

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учебно-методическое объединение по образованию
в области информатики и радиоэлектроники

УТВЕРЖДАЮ

Первый заместитель Министра образования
Республики Беларусь

_____ И.А. Старовойтова

Регистрационный № ТД-_____/тип.

КОНСТРУИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Типовая учебная программа по учебной дисциплине
для специальности

1-40 01 01 Программное обеспечение информационных технологий

СОГЛАСОВАНО

Начальник Управления
электроники и приборостроения,
электротехнической и оптико-
механической промышленности
Министерства промышленности
Республики Беларусь

_____ А.С. Турцевич

СОГЛАСОВАНО

Председатель Учебно-методического
объединения по образованию в
области информатики и
радиоэлектроники

_____ В.А. Богуш

СОГЛАСОВАНО

Начальник Главного управления
профессионального образования
Министерства образования
Республики Беларусь

_____ С.А. Касперович

СОГЛАСОВАНО

Проректор по научно-методической
работе Государственного учреждения
образования «Республиканский
институт высшей школы»

_____ И.В. Титович

Эксперт-нормоконтролер

СОСТАВИТЕЛЬ:

А.И. Парамонов, заведующий кафедрой информационных систем и технологий Института информационных технологий учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат технических наук, доцент;

Д.Е. Оношко, старший преподаватель кафедры программного обеспечения информационных технологий учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», магистр технических наук.

РЕЦЕНЗЕНТЫ:

Кафедра технологий программирования Белорусского государственного университета (протокол № 4 от 18.11.2021 г.);

Н.В. Пацей, заведующий кафедрой программной инженерии учреждения образования «Белорусский государственный технологический университет», кандидат технических наук, доцент.

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ В КАЧЕСТВЕ ТИПОВОЙ:

Кафедрой программного обеспечения информационных технологий учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № 7 от 15.11.2021 г.);

Научно-методическим советом учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол №__ от _____);

Научно-методическим советом по разработке программного обеспечения и информационно-коммуникационным технологиям Учебно-методического объединения по образованию в области информатики и радиоэлектроники (протокол № 4 от 02.12.2021 г.).

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

ХАРАКТЕРИСТИКА УЧЕБНОЙ ДИСЦИПЛИНЫ

Типовая учебная программа по учебной дисциплине «Конструирование программного обеспечения» разработана для студентов учреждений высшего образования, обучающихся по специальности 1-40 01 01 «Программное обеспечение информационных технологий» в соответствии с требованиями образовательного стандарта высшего образования первой ступени и типового учебного плана вышеуказанной специальности.

От современного специалиста в сфере программного обеспечения информационных технологий требуется, среди прочего, способность в кратчайшие сроки осваивать новые языки программирования, технологии и инструменты разработки программного обеспечения. Освоение учебной дисциплины «Конструирование программного обеспечения» закладывает основы для понимания студентами механизмов выполнения программ, их взаимодействия с операционными системами, аппаратным обеспечением, дополняет знания, полученные ранее в рамках учебной дисциплины «Основы алгоритмизации и программирования», и представления студентов о языках программирования, а также принципах проектирования программных средств, парадигмах программирования. Это позволяет студенту в последующей профессиональной деятельности опираться на более разнообразный опыт, а также на понимание общих принципов и механизмов, сокращая тем самым время изучения новых языков, технологий и инструментов разработки программного обеспечения.

Учебная дисциплина «Конструирование программного обеспечения» обеспечивает второй этап подготовки специалиста в области программирования: переход от разработки простых алгоритмов к проектированию сложных программных средств, в т.ч. требующих взаимодействия программы с другими программными и аппаратными компонентами.

Понимание внутреннего устройства программных средств и механизмов, задействованных при их выполнении процессором, позволяет студенту оценивать качество предлагаемых алгоритмических и архитектурных решений, отдавая предпочтение более эффективным, подходящим более широкому кругу пользователей и т.д.

ЦЕЛЬ, ЗАДАЧИ, РОЛЬ УЧЕБНОЙ ДИСЦИПЛИНЫ

Цель учебной дисциплины: систематизированная подготовка специалиста, способного осуществлять разработку программных средств на языках программирования низкого и высокого уровней, применять сопутствующие знания и навыки для отладки программных средств, соблюдения ограничений, установленных требованиями к конечному продукту.

