

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учебно-методическое объединение по образованию
в области информатики и радиоэлектроники

УТВЕРЖДАЮ

Первый заместитель Министра образования
Республики Беларусь

_____ А.Г. Баханович

Регистрационный № _____

КОНСТРУИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

**Примерная учебная программа по учебной дисциплине
для специальностей:**

6-05-0612-01 Программная инженерия

**7-07-0713-01 Информационные и управляющие системы физических
установок**

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.Н. Пищов

СОГЛАСОВАНО

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

_____ И.В. Титович

Эксперт-нормоконтролер

Минск 2024

СОСТАВИТЕЛИ:

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 программной инженерии учреждения образования «Белорусский государственный технологический университет» (протокол №9 от 07.03.2024);

А.С.Сидорович, ведущий инженер-программист иностранного производственного унитарного предприятия «ИССОФТ СОЛЮШЕНЗ»

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ В КАЧЕСТВЕ ПРИМЕРНОЙ:

Кафедрой программного обеспечения информационных технологий учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № 9 от 18.12.2023);

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 № 8 от 15.04.2024);

Научно-методическим советом по электронным системам и технологиям Учебно-методического объединения по образованию в области информатики и радиоэлектроники (протокол № 8 от 17.04.2024)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

ХАРАКТЕРИСТИКА УЧЕБНОЙ ДИСЦИПЛИНЫ

Примерная учебная программа по учебной дисциплине «Конструирование программного обеспечения» разработана для студентов учреждений высшего образования, обучающихся по специальностям 6-05-0612-01 «Программная инженерия», 7-07-0713-01 «Информационные и управляющие системы физических установок» в соответствии с требованиями образовательных стандартов общего и специального высшего образования и примерных учебных планов вышеуказанных специальностей. Настоящая программа устанавливает минимальные требования к знаниям и умениям обучающихся вышеуказанной специальности и определяет содержание и виды учебных занятий по учебной дисциплине.

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 обучающимися механизмов выполнения программ, их взаимодействия с операционными системами, аппаратным обеспечением, дополняет знания, полученные ранее в рамках учебной дисциплины «Основы алгоритмизации и программирования», и представления обучающихся о языках программирования, а также принципах проектирования программных средств, парадигмах программирования. Это позволяет обучающемуся 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 обучающимся оценивать качество предлагаемых алгоритмических и архитектурных решений, отдавая предпочтение более эффективным, подходящим более широкому кругу пользователей и т.д.

Воспитательное значение учебной дисциплины «Конструирование программного обеспечения» заключается в формировании у обучающихся математической культуры и научного мировоззрения; развитии исследовательских умений, аналитических способностей, креативности, необходимых для решения научных и практических задач; развитии познавательных способностей и активности: творческой инициативы,

самостоятельности, ответственности и организованности; формировании способностей к саморазвитию, самосовершенствованию и самореализации.

Изучение данной учебной дисциплины способствует созданию условий для формирования интеллектуально развитой личности обучающегося, которой присущи стремление к профессиональному совершенствованию, активному участию в экономической и социально-культурной жизни страны, гражданская ответственность и патриотизм.

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 учетом 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 учетом 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», «Основы программной инженерии» (для специальности 6-05-0612-01 «Программная инженерия»). В свою очередь учебная дисциплина «Конструирование программного обеспечения» является базой для таких учебных дисциплин специальности 6-05-0612-01 «Программная инженерия», как «Системное программирование», «Объектно-ориентированные технологии программирования и стандарты проектирования», а также для таких учебных дисциплин компонента учреждения образования для специальности 7-07-0713-01 «Информационные и управляющие системы физических установок», как «Архитектура компьютерной техники и операционных систем», «Программное обеспечение встроенных систем».

