Министерство образования и науки Хабаровского края
Краевое государственное автономное
профессиональное образовательное учреждение
«Губернаторский авиастроительный колледж г. Комсомольска – на - Амуре
(Межрегиональный центр компетенций)»

ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ОП.06 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

09.02.04. Информационные системы (по отраслям)

Комсомольск – на – Амуре, 2018 г.

СОДЕРЖАНИЕ

	стр.
1. ПАСПОРТ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	4
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	9
3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	14
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	16

1. ПАСПОРТ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ ОП.06 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

1.1. Место дисциплины в структуре основной профессиональной образовательной программы:

Учебная дисциплина «Основы алгоритмизации и программирования» входит в цикл общепрофессиональных дисциплин.

Студент в результате усвоения курса должен уметь самостоятельно анализировать поставленную задачу, разбивать ее на подзадачи. Формулировать и определять этапы решения задачи. Используя язык программирования составлять программный код на данную задачу. Исправлять ошибки и оптимизировать результаты.

1.2. Цели и задачи учебной дисциплины:

Программа ориентирована на выполнение следующих **задач**:

- ознакомить студентов с основными видами алгоритмических конструкций;
- дать студенту знания, которые будут способствовать формированию у них логического мышления, анализа поставленных задач;
- помочь студенту научиться создавать программы с использованием языка программирования.

Цель программы: сформировать, систематизировать полученные знания в области алгоритмизации и программирования.

1.3. Требования к результатам освоения дисциплины:

В результате освоения учебной дисциплины обучающийся должен **уметь**:

- использовать языки программирования,
- строить логически правильные и эффективные программы.

В результате освоения учебной дисциплины обучающийся должен **знать**:

- общие принципы построения алгоритмов, основные алгоритмические конструкции;
- понятие системы программирования;
- основные элементы процедурного языка программирования, структура программы, операторы и операции, управляющие структуры, структуры данных, файлы, кассы памяти;
- подпрограммы, составление библиотек программ;
- объектно-ориентированная модель программирования, понятие классов и объектов, их свойств и методов

Перечень формируемых компетенций:

ПК. 1.2 Взаимодействовать со специалистами смежного профиля при разработке методов, средств и технологий применения объектов профессиональной деятельности.

ПК.1.3 Производить модификацию отдельных модулей информационной системы в соответствии с рабочим заданием, документировать произведенные изменения.
ПК. 2.2 Программировать в соответствии с требованиями технического задания
ПК. 2.3 Применять методики тестирования разрабатываемых приложений

Общие компетенции (ОК)
ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.
ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.
ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.
ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.
ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.
ОК 6. Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями.
ОК **7.** Брать на себя ответственность за работу членов команды (подчиненных), результат выполнения заданий.
ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.
ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.
ОК 10. Исполнять воинскую обязанность, в том числе с применением полученных профессиональных знаний (для юношей).

В дисциплину по требованию работодателя введены *вариативные часы* в объеме **74 часа** для формирования профессионально значимых умений, знаний, компетенций:
- умение отличать свойства алгоритма;
- умение выделять первостепенные свойства алгоритма;
- умение различать способы представления алгоритмов;
- умение решать типовые задачи с составлением алгоритмов (различных типов) с помощью языка блок-схем;
- умение прописывать комментарии, отличать многострочный комментарий от однострочного;
- умение отличать идентификаторы, константы и ключевые слова;
- умение различать типы данных;
- умение объявлять переменные различных типов и инициализировать их;
- умение создавать именованные константы двумя способами;
- умение читать программный код;

- умение решать задачи с использованием логических, арифметических операций, операций отношения, присваивания и др.;
- умение правильно применять операцию явного преобразования типов;
- умение правильно использовать и читать знак инкремента и декремента;
- умение определять вид задачи, т.е. какой оператор будет использован при ее решении;
- умение решать типовые задачи с использованием данных операторов;
- умение создавать проект в системе программирования C++Builder;
- умение находить ошибки в программном коде;
- умение решать задачи на одномерные и двумерные массивы;
- умение формировать массив из чисел соответствующего диапазона;
- умение определять диапазон по программной строке;
- умение различать прототип, определение, обращение;
- умение различать формальные и фактические параметры функции;
- умение решать задачи с использованием пользовательских функций;
- умение правильно использовать локальные и глобальные переменные;
- умение решать задачи с использованием операторов new и delete;
- умение решать задачи на динамические массивы (одномерные и двумерные);
- умение решать задачи на строки;
- умение создавать указатель на поток;
- умение создавать многофайловые проекты;
- умение правильно использовать потоковый ввод-вывод;
- умение считывать и записывать информацию разного типа в файл;
- умение решать задачи с описанием различных структур;
- умение описывать объединения;
- умение определять вид необходимого для использования шаблона;
- умение пользоваться шаблонами разных видов;
- умение решать задачи с использованием шаблонов;
- умение добавлять еще одну форму к проекту;
- умение связывать два и более файлов проекта через директиву препроцессора;
- умение работать с компонентом StringGrid;
- умение описывать свойства объектов и события;
- умение создавать простые проекты.
- знать историю развития данной дисциплины;
- знать цели изучения;
- знать определение понятия алгоритма;
- знать свойства алгоритма;
- знать способы представления алгоритмов;
- знать элементы языка блок-схем;
- знать типы алгоритмов;
- знать определение линейного алгоритма, алгоритма ветвления, циклического алгоритма;
- знать разновидности алгоритмов ветвления и их структуры;
- знать разновидности циклических алгоритмов и их структуры;
- знать этапы обработки программы;