Задачи учебной дисциплины:

приобретение знаний о языках программирования низкого и высокого уровней, о разработке программ с учётом особенностей языка программирования и целевой программно-аппаратной платформы;

изучение принципов взаимодействия программных средств с операционной системой и другими программными средствами, принципов конструирования программных средств в соответствии с заданными требованиями к ним;

приобретение навыков разработки программных средств на языках программирования низкого и высокого уровней, отладки и тестирования программных средств с учётом особенностей целевой программно-аппаратной платформы;

овладение методами исследования исходного и исполняемого (машинного) кода программ, написанных на языках низкого и высокого уровня.

Базовыми учебными дисциплинами по курсу «Конструирование программного обеспечения» являются «Основы алгоритмизации и программирования», «Алгоритмы и структуры данных», «Основы программной инженерии». В свою очередь учебная дисциплина «Конструирование программного обеспечения» является базой для таких учебных дисциплин, как «Системное программирование», «Объектно-ориентированные технологии программирования и стандарты проектирования».

ТРЕБОВАНИЯ К УРОВНЮ ОСВОЕНИЯ СОДЕРЖАНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

В результате изучения учебной дисциплины «Конструирование программного обеспечения» формируются следующие компетенции:

универсальные:

владеть основами исследовательской деятельности, осуществлять поиск, анализ и синтез информации;

обладать навыками саморазвития и самосовершенствования в профессиональной деятельности;

проявлять инициативу и адаптироваться к изменениям в профессиональной деятельности;

базовые профессиональные:

использовать в разработке программных продуктов формальные методы конструирования программного обеспечения, оценки сложности алгоритмов и их практической реализации.

В результате изучения учебной дисциплины обучающийся должен:

знать:

теоретические основы разработки программ с учётом особенностей языка программирования и целевой программно-аппаратной платформы;

синтаксис и семантику машинно-ориентированного языка программирования низкого уровня;

синтаксис и семантику процедурно-ориентированного языка высокого уровня, компилируемого непосредственно в машинный код;

уметь:

выбирать платформу, язык программирования и инструментарий для решения поставленной задачи;

разрабатывать программные средства на языках программирования высокого и низкого уровней;

исследовать исходный и исполняемый (машинный) код программ, написанных на языках программирования высокого и низкого уровней;

выполнять отладку и тестирование программ, написанных на языках программирования высокого и низкого уровней;

владеть:

навыками работы в средах программирования;

навыками разработки, отладки и тестирования программ, написанных на машинно-ориентированном языке программирования;

навыками разработки, отладки и тестирования программ, написанных на процедурно-ориентированном языке высокого уровня;

навыками применения возможностей препроцессоров (макрорасширений) языков программирования;

способностью применения дополнительных средств языков программирования (например, стандартной библиотекой языка C, наборами инструкций SSE, AVX в языке ассемблера);

навыками вставки текста программы, написанной на языке низкого уровня, в текст программы, написанной на языке высокого уровня.

В рамках образовательного процесса по учебной дисциплине «Конструирование программного обеспечения» студент должен приобрести не только теоретические и практические знания, умения и навыки по специальности, но и развить свой ценностно-личностный, духовный потенциал, сформировать качества патриота и гражданина, готового к активному участию в экономической, производственной, социально-культурной и общественной жизни страны.

Программа рассчитана на 210 учебных часов, из них – 120 аудиторных.

Примерное распределение аудиторных часов по видам занятий: лекций – 60 часов, лабораторных занятий – 60 часов. Курсовой проект – 40 часов.