ТРЕБОВАНИЯ К УРОВНЮ ОСВОЕНИЯ СОДЕРЖАНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

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

базовая профессиональная: использовать в разработке программных продуктов формальные методы конструирования программного обеспечения, оценки сложности алгоритмов и их практической реализации;

универсальные для специальности 6-05-0612-01 «Программная инженерия»: владеть основами исследовательской деятельности, осуществлять поиск, анализ и синтез информации;

быть способным к саморазвитию и совершенствованию в профессиональной деятельности;

проявлять инициативу и адаптироваться к изменениям в профессиональной деятельности;

универсальные специальности 7-07-0713-01 «Информационные и управляющие системы физических установок»:

применять методы научного познания в исследовательской деятельности, генерировать и реализовывать инновационные идеи;

быть способным к саморазвитию и самосовершенствованию в профессиональной деятельности, развивать инновационную восприимчивость и способность к инновационной деятельности;

проявлять инициативу и адаптироваться к изменениям в профессиональной деятельности, быть способным к прогнозированию условий реализации профессиональной деятельности и решению задач в условиях неопределенности.

В результате изучения учебной дисциплины обучающийся должен:

знать:

теоретические основы разработки программ с учетом 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.

Примерная учебная программа рассчитана на 210 учебных часов, из них – 120 аудиторных. Примерное распределение аудиторных часов по видам занятий: лекции – 60 часов, лабораторные занятия – 60 часов.

ПРИМЕРНЫЙ ТЕМАТИЧЕСКИЙ ПЛАН

Наименование раздела, темы	Всего аудиторных часов	Лекции	Лабораторные занятия
Раздел 1. Машинно-ориентированный язык программирования	32	16	16
Тема 1. Основные элементы архитектуры процессора	8	4	4
Тема 2. Основы программирования на машинно-ориентированном языке	6	2	4
Тема 3. Организация вычислительных процессов	8	4	4
Тема 4. Обработка данных	6	2	4
Тема 5. Работа с портами и устройствами	2	2	-
Тема 6. Дополнительные возможности языка	2	2	-
Раздел 2. Язык программирования высокого уровня	60	30	30
Тема 7. Основы программирования на языке высокого уровня	4	4	-
Тема 8. Операторы управления вычислительным процессом	4	2	2
Тема 9. Составные типы данных	8	4	4
Тема 10. Функции	8	4	4
Тема 11. Классы хранения и видимость переменных	4	4	-
Тема 12. Работа с файлами	6	2	4
Тема 13. Директивы препроцессора	6	2	4
Тема 14. Модули и компоновка	2	2	-
Тема 15. Неопределенное поведение	6	2	4
Тема 16. Особенности организации структур данных	2	2	-
Тема 17. Парадигмы программирования	10	2	8
Раздел 3. Визуальное программирование	28	14	14
Тема 18. Средства визуального программирования	2	2	-
Тема 19. Формы и пользовательский интерфейс	6	2	4
Тема 20. Компоновка элементов управления	4	2	2
Тема 21. Управляющие элементы	8	4	4
Тема 22. Компоненты внешнего оформления	4	2	2
Тема 23. Пользовательские элементы управления	4	2	2
Итого:	120	60	60

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

РАЗДЕЛ 1. МАШИННО-ОРИЕНТИРОВАННЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ

Тема 1. ОСНОВНЫЕ ЭЛЕМЕНТЫ АРХИТЕКТУРЫ ПРОЦЕССОРА

Режимы работы процессоров. Понятие инструкции. Операнды и их виды. Ограничения при использовании операндов. Классификация регистров, их назначение и основные характеристики. Основные флаги регистра флагов. Принцип работы процессора. Понятия сегмента и смещения. Полный адрес. Физический адрес. Правила использования сегментных регистров. Эффективный адрес. Возможности и ограничения адресации операндов в памяти. Стек. Принцип работы стека.

