**Цикл дисциплин направления**

**Информатика и программирование**

**1, 2, 3, 4 модуль 2015/2016 уч. год**

1. Краткое содержание дисциплины.

Изучение основ и терминологии низкоуровнего программирования на языке Ассемблер и объектно-ориентированного программирования. Формирование представления о возможностях и особенностях языка Ассемблер и объектно-ориентированных языков программирования при проектировании, разработке и отладке компьютерных программ.

1. Временной ресурс /Кредитная стоимость дисциплины: 228 ч. / 6 ЗЕТ
2. Цель: Изучение возможностей базовых принципов низкоуровневого и объектно-ориентированного программирования, применяемых при разработке и реализации алгоритмов, создании и отладке компьютерных программ, совместимости систем программирования, встраивание программных секций на языке Ассемблер низкого уровня в программу на языке высокого уровня. Формирование практических навыков разработки прикладных программ на языке программирования Ассемблер и C++.
3. Результаты обучения: В результате освоения дисциплины студент должен:

**Знать:**

* парадигмы и методологии программирования, особенности языка Ассемблер и объектно-ориентированных языков программирования, наиболее широко используемых языков и средств программирования; концепции, синтаксической и семантической организации, методы использования современных объектно-ориентированных языков программирования; теоретические и методические основы, понимать функциональные возможности в области разработки информационных систем (УК-3, УК-4, УК-5, УК-6, ПК-4, ПК-8, ПК-11);

**Уметь:**

* применять в профессиональной деятельности язык Ассемблер и современные объектно-ориентированные языки программирования; профессионально решать задачи производственной и технологической деятельности с учетом современных достижений науки и техники, включая: разработку алгоритмических и программных решений в области системного и прикладного программирования; разработку математических, информационных и имитационных моделей по тематике выполняемых исследований (УК-3, УК-4, УК-5, УК-6, ПК-4, ПК-8, ПК-11);

**Владеть:**

* базовыми математическими знаниями и информационными технологиями, эффективно применять их для решения научно-технических задач и прикладных задач, связанных с развитием и использованием информационных технологий (УК-3, УК-4, УК-5, УК-6, ПК-4, ПК-8, ПК-11);

**Иметь навыки (приобрести опыт):**

* постановки математических и информационных задач, разработки технического задания по проектированию программного обеспечения, применения математических методов к решению практических задач, описания алгоритмов решения задачи, разработки программного кода на языках Ассемблер и C++, тестирования программного обеспечения, подбора и анализа материалов по теме задания, презентации и защиты полученных результатов (УК-3, УК-4, УК-5, УК-6, ПК-4, ПК-8, ПК-11).

В результате освоения дисциплины студент осваивает следующие компетенции:

| Компетенция | Код по ФГОС/ НИУ |
| --- | --- |
| Способен решать проблемы в профессиональной деятельности на основе анализа и синтеза | УК-3 |
| Способен оценивать потребность в ресурсах и планировать их использование при решении задач в профессиональной деятельности | УК-4 |
| Способен работать с информацией: находить, оценивать и использовать информацию из различных источников, необходимую для решения научных и профессиональных задач (в том числе на основе системного подхода) | УК-5 |
| Способен вести исследовательскую деятельность, включая анализ проблем, постановку целей и задач, выделение объекта и предмета исследования, выбор способа и методов исследования, а также оценку его качества | УК-6 |
| Способен проектировать и разрабатывать компоненты программного обеспечения на основе современных парадигм, технологий и языков программирования | ПК-4 |
| Способен применять знания жизненного цикла современных проектов по созданию и эксплуатации программных систем и инструментальные средства управления проектами в области ИТ. | ПК-8 |
| Способен использовать и развивать методы математического моделирования и применять аналитические и научные пакеты прикладных программ | ПК-11 |