ПРИМЕРНЫЙ ТЕМАТИЧЕСКИЙ ПЛАН

Наименование раздела, темы	Всего ауди- торных	Лекции	Лабораторные занятия
Раздел 1. Машинно-ориентированный язык программирования	60	30	30
Тема 1. Основы программирования на машинно-ориентированном языке	4	4	-
Тема 2. Арифметические и побитовые операции	6	2	4
Тема 3. Организация ветвлений и циклов	6	2	4
Тема 4. Обработка строковых данных	6	2	4
Тема 5. Процедуры	12	4	8
Тема 6. Работа с видеопамятью	4	2	2
Тема 7. Прерывания	8	4	4
Тема 8. Работа с вещественными числами	4	2	2
Тема 9. Работа с портами ввода-вывода	4	2	2
Тема 10. Обзор расширенных возможностей процессора	2	2	-
Тема 11. Аппаратные оптимизации	4	4	-
Раздел 2. Язык программирования высокого уровня	60	30	30
Тема 12. Основы программирования на языке высокого уровня	4	4	-
Тема 13. Операторы управления вычислительным процессом	4	2	2
Тема 14. Массивы, строки и указатели	8	4	4
Тема 15. Функции	8	4	4
Тема 16. Классы хранения и видимость переменных	2	2	-
Тема 17. Структуры, объединения и перечисления	6	2	4
Тема 18. Работа с файлами	6	2	4
Тема 19. Директивы препроцессора	2	2	-
Тема 20. Модули и компоновка	6	2	4
Тема 21. Неопределённое поведение	2	2	-
Тема 22. Структуры данных	10	2	8
Тема 23. Парадигмы программирования	2	2	-
Итого:	120	60	60

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. МАШИННО-ОРИЕНТИРОВАННЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ

Тема 1. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА МАШИННО-ОРИЕНТИРОВАННОМ ЯЗЫКЕ

Ассемблер и язык ассемблера. Понятие инструкции. Операнды и их виды. Классификация регистров. Регистр флагов. Регистр программного счётчика. Принцип работы ассемблера. Машинный код. Понятие точки входа. Простейшая программа на языке ассемблера. Метки. Комментарии. Директивы определения данных. Механизм запуска программ операционной системой. Понятия сегмента и смещения. Полный адрес. Физический адрес. Эффективный адрес.

Тема 2. АРИФМЕТИЧЕСКИЕ И ПОБИТОВЫЕ ОПЕРАЦИИ

Сложение и вычитание знаковых и беззнаковых чисел. Сложение и вычитание чисел большой разрядности. Инкремент и декремент. Изменение знака. Умножение и деление. Знаковое и беззнаковое расширение чисел. Инструкции изменения флагов. Побитовые сдвиги.

Тема 3. ОРГАНИЗАЦИЯ ВЕТВЛЕНИЙ И ЦИКЛОВ

Инструкции условного и безусловного перехода. Специализированные инструкции для проверки условий. Организация циклов с использованием инструкций условного и безусловного перехода. Специализированные инструкции для организации циклов.

Тема 4. ОБРАБОТКА СТРОКОВЫХ ДАННЫХ

Инструкции для обработки строковых данных (строковые инструкции). Префиксы повторения: условные и безусловный.

Тема 5. ПРОЦЕДУРЫ

Стек. Инструкции для работы со стеком. Понятие процедуры. Вызов процедуры. Возврат из процедуры. Способы передачи параметров. Способы возврата значений. Пролог и эпилог. Понятие дальнего вызова. Дальний возврат.

Тема 6. РАБОТА С ВИДЕОПАМЯТЬЮ

Понятие видеоадаптера. Переключение режимов видеоадаптера. Понятие видеопамати. Запись и чтение видеопамати.

Тема 7. ПРЕРЫВАНИЯ

Понятие прерывания. Обработчик прерывания. Виды прерываний. Таблица векторов прерываний. Программные прерывания. Механизм вызова обработчика прерывания. Возврат из обработчика прерывания. Аппаратные прерывания. Разрешение и запрет отработки аппаратных прерываний. Перехват прерываний.

Тема 8. РАБОТА С ВЕЩЕСТВЕННЫМИ ЧИСЛАМИ

Принципы работы с FPU. Поддерживаемые целочисленные и вещественные форматы. Инструкции загрузки, пересылки и выгрузки данных. Арифметические инструкции. Тригонометрические инструкции.

Тема 9. РАБОТА С ПОРТАМИ ВВОДА-ВЫВОДА

Инструкции для работы с портами ввода-вывода. Примеры взаимодействия с legacy-устройствами через порты ввода-вывода: управление регистрами видеоадаптера, вывод звука.