Тема 2. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА МАШИННО- ОРИЕНТИРОВАННОМ ЯЗЫКЕ

Ассемблер и язык ассемблера. Обзор программного обеспечения для разработки на машинно-ориентированном языке. Принцип работы ассемблера. Машинный код. Основные правила взаимодействия с операционной системой. Понятие точки входа. Простейшая программа. Основные элементы языка ассемблера. Метки. Комментарии. Директивы. Механизм запуска программ.

Тема 3. ОРГАНИЗАЦИЯ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ

Инструкции условного и безусловного перехода. Использование регистра флагов для проверки условий. Специализированные инструкции для проверки условий. Виды меток. Организация циклов. Понятие процедуры. Инструкции для работы с процедурой. Соглашения вызова. Способы передачи параметров. Способы возврата значений. Передача параметров через стек. Понятие дальнего вызова. Отличия дальнего вызова от ближнего. Дальний возврат. Дальний/ближний вызов как элемент соглашения вызова. Разработка программ с использованием дальнего вызова. Понятие прерывания. Обработчик прерывания. Виды прерываний. Таблица векторов прерываний, ее расположение в памяти и формат. Программные прерывания. Механизм вызова обработчика прерывания. Возврат из обработчика прерывания. Аппаратные прерывания. Разрешение и запрет отработки аппаратных прерываний. Перехват прерываний.

Тема 4. ОБРАБОТКА ДАННЫХ

Сложение и вычитание знаковых и беззнаковых чисел. Изменение знака. Умножение и деление знаковых и беззнаковых чисел. Расширение чисел. Инструкции изменения флагов. Побитовые сдвиги. Инструкции для обработки строковых данных (строковые инструкции). Префиксы повторения. Принципы работы с FPU. Поддерживаемые целочисленные и вещественные форматы. Инструкции загрузки, пересылки и выгрузки данных. Арифметические инструкции. Тригонометрические инструкции.

Тема 5. РАБОТА С ПОРТАМИ И УСТРОЙСТВАМИ

Понятие видеоадаптера. Переключение режимов видеоадаптера. Понятие видеопамяти. Запись и чтение видеопамяти. Инструкции для работы с портами ввода-вывода. Примеры взаимодействия с legacy-устройствами через порты ввода-вывода: управление регистрами видеоадаптера, вывод звука.

Тема 6. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ ЯЗЫКА

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. 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. Понятие уязвимости. Понятие уровня привилегий. Понятие security boundary. Обзор уязвимостей, основанных на аппаратных оптимизациях. Проблема analog hole.

РАЗДЕЛ 2. ЯЗЫК ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ

Тема 7. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ

Стандартизация языка высокого уровня и его взаимосвязь с другими языками программирования. Основные типы данных. Синтаксис объявления переменных. Модификаторы. Структура программы. Основные средства ввода-вывода. Простейшая программа на языке высокого уровня. Арифметические операции. Преобразование типов. Поразрядные (побитовые) операции. Операции сдвига. Операции сравнения. Логические операции и их отличия от одноименных поразрядных операций. Приоритеты операций.

Тема 8. ОПЕРАТОРЫ УПРАВЛЕНИЯ ВЫЧИСЛИТЕЛЬНЫМ ПРОЦЕССОМ

Синтаксис записи операторов. Оператор проверки условия. Оператор безусловной передачи управления. Метки. Оператор множественного выбора. Оператор цикла с параметром. Оператор цикла с предусловием. Оператор цикла с постусловием. Выход из цикла и перехода к следующей итерации.

Тема 9. СОСТАВНЫЕ ТИПЫ ДАННЫХ

Массивы. Инициализация массивов. Одномерные и многомерные массивы. Указатели. 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.

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

Тема 10. ФУНКЦИИ

Объявление функции. Опережающее и определяющее описания. Возврат значения из функции. Области видимости элементов программы. Способы передачи параметров. Указатели на функции. Объявление функций с переменным числом параметров. Доступ к параметрам. Синтаксис объявления функции с переменным числом параметров. Рекурсивные функции. Виды рекурсии.

