\_ час.)

Классификация ошибок. Системы логирования фактов. Средства сбора и анализа данных.

**Описание практико-ориентированных заданий и кейсов**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Номер темы/модуля** | **Наименование практического занятия** | **Описание** |
| **1** | Введение в C++ | Знакомство с инструментарием разработчика | Интегрированные средства разработки.  Простые типы данных. Операции и выражения Управляющие инструкции языка С++  Массивы и указатели. |
| 2 | Объектно-ориентированное программирование | Объектно-ориентированное программирование | Объявления и работа с классами. Абстрактные классы и интерфейсы. Наследование. Шаблоны. |
| 3 | Динамические структуры данных | Динамические структуры данных | Операции вставки, удаления для однонаправленные, двунаправленных и круговых списков, стеков, очередей, деков. |
| 4 | Рекурсивные алгоритмы | Рекурсивные алгоритмы | Обработка древовидных структур данных |
| 5 | Методы тестирования и отладки ПО | Инструменты тестирования и отладки ПО | Встроенные отладчики в средах разработки.  Методы автоматизированного тестирования. |

**8.Оценочные материалы по образовательной программе**

**8.1. Вопросы тестирования по модулям**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ модуля** | **Вопросы входного тестирования** | **Вопросы промежуточного тестирования** | **Вопросы итогового тестирования** |
| **1.** | 1. Программирование это:  а. это процесс создания компьютерной программы  б. это процесс выбора программного продукта под нужды предприятия  в. это настройка и наладка рограммного решения  2.Отладка программы это:  а. процесс поиска и устранения ошибок в программе  б.процесс написания программного кода  в. процесс наладки готовой программы под нужды предприятия  3.Какие типы ошибок могут возникнуть на этапе исполнения программы:  а. логические  б. синтаксические  в. Runtime  4. Блок-схема алгоритма это:  а. графическое представление состава компьютера  б. графическое представление, отражающее логику работы алгоритма или его части  в.схематическое представление конфигурационных файлов для программного решения.  5. Алгоритм это:  а. графическое представление программы  б. конечный набор шагов, которые при следовании им решают какую-то задачу  в. структура программы, определяющая ее функциональные возможности  6. Программные переменные это:  а. конструкции языка программирования, влияющие на ход исполнения программы  б. элементы языка программирования, содержащие исполняемый код программы  в. именованные ячейки памяти, которые используются для хранения данных программы  7. Зарезервированные слова в программировании это:  а. введенные пользователем элементы и типы, которые могут использоваться для написания программы  б. символьные элементы языка, которые не могут использоваться для написания программы, но выполняют зарезервированные функции  в. слова и выражения, которые имеют предопределенные значения и могут использоваться при написании кода  8. Циклы это:  а. структурный элемент программы, содержащий массивы данных  б. части компилятора, использующиеся для многократной сборки программы  в. языковая конструкция, которая может определять участок программы для многократного повторения  9. Что такое документация  а. техническое задание на создание программы, содержащее ее структура, используемые компоненты и описание способов тестирования  б. подробное описание алгоритмов и частей программы, процесса проектирования, тестирования и паттернов правильного использования  в. описание инструментов и средств, необходимых для работы программы  10. Что делает компилятор  а. выполняет код, написанный программистом при запуске программы  б. преобразует исполняемый код в программный  в. преобразуем программный код в исполняемый  11. Что такое исполняемый двоичный код:  а. бинарная форма представления программы, исполняемая процессором  б. двоичное представление массивов данных, загружаемых в программу  в. коды ошибок, возникающие при исполнении программы. | Константы в языке С++.  Типы данных в языке С++. Преобразование типов.  Арифметические и побитовые операции  Операции отношения, логические и дополнительные операции  Указатели в языке С++.  Связь указателей с массивами  Классы памяти и область действия переменных  Функции в языке С++. Связь между функциями через параметры  Ввод/вывод строк. Функции работы со строками.  Директивы препроцессору в языке С++ - include,define.  Структуры. Массивы структур  Связь указателей со структурами  Работа с файлами. Посимвольный и строковый ввод/вывод  Работа с файлами. Форматированный ввод/вывод и ввод/вывод блоками  Динамическое выделение памяти  Объявление имени типа. Абстрактный описатель типа | Константы в языке С++.  