Тема 10. ОБЗОР РАСШИРЕННЫХ ВОЗМОЖНОСТЕЙ ПРОЦЕССОРА

Режимы работы процессора. Эмуляция многозадачности с использованием механизма обработки прерываний. Расширенные возможности процессора для поддержки многозадачности. Механизмы защиты. Сегментная и страничная адресация. Понятие адресного пространства. Виртуальная память. Расширенные наборы инструкций. Параллельные вычисления и SIMD-инструкции.

Тема 11. АППАРАТНЫЕ ОПТИМИЗАЦИИ

Конвейерная архитектура. Понятие конфликта. Виды конфликтов. Предсказание переходов: статическое и динамическое. Оптимизация вызовов подпрограмм. Внеочередное (спекулятивное) исполнение инструкций. Переименование регистров. Суперскалярная архитектура. Предзагрузка инструкций. Кэши процессора. Когерентность кэшей. RISC- и CISC-архитектуры, гибридная архитектура.

Раздел 2. ЯЗЫК ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ

Тема 12. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ

Основные типы данных. Синтаксис объявления переменных. Модификаторы. Структура программы. Основные средства ввода-вывода. Простейшая программа на языке высокого уровня. Арифметические операции. Преобразование типов. Поразрядные (битовые) операции. Операции сдвига. Операции сравнения. Логические операции и их отличия от одноимённых поразрядных операций. Приоритеты операций.

Тема 13. ОПЕРАТОРЫ УПРАВЛЕНИЯ ВЫЧИСЛИТЕЛЬНЫМ ПРОЦЕССОМ

Оператор проверки условия. Оператор безусловной передачи управления. Метки. Оператор множественного выбора. Оператор цикла с параметром. Оператор цикла с предусловием. Оператор цикла с постусловием. Операторы выхода из цикла и перехода к следующей итерации.

Тема 14. МАССИВЫ, СТРОКИ И УКАЗАТЕЛИ

Объявление и использование массивов. Инициализация массивов. Одномерные и многомерные массивы. Указатели. Нулевой указатель. Операции над указателями. Типизированные и нетипизированные указатели. Связь массивов

и указателей. Динамическое выделение памяти. Взаимосвязь строковых данных и массивов. Объявление строковых данных. Работа со строковыми данными.

Тема 15. ФУНКЦИИ

Объявление функции. Передающее и определяющее описание. Возврат значения из функции. Области видимости элементов программы. Способы передачи параметров. Рекурсивные функции. Явная и неявная рекурсия. Указатели на функции. Синтаксис объявления функций с переменным числом параметров. Доступ к параметрам.

Тема 16. КЛАССЫ ХРАНЕНИЯ И ВИДИМОСТЬ ПЕРЕМЕННЫХ

Понятие класса хранения. Автоматический класс хранения. Регистровый класс хранения. Статический класс хранения. Внешний класс хранения.

Тема 17. СТРУКТУРЫ, ОБЪЕДИНЕНИЯ И ПЕРЕЧИСЛЕНИЯ

Структуры. Объявление и инициализация структурных переменных. Работа со структурами. Вложенные структуры. Указатели на структуры. Объявление пользовательских типов. Объединения. Битовые поля. Перечисления.

Тема 18. РАБОТА С ФАЙЛАМИ

Понятие файла. Текстовый и бинарный режимы работы с файлами. Режимы открытия файлов. Закрытие файлов. Стандартные потоки ввода/вывода. Функции чтения/записи. Последовательный и произвольный доступ к файлу. Функции управления буферизацией файлового ввода/вывода.

Тема 19. ДИРЕКТИВЫ ПРЕПРОЦЕССОРА

Назначение и принцип работы препроцессора. Директивы препроцессора и макропеременные. Работа с макропеременными. Макросы. Преимущества и недостатки макросов по сравнению с функциями. Inline-функции. Условная компиляция. Операции препроцессора.

Тема 20. МОДУЛИ И КОМПОНОВКА

Компоновка как этап сборки проекта. Организация модулей. Понятия объектного файла и библиотеки. Принцип работы компоновщика. Понятие импорта и экспорта. Декорирование имён.

Тема 21. НЕОПРЕДЕЛЁННОЕ ПОВЕДЕНИЕ

Неопределённое поведение, неуточняемое поведение и поведение, определяемое реализацией. Понятие точки следования. Связь точек следования с побочными эффектами операций.