Тема 11. КЛАССЫ ХРАНЕНИЯ И ВИДИМОСТЬ ПЕРЕМЕННЫХ

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

Тема 12. РАБОТА С ФАЙЛАМИ

Понятие файла и файловой системы. Текстовый и бинарный режимы работы. 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 13. 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 14. МОДУЛИ И КОМПОНОВКА

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

Тема 15. НЕОПРЕДЕЛЕННОЕ ПОВЕДЕНИЕ

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

Тема 16. ОСОБЕННОСТИ ОРГАНИЗАЦИИ СТРУКТУР ДАННЫХ

Обзор основных структур данных и способов их реализации. Понятие согласованности данных. Способы обработки ошибок в программах. Обработка исключений. Понятие транзакции. Обеспечение транзакционности при работе со структурами данных.

Тема 17. ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ

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

РАЗДЕЛ 3. ВИЗУАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Тема 18. СРЕДСТВА ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ

Назначение и возможности системы визуального программирования
Состав и назначение инструментов среды. Создание простого приложения.
Создание установочного пакета. Подготовка приложения к развертыванию.

Тема 19. ФОРМЫ И ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

Формы. Состав проекта. Режимы дизайна и кода. Размещение элементов управления на форме. Компоненты, входящие в состав проекта. Изменение внешнего вида и поведения формы. Настройка состояния формы. Свойства проекта. Форматирование элементов управления.

Тема 20. КОМПОНОВКА ЭЛЕМЕНТОВ УПРАВЛЕНИЯ

Контейнерные элементы управления. Коллекции. Свойства элемента управления, определяющие его поведение. Группировка элементов управления. Вкладки. Подразделы формы. Табличные контейнеры. Связывание элементов управления с данными. Использование привязки данных для автоматического обновления пользовательского интерфейса.

Тема 21. УПРАВЛЯЮЩИЕ ЭЛЕМЕНТЫ

Управляющие элементы, обеспечивающие выбор из фиксированного набора альтернатив, предоставленных пользователю. Список и комбинированный список. Обработка и настройка списков. Обработка выбора в списке. Свойства списка. Стиль списка. Флажки и переключатели. Состояние флажков и переключателей. Обработка выбора переключателя. Программное управление переключателем. Рисунки, изображения, рамки. Работа с массивами элементов управления.

Тема 22. КОМПОНЕНТЫ ВНЕШНЕГО ОФОРМЛЕНИЯ

Ползунки, полосы прокрутки. Полоса состояния. Панели общего назначения. Многоуровневое меню. Клавиши быстрого вызова. Обработка команд меню. Контекстное меню. Визуальный компонент «панель инструментов».

Тема 23. ПОЛЬЗОВАТЕЛЬСКИЕ ЭЛЕМЕНТЫ УПРАВЛЕНИЯ

Организация пользовательских элементов управления. Расширение функциональности приложений с помощью элементов управления.

При разработке учебных программ учреждения образования с целью оптимизации образовательного процесса в рамках учебной дисциплины «Конструирование программного обеспечения» для разных форм получения образования допускается перераспределением часов, отведенных на изучение тем одного раздела, между темами другого раздела, при условии достижения основных задач учебной дисциплины.