1. Содержание:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название раздела | Всего часов | ЗЕТ |
|  | **Поколения языков программирования.** Поколения языков программирования по уровню абстрагирования от конкретного типа аппаратной части (1 машинные, 2 ассемблеры, 3 процедурные, объектно-ориентированные, гибридные, доменоспециализированные, декларативные, 4 скриптовые, описательные, 5 логические, функциональные языки). | 7 | 0,2 |
|  | **Архитектура микропроцессоров системы команд IA-32 и IA-64.** Архитектура микропроцессоров IA-32 (основные устройства). Архитектура RISC– и CISC–процессоров (группы команд). Совместимость систем программирования. Встраивание программных секций на ассемблере в программу на языке высокого уровня. Поддержка технологии сегментации трансляторами MASM и TASM, компилятором Visual С++ 2010. Принципы архитектуры IA-64 (Intel). Сравнение IA-64 и традиционных архитектур. Принципы архитектуры x86-64 (AMD). Сравнения архитектур: x86-64 против IA-64. | 22 | 0,3 |
|  | **Программирование на языке Ассемблер.** Классический подход к разработке программного обеспечения. Модель вычислений, поддерживаемая языком Ассемблер. Синтаксис ассемблера. Формат команд и макрокоманд. Операнды. Операторы. Директивы сегментации. Типы данных. Макросредства Ассемблера. | 26 | 0,4 |
|  | **Введение в программирование на С**++. Пространства имен. Ввод/вывод в С++. Операции инкремента и декремента. Особенности операторов инкремента и декремента при работе с указателями. Ссылки. Выделение/освобождение памяти в С++. Константы и макроопределения. | 13 | 0,6 |
|  | **Знакомство с библиотекой STL.** Понятие объекта и класса. Классы стандартной библиотеки для работы со строками. Классы стандартной библиотеки для работы с контейнерами. Операции с объектами стандартной библиотеки. Преобразование типов. | 18 | 0,5 |
|  | **Построение пользовательских типов данных**. Создание пользовательских классов. Принцип композиции при конструировании новых классов. Инициализация данных объекта класса (Конструктор по умолчанию, конструктор с параметрами). Назначение деструктора. Способы передачи объекта в функцию, конструктор копирования. Перегрузка операторов. Статические переменные и методы класса. Константные методы. Перегруженные методы. | 18 | 0,5 |
|  | **Наследование, виртуальные функции, полиморфизм**. Спецификаторы доступа public, protected, private. Понятие и механизм наследования классов. Преимущества и недостатки наследования по сравнению с композицией. Виртуальные функции, понятие полиморфизма. Чистые виртуальные функции и абстрактные классы. Доступ к объекту через указатель, преобразование типов. Непрямые базовые классы. Виртуальный деструктор. Множественное наследование. | 38 | 0,5 |
|  | **Концепция программирования для Windows. Знакомство с библиотекой MFC.** Функция WinMain. Цикл обработки сообщений. Минимальная программа для Windows. Использование библиотеки MFC. Класс Capp. Основные элементы программ на базе MFC. Типы классов окон MFC. Основные сообщения Winwows. Графические возможности GDI. | 20 | 0,5 |
|  | **Элементы пользовательского интерфейса**. Создание и работа с меню. Создание и работа с панелями инструментов. Поддерживаемые элементы управления (кнопки, текстовые поля, списки и т.д.). Создание, редактирование, получение уведомлений от элементов управления. | 18 | 0,5 |
|  | **Архитектура документ-представление**. Использование мастера Visual Studio для создания SDI и MDI приложений. Архитектура SDI приложений. Архитектура MDI приложений. Различные представления одного документа. Уведомление представлений об изменении документа. Понятие сериализации, сохранение и печать документа. | 18 | 0,5 |
|  | **Шаблонны С++.** Понятие обобщенного программирования. Определение шаблона класса. Создание объектов шаблона класса. Наследование шаблонных классов. Шаблоны функций. | 14 | 0,5 |
|  | **Отладка программ в Visual Studio. Обработка исключений.** Трассировка программы в Visual Studio. Макросы ASSERT, VERIFY и TRACE. Понятие исключения. Механизм генерации и обработки исключений. | 14 | 0,5 |
|  | **Многопоточное программирование в MFC.** Создание потоков. Рабочие и интерфейсные потоки. Синхронизация потоков. Остановка потоков. | 14 | 0,5 |
|  | Всего: | **228** | 6,0 |

1. Пререквизиты: Отсутствуют.