Типы данных в языке С++. Преобразование типов.  Арифметические и побитовые операции  Операции отношения, логические и дополнительные операции  Указатели в языке С++.  Связь указателей с массивами  Классы памяти и область действия переменных  Функции в языке С++. Связь между функциями через параметры  Ввод/вывод строк. Функции работы со строками.  Директивы препроцессору в языке С++ - include,define.  Структуры. Массивы структур  Связь указателей со структурами  Работа с файлами. Посимвольный и строковый ввод/вывод  Работа с файлами. Форматированный ввод/вывод и ввод/вывод блоками  Динамическое выделение памяти  Объявление имени типа. Абстрактный описатель типа |
| 2. |  | Достоинства ООП  Недостатки ООП  Описание класс  Поля класса. Методы класса  Описание объектов класса  Хранение полей и методов класса.  Указатели на методы-члены классов  Доступ к элементам класса (private, public, protected)  Конструкторы с параметрами  Копирующий конструктор  Деструкторы  Вызов конструкторов и деструкторов  Объекты класса и указатель this  Встраиваемые или inline – функции  Дружественные функции и дружественные классы  Статические поля класса  Статические методы  Константные поля и константные методы  Наследование и композиция  Базовые классы и производные классы  Управление доступом к элементам базовых классов  Переопределение элементов базового класса в производном классе  Использование конструкторов и деструкторов в производных классах  Функции, которые не наследуются автоматически  Виртуальные функции и полиморфизм  Виртуальные функции  Позднее связывание  Механизм вызова виртуальных функций  Абстрактные классы и конкретные классы  Виртуальные функции и деструкторы  Отсутствие переопределения  Обработка ошибок и исключительные ситуации  Синтаксис исключений  Основная идея механизма обработки исключений  Генерация исключений  Перехват исключений блоком catch  Использование обработчика catch(…)  Когда исключение считается обработанным  Исключения, связанные с ошибкой выделения памяти  Конструкторы, деструкторы и исключения  Спецификация исключений  Алгоритм обработки исключительной ситуации | Достоинства ООП  Недостатки ООП  Описание класс  Поля класса. Методы класса  Описание объектов класса  Хранение полей и методов класса.  Указатели на методы-члены классов  Доступ к элементам класса (private, public, protected)  Конструкторы с параметрами  Копирующий конструктор  Деструкторы  Вызов конструкторов и деструкторов  Объекты класса и указатель this  Встраиваемые или inline – функции  Дружественные функции и дружественные классы  Статические поля класса  Статические методы  Константные поля и константные методы  Наследование и композиция  Базовые классы и производные классы  Управление доступом к элементам базовых классов  Переопределение элементов базового класса в производном классе  Использование конструкторов и деструкторов в производных классах  Функции, которые не наследуются автоматически  Виртуальные функции и полиморфизм  Виртуальные функции  Позднее связывание  Механизм вызова виртуальных функций  Абстрактные классы и конкретные классы  Виртуальные функции и деструкторы  Отсутствие переопределения  Обработка ошибок и исключительные ситуации  Синтаксис исключений  Основная идея механизма обработки исключений  Генерация исключений  Перехват исключений блоком catch  Использование обработчика catch(…)  Когда исключение считается обработанным  Исключения, связанные с ошибкой выделения памяти  Конструкторы, деструкторы и исключения  Спецификация исключений  Алгоритм обработки исключительной ситуации |
| 3. |  | Создание и обработка списков  Однонаправленные и двунаправленные списки  Операции со стеком  Операции с очередью  Операции с деком | Создание и обработка списков  Однонаправленные и двунаправленные списки  Операции со стеком  Операции с очередью  Операции с деком |
| 4. |  | Методика решения задач с помощью рекурсии  Кадры стека и вызов подпрограмм  Создание деревьев  Обход дерева в ширину  Обход дерева в глубину | Методика решения задач с помощью рекурсии  Кадры стека и вызов подпрограмм  Создание деревьев  Обход дерева в ширину  Обход дерева в глубину |
| 5. |  | Типы тестирования ПО. Тестирование чёрного и белого ящиков. Классификация ошибок. Автоматизированное тестирование  Классификация ошибок. Системы логирования фактов.  Средства сбора и анализа данных  Симптоматика ошибок  Отладочные средства | Типы тестирования ПО. Тестирование чёрного и белого ящиков. Классификация ошибок. Автоматизированное тестирование  Классификация ошибок. Системы логирования фактов.  Средства сбора и анализа данных  Симптоматика ошибок  Отладочные средства |