- знать определение понятий: компилятор, линкерование, препроцессор, директива препроцессора, ключевое слово, лексема, идентификатор;
- знать форматы директив препроцессора;
- знать расширение файлов при поэтапной обработке текста программы;
- знать непечатаемые символы;
- знать с каких символов начинается идентификатор;
- знать символы определяющие однострочные и многострочные комментарии;
- знать правила оформления комментариев;
- знать о том что язык C++ отличает прописные и строчные буквы;
- знать базовые типы данных;
- знать модификаторы размера;
- знать модификаторы знака;
- знать формат объявления переменной;
- знать понятие инициализации переменной;
- знать понятие именованной константы;
- знать обозначение целых чисел в: десятичной, восьмеричной и шестнадцатеричной системах счисления;
- знать отличие символьной константы от строковой;
- знать символы арифметических операций;
- знать символы логических операций;
- знать символы операций отношения;
- знать формат операции sizeof;
- знать формат операции условие;
- знать правила деления целых чисел;
- знать особенности записи операции присваивания;
- знать формат операции явного преобразования типа и назначение этой операции;
- знать ранг операций;
- знать назначение операции запятая;
- знать понятие инкремента и декремента;
- знать форматы операторов: scanf, printf, if…else, switch, while, for, goto, break, continue$
- знать управляющие символы;
- знать спецификаторы формата;
- знать математические функции;
- знать определение одномерного и двумерного массива;
- знать форматы объявления массивов;
- знать способы инициализации массивов;
- знать способы задания константного выражение определяющего размер массива;
- знать свойства массивов;
- знать как определяется имя элемента массива;
- знать разницу между элементом массива и его индексом;
- знать правила инициализации элементов массива через генератор случайных чисел;
- знать различные способы сортировки массивов;

- знать определение функции;
- знать понятие возвращаемого значения;
- знать понятия прототип функции, определение функции и обращение к функции, параметры функции;
- знать форматы прототипа, определения и обращения;
- знать разновидности функций;
- знать структуру программы содержащую пользовательские функции;
- знать назначение оператора return и его форматы;
- знать особенности объявления и присвоения имен локальным и глобальным переменным;
- знать определение указателя и его назначение;
- знать формат объявления указателя;
- знать назначение операции взятия адреса переменной;
- знать формат оператора new и delete;назначение этих операторов;
- знать понятие динамической памяти, ее особенности;
- знать принцип выделения памяти под одномерные и двумерные динамические массивы;
- знать различия между динамическими и статическими массивами;
- знать понятие утечки памяти, и в каких случаях она возникает;
- знать особенности организации символьных массивов;
- знать различие между строковой константой и символьной;
- знать способы и правила инициализации символьных массивов;
- знать определение потока; форматы операторов: fopen, fclose, fprintf и др.;
- знать режимы открытия потока; отличительные особенности различных видов потоков;
- знать понятие указатель на поток и его назначение;
- знать особенности записи и чтения целых чисел;
- знать формат объявления структуры;
- знать определение структуры;
- знать понятие шаблона; назначение шаблона; форматы описания;
- знать организацию работы в программе C++ Builder;
- знать правила описания свойств объектов; правила создания и описания событий объектов;
- знать особенности работы с компонентом StringGrid и его основные свойства;
- знать особенности преобразования типов в C++ Builder.

ПК 2.7. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.

1.4. количество часов на освоение программы учебной дисциплины:
максимальной учебной нагрузки обучающегося 240 часа, в том числе:
обязательной аудиторной учебной нагрузки обучающегося 160 часов;
самостоятельной работы обучающегося 80 часа.