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. Зубков, С. В. Ассемблер для DOS, Windows и UNIX / С. В. Зубков. – Москва : ДМК Пресс, 2017. – 638 с.
6. Жуков, А. Ассемблер и программная модель процессоров x86/x64 / А. Жуков. – Санкт-Петербург : БХВ-Петербург, 2023. – 400 с.
7. Керниган, Б. Практика программирования / Б. Керниган, Р. Пайк. – Москва : Вильямс, 2017. – 288 с.
8. Керниган, Б. Язык программирования C / Б. Керниган, Д. Ритчи. – Москва : Вильямс, 2017. – 288 с.
9. Прешерн, К. Язык C. Мастерство программирования. Принципы, практики и паттерны / К. Прешерн. – Москва : ДМК Пресс, 2023. – 300 с.
10. Павловская, Т. А. C/C++. Программирование на языке высокого уровня : учебник для вузов / Т. А. Павловская. – Санкт-Петербург : Питер, 2021. – 461 с.
11. Павловская, Т. А. C#. Программирование на языке высокого уровня : учебник для вузов / Т. А. Павловская. – Санкт-Петербург : Питер, 2014. – 432 с.
12. Шарп, Дж. Microsoft Visual C#. Подробное руководство / Дж. Шарп. – 8-е изд. – Санкт-Петербург : Питер, 2017. – 848 с.
13. Стиллмен, Э. Head First. Изучаем C# / Э. Стиллмен, Дж. Грин ; пер. с англ. Е. Матвеева. – 4-е изд. – Санкт-Петербург : Питер, 2022. – 768 с.
14. Албахари, Д. C# 12. Карманный справочник / Д. Албахари, Б. Албахари. – Санкт-Петербург : Вильямс, 2024. – 272 с.

ДОПОЛНИТЕЛЬНАЯ

15. Куссвюрм, Д. Профессиональное программирование на ассемблере x64 с расширениями AVX, AVX2 и AVX-512 / Д. Куссвюрм. – Москва : ДМК Пресс, 2021. – 628 с.
16. Йо, В. Г. Программирование на ассемблере x64 для начинающих / В. Г. Йо. – Москва : ДМК Пресс, 2021. – 332 с.
17. Гриффитс, И. Програмируем на C# 8.0. Разработка приложений / И. Гриффитс. – Санкт-Петербург : Питер, 2021. – 944 с.
18. Васильев, А. Н. Программирование на C# для начинающих. Особенности языка / А. Васильев. – Москва : Эксмо, 2019. – 528 с.
19. Троелсен, Э. Язык программирования C# 9 и платформа .NET 5 : основные принципы и практики программирования / Э. Троелсен, Ф. Джепикс. – 10-е изд. – Санкт-Петербург : Диалектика ; Вильямс, 2022 – 1392 с.

20. Тюгашев, А. А. Визуальное программирование : учебное пособие для вузов / А. А. Тюгашев. – Самара : СамГУПС, 2020. – 147 с.

21. Тюгашев, А. А. Языки программирования : учебное пособие / А. А. Тюгашев. – Санкт-Петербург : Питер, 2021. – 334 с.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

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 ОБУЧАЮЩИХСЯ

Примерными учебными планами по специальностям 6-05-0612-01 «Программная инженерия» и 7-07-0713-01 «Информационные и управляющие системы физических установок» в качестве формы промежуточной аттестации по учебной дисциплине «Конструирование программного обеспечения» рекомендуется зачет, экзамен и курсовой проект.

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

Для текущего контроля по учебной дисциплине и диагностики компетенций обучающихся 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 ходе разработки курсового проекта;

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

РЕКОМЕНДАЦИИ ПО КУРСОВОМУ ПРОЕКТИРОВАНИЮ

В случае планирования выполнения курсового проекта по учебной дисциплине в его подготовке следует руководствоваться следующими рекомендациями.

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 обучающийся 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 объем исходного текста программы — не менее 5 алгоритмов, реализованных обучающимся. Разработанное обучающимся 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 объема проекта.

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

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 трехмерных 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.
13. Разработка приложений с графическим интерфейсом.
14. Создание пользовательских элементов управления.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ КОМПЬЮТЕРНЫХ ПРОГРАММ

(необходимого оборудования, наглядных пособий и т. п.)

1. Ассемблер (например, MASM, TASM, FASM, NASM).
2. Среда программирования с поддержкой языка программирования высокого уровня (например, Microsoft Visual Studio).
3. Отладчик.
4. Программное обеспечение для виртуализации (например, VMWare Workstation).