**8.2.**  описание показателей и критериев оценивания, шкалы оценивания

Решение задачи с помощью объектно-ориентированного языка С++. (Зачет/Незачет) **.**

**8.3.**  примеры контрольных заданий по модулям или всей образовательной программе Реализовать класс дерево с операциями вставки вершины. удаления вершины. Нахождения вершины. Реализовать функции работы со стеком (удаление, вставка) Обойти бинарное дерево в глубину в прямом и обратном порядке. Удалить самый высокий(ближайший) лист в дереве. **.**

**8.4.**  тесты и обучающие задачи (кейсы), иные практикоориентированные формы заданий Реализовать класс произвольной предметной области и полями и методами. Объединить n списков в один список. Добавить новый элемент в список перед заданным элементом. Обойти бинарное дерево в ширину. Провести автоматизированное тестирование функции. **.**

**8.5.**  описание процедуры оценивания результатов обучения

Инспектирование разработанного программного обеспечения на наличие логических ошибок, защиты от недопустимых входных данных и стиля программирования **.**

**9.Организационно-педагогические условия реализации программы**

**9.1. Кадровое обеспечение программы**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№**  **п/п** | **Фамилия, имя, отчество (при наличии)** | **Место основной работы и должность, ученая степень и ученое звание (при наличии)** | **Ссылки на веб-страницы с портфолио (при наличии)** | **Фото в формате jpeg** | **Отметка о полученном согласии на обработку персональных данных** |
| **1** | Ляхманов Дмитрий Александрович | НГТУ им Р.Е. Алексеева, к.т.н., доцент |  | C:\Users\ИПС\Downloads\Ляхманов.png | V |
| **2** | Капранов Сергей Николаевич | НГТУ им Р.Е. Алексеева, к.т.н., доцент |  | C:\Users\ИПС\Downloads\Капранов.jpg | V |
| 3 | Дмитриев Дмитрий Валерьевич | НГТУ им Р.Е. Алексеева, к.т.н., доцент |  | C:\Users\ИПС\Downloads\Дмитриев.jpg | V |

**9.2.Учебно-методическое обеспечение и информационное сопровождение**

|  |  |
| --- | --- |
| **Учебно-методические материалы** | |
| Методы, формы и технологии | Методические разработки,  материалы курса, учебная литература |
| Лекции с мультимедийным сопровождением | Программирование на языке C++ : Учеб.пособие. Ч.1 : Введение в программирование на языке C++ / Э.С. Соколова, Д.В. Дмитриев, С.Н. Капранов; НГТУ им.Р.Е.Алексеева. - Н.Новгород : [Б.и.], 2011. - 156 с. - Библиогр.:с.155. - ISBN 978-5-93272-928-1 |
| Практические занятия с использованием интегрированных средств разработки программного обеспечения | Технология программирования : Комплекс учебно-метод.материалов. Ч.1 : Алгоритмы и структуры данных / Ю.С. Бажанов [и др.]; НГТУ им.Р.Е.Алексеева. - Н.Новгород : [Б.и.], 2009. - 126 с. : ил. - Глоссарий:с.121-122. - 182-83. |