Тема 22. СТРУКТУРЫ ДАННЫХ

Обзор основных структур данных и способов их реализации на языке высокого уровня. Понятие согласованности данных. Понятие транзакции и ато-

марной операции. Обеспечение транзакционности при работе со структурами данных.

Тема 23. ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ

Понятие парадигмы программирования. Обзор популярных парадигм программирования и их реализации в языках программирования.

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ**ЛИТЕРАТУРА****ОСНОВНАЯ**

1. Intel® 64 and IA-32 Architectures Software Developer's Manual. – Intel Corporation, 2021. – 4778 p.
2. AMD64 Architecture Programmer's Manual. – Advanced Micro Devices, Inc., 2021. – 3269 p.
3. International Standard ISO/IEC 9899:2018. – Programming languages C.
4. International Standard ISO/IEC 14882:2020. – Programming Language C++.
5. Ирвин, К. Язык ассемблера для процессоров Intel / К. Ирвин. – Москва : Вильямс, 2005. – 912 с.
6. Абель, П. Ассемблер. Язык и программирование для IBM PC / П. Абель. – Москва : Век, 2009. – 736 с.
7. Зубков, С. В. Assembler для DOS, Windows и UNIX / С. В. Зубков. – Москва : ДМК Пресс, 2017. – 638 с.
8. Пирогов, В. Ю. Ассемблер и дизассемблирование / В. Ю. Пирогов. – СПб. : БХВ-Петербург, 2007. – 464 с.
9. Магда, Ю. С. Ассемблер для процессоров Intel Pentium / Ю. С. Магда. – СПб. : Питер, 2006. – 410 с.
10. Голубь, Н. Г. Искусство программирования на Ассемблере / Н. Г. Голубь. – СПб. : Питер, 2006. – 832 с.
11. Пильщиков, В. Н. Assembler. Программирование на языке ассемблера IBM PC / В. Н. Пильщиков. – Москва : Диалог-МИФИ, 2010. – 288 с.
12. Себеста, Р. Основные концепции языков программирования / Р. Себеста. – Москва : Вильямс, 2001. – 672 с.
13. Керниган, Б. Практика программирования / Б. Керниган, Р. Пайк. – Москва : Вильямс, 2017. – 288 с.
14. Керниган, Б. Язык программирования C / Б. Керниган, Д. Ритчи. – Москва : Вильямс, 2017. – 288 с.
15. Кочан, С. Программирование на языке C / С. Кочан. – Москва : Вильямс, 2007. – 496 с.
16. Харибсон, С. Язык программирования C / С. Харибсон, Г. Стил. – Москва : БИНОМ, 2009. – 526 с.
17. Демидович, Е. М. Основы алгоритмизации и программирования. Язык СИ / Е. М. Демидович. – СПб. : БХВ-Петербург, 2008. – 440 с.

ДОПОЛНИТЕЛЬНАЯ

18. Касперски, К. Искусство дизассемблирования / К. Касперски, Е. Рокко. – СПб. : БХВ-Петербург, 2008. – 896 с.
19. Шилдт, Г. Искусство программирования на C++ / Г. Шилдт. – СПб. : БХВ-Петербург, 2005. – 496 с.

20. Шелест, В. Д. Программирование: Структурный подход. Алгоритмы. Turbo Pascal. Borland C++. Современный Fortran / В. Д. Шелест. – СПб. : БХВ-Петербург, 2002. – 585 с.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

При изучении учебной дисциплины рекомендуется использовать следующие формы самостоятельной работы:

решение дополнительных задач повышенной сложности, предлагаемых преподавателем;

участие в конкурсе студенческих работ;

представление результатов самостоятельной работы в рамках студенческих конференций.

ПЕРЕЧЕНЬ РЕКОМЕНДУЕМЫХ СРЕДСТВ ДИАГНОСТИКИ КОМПЕТЕНЦИЙ СТУДЕНТА

Типовым учебным планом по специальности 1-40 01 01 «Программное обеспечение информационных технологий» в качестве формы текущей аттестации по учебной дисциплине «Конструирование программного обеспечения» рекомендуется зачёт, экзамен и курсовой проект.

Оценка учебных достижений студента по итогам зачёта производится по системе «зачтено/не зачтено», по итогам экзамена – по десятибалльной шкале.