Для успешного усвоения дисциплины студенты должны:

* Уметь пользоваться компьютером и интернетом;
* Владеть основами информатики, алгоритмизации и программирования на языке С;
* Знать методы и базовые алгоритмы обработки информационных структур, методы анализа сложности алгоритмов;
* Уметь ясно строить устную и письменную речь;
* Уметь пользоваться библиографией, в том числе на английском языке, понимать суть найденной ссылки или статьи.

1. Основная литература:

Базовый учебник

1. Магда Ю.С. Ассемблер для процессоров Intel Pentium. –СПб.: Питер, 2006.-410с., ISBN 5-469-00662-X

Основная литература

1. Внуков А.А. Лекции в электронной форме по курсу Информатика и программирование.– файл: Архитектура системы команд и язык Ассемблер.doc Лекции, 1-й модуль 2015. -145 с.
2. Пирогов В.Ю. ASSEMBLER. Учебный курс. - М.: Издатель Молгачева С.В., Издательство Нолидж, 2001. - 848 с., ил., ISBN 5-89251-101-4
3. Голубь Н.Г. Искусство программирования на Ассемблере. Лекции и упражнения. -2 изд., испр. и доп. –СПб.: ООО «ДиаСофтЮП». 2002.-656 с. ISBN 5-93772-056-3
4. Рудаков П.И., Финогенов К.Г. Язык ассемблера: уроки программирования. – ДИАЛОГ-МИФИ, 2001. -640 с. ISBN 5-86404-160-2
5. Ирвин К. Язык ассемблера для процессоров Intel. 3-е изд. / Пер. с анг. – М:. Издательский дом «Вильямс», 2005 -ISBN: 5-8459-0779-9
6. Внуков А.А., Раев А.П., Можаитин Р.В. Организация ЭВМ, комплексов и систем. Учебное пособие — М.: Московский государственный институт электроники и математики, 2001. — 112 с. ISBN 5-230-16343-7
7. Внуков А.А. Лекции в электронной форме по курсу Архитектура ЭВМ и систем – файл: МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ИНСТИТУТ ЭЛЕКТРОНИКИ И МАТЕМАТИКИ -05.doc, лекции, 4 семестр 2013. -142 с.
8. Внуков А.А. Лекции в электронной форме по курсу Архитектура ЭВМ и систем – файл: Архитектура\_ЭВМ+система\_команд 2009.doc, лекции, 4 семестр 2009. -128 с.
9. Ivor Horton, Beginning Visual C++ 2010, Indianapolis : Wiley Publishing, Inc., 2011,
10. P.J. Deitel, H.M. Deitel, C++ How to Program, 7th Pearson Education, 2010,
11. Дополнительная литература
12. IA-32 Intel Architecture Software Developer’s Manual, Intel® Corp. 2001
13. IA-32 Intel Architecture Optimization, Intel® Corp. 2001
14. Э. Кенинг, Б. Му, Эффективное программирование на С++ // М. : «Вильямс», 2002,
15. Айвор Хортон, Visual C++ 2010 полный курс, Москва : Вильямс, 2011
16. С. Мейерс, Эффективное использование С++: 50 рекомендаций по улучшению ваших программ и проектов // Москва : Питер, 2006,
17. С. Мейерс, Эффективное использование STL // Москва : Питер, 2002,
18. Г. Саттер, Решение сложных задач на С++ // Москва : Вильямс, 2002,
19. А. Александреску, Современное проектирование на С++ // Москва : Вильямс, 2002,
20. Координатор: Внуков Андрей Анатольевич, к.т.н., доцент, 8(495) 772-95-90 \*15102, a.vnukov@hse.ru
21. Использование технических средств обучения: для проведения лекций необходима аудитория с компьютером и проекционным оборудованием, для - практических занятий компьютерный класс с современным программным обеспечением, IDE для разработки программ, Visual С++ 2010, среда ОС Windows 7;
22. Лабораторные работы: 36 часов.

Преподаватель: Аксенов Сергей Алексеевич

Преподаватель: Внуков Андрей Анатольевич