|  |  |
| --- | --- |
| **Информационное сопровождение** | |
| Электронные  образовательные ресурсы | Электронные  информационные ресурсы |
| Dpo.nntu.ru | ЭБС Лань <http://e.lanbook.com/>  Договор № 28-14/19-78 от 21.12.2019 г. |
|  | ЭБС Консультант студента (электронная библиотека технического ВУЗа) <http://www.studentlibrary.ru/>  Договор № 28-14/19-77 от 19.12.2019 г. |
|  | ЭБС Юрайт <http://www.biblio-online.ru/>  Договор № от 24.07.2020г. |
|  | Информационно-справочная система «Техэксперт»  Договор № 0332100025419000101 от 30.12.2019 |
|  | Ресурсы Научно-технической библиотеки НГТУ <https://www.nntu.ru/structure/view/podrazdeleniya/nauchno-tehnicheskaya-biblioteka/resursy> |

**9.3.Материально-технические условия реализации программы**

|  |  |
| --- | --- |
| Вид занятий | Наименование оборудования,  программного обеспечения |
| Практические занятия | MS Visual Studio 2012 (freeware) |
| Лекционные занятия | Microsoft Office Professional Plus 2010 Russian Open License Pack NoLevel AcademicEdition, акт предоставления прав №Us000193 от 30.07.2012 |
|  |  |

**III.Паспорт компетенций (Приложение 2)**

Описание перечня профессиональных компетенций, качественное изменение которых осуществляется в результате обучения.

Планируемые результаты обучения должны быть определены в виде знаний, умений, навыков и (или) опыта деятельности, характеризующих формирование/развитие компетенции(-й) в области цифровой экономики и представлены в виде Паспорта компетенций в машиночитаемом текстовом формате. Структура паспорта представлена в приложении.