Для промежуточного контроля по учебной дисциплине и диагностики компетенций студентов могут использоваться следующие формы:

отчеты по лабораторным работам;

контрольные работы;

устные опросы;

электронные тесты по разделам учебной дисциплины;

доклады;

защита курсового проекта;

коллоквиум;

защита индивидуальных практических работ.

РЕКОМЕНДУЕМЫЕ МЕТОДЫ (ТЕХНОЛОГИИ) ОБУЧЕНИЯ

Основные рекомендуемые методы (технологии) обучения, отвечающие целям и задачам учебной дисциплины:

проблемное обучение (проблемное изложение, вариативное изложение, частично-поисковый метод), реализуемое на лекционных занятиях;

учебно-исследовательская деятельность, творческий подход, реализуемые на лабораторных занятиях и в разработки курсового проекта;

проектные технологии, используемые при проектировании конкретного объекта, реализуемые при выполнении самостоятельной работы.

КУРСОВОЕ ПРОЕКТИРОВАНИЕ

Целью курсового проектирования является углубление теоретических знаний и практических навыков в области разработки программных средств, решающих реальные практические задачи, развитие навыков самостоятельного изучения предметной области поставленной задачи.

В ходе курсового проектирования студент получает индивидуальное задание, осуществляет анализ предметной области, формулирует требования к предполагаемому программному средству, выбирает инструменты для его разработки (язык программирования, среду разработки, вспомогательные библиотеки и т.п.), проектирует архитектуру программного средства, выбирает структуры данных, разрабатывает алгоритмы для решения поставленной задачи и исходный текст программы на выбранном языке программирования, отлаживает и выполняет программное средство на компьютере.

Примерный объём исходного текста программы — не менее 5 алгоритмов, реализованных студентом. Разработанное студентом программное средство должно реализовывать достаточное количество функций, чтобы быть полезным для решения поставленной задачи конечным пользователем. Если иное не согласовано с руководителем курсового проектирования, программное средство должно иметь графический пользовательский интерфейс, не должно требовать специальных знаний для подготовки его к работе.

Язык программирования выбирается из числа языков, изучаемых в рамках учебной дисциплины «Конструирование программного обеспечения». Использование других языков программирования допускается по согласованию с руководителем курсового проектирования при разработке вспомогательных компонентов, но не должно превышать 40% от общего объёма проекта.

В пояснительной записке должно присутствовать обоснование выбора языка программирования, среды разработки, вспомогательных библиотек и иного инструментария, использованного при работе над проектом.

Количество часов на выполнение курсового проекта – 40 часов. Зачётных единиц – 1.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ КУРСОВЫХ ПРОЕКТОВ

1. Разработка текстового редактора.
2. Разработка графического редактора.
3. Разработка аудиоредактора.
4. Разработка видеоредактора.
5. Разработка файлового менеджера.
6. Разработка утилиты для анализа исполняемых файлов.
7. Разработка транслятора языка программирования.
8. Разработка компьютерной игры.
9. Разработка интернет-браузера.
10. Разработка сетевого чата.
11. Разработка информационно-справочных каталогов.

12. Разработка программных средств шифрования/архивации файлов.
13. Формирование и обработка трёхмерных графических изображений.
14. Применение численных методов решения математических задач.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

1. Знакомство со средой программирования для языка низкого уровня. Разработка, трансляция, компоновка, выполнение и отладка на примере простой программы.
2. Работа с массивами в языке низкого уровня.
3. Работа со строковыми данными в языке низкого уровня.
4. Использование подпрограмм в языке низкого уровня.
5. Построение графических изображений в программах на языке низкого уровня.
6. Обработка прерываний.
7. Работа с массивами в языке высокого уровня.
8. Работа со строковыми данными в языке высокого уровня.
9. Работа со структурными типами данных в языке высокого уровня.
10. Работа с файлами в программах на языке высокого уровня.
11. Работа с динамическими структурами данных.
12. Построение графических изображений в программах на языке высокого уровня.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ КОМПЬЮТЕРНЫХ ПРОГРАММ (необходимого оборудования, наглядных пособий и т. п.)

1. Ассемблер (например, MASM, TASM, FASM, NASM).
2. Среда программирования (например, Microsoft Visual Studio).
3. Отладчик.
4. ПО для виртуализации (например, VMWare Workstation).