ПАСПОРТ КОМПЕТЕНЦИИ

Язык программирования С++

Нижегородский государственный технический университет им. Р.Е. Алексеева

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | Наименование компетенции | Способен применять знания в области методов и языков программирования при разработке программного обеспечения | |
| 2. | Указание типа компетенции | профессиональная | |
| 3. | Определение, содержание и основные сущностные характеристики компетенции | Знать:  - этапы решения задач на ЭВМ, основные конструкции языков высокого уровня, структурные типы данных, основы алгоритмизации;  - базовые особенности реализации принципов структурного и объектно-ориентированного программирования;  - базовые методы и технологии разработки программного обеспечения;  Уметь  - оптимизировать алгоритмы для решения прикладных задач;  - выбирать стандартные структуры данных для реализации программного обеспечения;  - разрабатывать специальные структуры данных и алгоритмы их обработки.  Владеть  - практическими навыками реализации программного обеспечения средствами интегрированных сред разработки;  - практическими навыками владения инструментами по отладке программного обеспечения  - практическими навыками владения методами тестирования программного обеспечения | |
| 4. | Дескриптор знаний, умений и навыков по уровням | Уровни сформированности компетенции  обучающегося | Индикаторы |
|  | Начальный уровень  (Компетенция недостаточно развита. Частично проявляет навыки, входящие в состав компетенции. Пытается, стремится проявлять нужные навыки, понимает их необходимость, но у него не всегда получается.) | Знать:  - этапы решения задач на ЭВМ, основные конструкции языков высокого уровня, структурные типы данных, основы алгоритмизации;  Уметь  - реализовывать алгоритмы для решения прикладных задач;  - выбирать стандартные структуры данных для реализации программного обеспечения;  Владеть  - практическими навыками реализации программного обеспечения средствами интегрированных сред разработки; |
|  | Базовый уровень  (Уверенно владеет навыками, способен, проявлять соответствующие навыки в ситуациях с элементами неопределён-ности, сложности.) | Знать:  - этапы решения задач на ЭВМ, основные конструкции языков высокого уровня, структурные типы данных, основы алгоритмизации;  - особенности реализации принципов структурного программирования;  Уметь  - реализовывать алгоритмы для решения прикладных задач;  - выбирать стандартные структуры данных и алгоритмы их обработки для реализации программного обеспечения;  Владеть  - практическими навыками реализации программного обеспечения средствами интегрированных сред разработки;  - практическими навыками владения инструментами по отладке программного обеспечения |
|  | Продвинутый  (Владеет сложными навыками, способен активно влиять на происходящее, проявлять соответствующие навыки в ситуациях повышенной сложности.) | Знать:  - этапы решения задач на ЭВМ, конструкции языков высокого уровня, структурные типы данных, основы алгоритмизации;  - принципы объектно-ориентированного программирования (наследование, полиморфизм);  - имеет представление о современных методы и технологии разработки программного обеспечения;  Уметь  - оптимизировать алгоритмы для решения прикладных задач;  - выбирать стандартные структуры данных для реализации программного обеспечения;  - разрабатывать специальные структуры данных и алгоритмы их обработки.  Владеть  - практическими навыками реализации программного обеспечения средствами интегрированных сред разработки;  - практическими навыками владения инструментами по отладке программного обеспечения  - практическими навыками владения методом тестирования программного обеспечения (черный ящик) |
|  | Профессиональный  (Владеет сложными навыками, создает новые решения для сложных проблем со многими взаимодействую-щими факторами, предлагает новые идеи и процессы, способен активно влиять на происходящее, проявлять соответствующие навыки  в ситуациях повышенной сложности.) | Знать:  - этапы решения задач на ЭВМ, основные конструкции языков высокого уровня, структурные типы данных, основы алгоритмизации;  - особенности реализации принципов структурного и объектно-ориентированного программирования;  - имеет четкое представление о современных методах и технологиях разработки программного обеспечения;  Уметь  - оптимизировать алгоритмы по различным критериям для решения прикладных задач;  - выбирать стандартные структуры данных для реализации программного обеспечения и уметь обосновывать свой выбор;  - разрабатывать специальные структуры данных и алгоритмы их обработки.  Владеть  - практическими навыками реализации программного обеспечения средствами интегрированных сред разработки;  - практическими навыками владения инструментами по отладке программного обеспечения ()  - практическими навыками владения методами тестирования программного обеспечения (в том числе автоматизированного) |
| 5. | Характеристика взаимосвязи данной компетенции с другими компетенциями/ необходимость владения другими компетенциями для формирования данной компетенции | Необходимо владения ПК на уровне уверенного пользователя | |
| 6. | Средства и технологии оценки | Инспектирование разработанного программного обеспечения на наличие логических ошибок, защиты от недопустимых входных данных и стиля программирования | |

**VI.Иная информация о качестве и востребованности образовательной программы** (результаты профессионально-общественной аккредитации образовательной программы, включение в системы рейтингования, призовые места по результатам проведения конкурсов образовательных программ и др.) (при наличии)

-

**V.Рекомендаций к программе от работодателей**: наличие не менее двух писем и/или подтверждения на цифровой платформе Государственной системы предоставления ПЦС от работодателей о рекомендации образовательной программы для реализации в рамках Государственной системы предоставления ПЦС на формирование у трудоспособного населения компетенций цифровой экономики с указанием востребованности результатов освоения программы в сфере деятельности соответствующих компаний и готовности к рассмотрению заявок наиболее успешно освоивших образовательную программу граждан на прохождение стажировки и (или) собеседования на предмет трудоустройства путем проставления отметки в профиле программы

1. Рекомендационное письмо НАПП

2. Рекомендационное письмо ООО "Сборочное оборудование ВИТРИ"

3. Рекомендационное письмо ООО "Металлист-Казаково"

4. Рекомендационное письмо АО "НЗ 70-летия Победы"

5. Рекомендационное письмо АО ПКО "Теплообменник"

**VI.Указание на возможные сценарии профессиональной траектории граждан** по итогам освоения образовательной программы (в соответствии с приложением)

1. Трудоустройство

2. Развитие компетенций в текущей деятельности

3. Переход в новую сферу занятости

**VII.Дополнительная информация**

-

**VIII.Приложенные Скан-копии**

Утвержденной рабочей программа (подпись, печать, в формате pdf)