

**Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Санкт-Петербургский политехнический университет Петра Великого»**

---

УТВЕРЖДАЮ  
Директор ИКНК  
\_\_\_\_\_ Д.П. Зегжда  
«17» июня 2025 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

**«Архитектура компьютера»**

Разработчик	Высшая школа программной инженерии
Направление (специальность) подготовки	09.03.04 Программная инженерия
Наименование ОП	09.03.04_01 Технология разработки и сопровождения качественного программного продукта
Квалификация (степень) выпускника	<b>бакалавр</b>
Образовательный стандарт	<b>СУОС</b>
Форма обучения	<b>Очная</b>

СОГЛАСОВАНО	Соответствует СУОС
Руководитель ОП	Утверждена протоколом заседания
_____ А.В. Петров «01» апреля 2025 г.	высшей школы "ВШПИ" от «01» апреля 2025 г. № 1

РПД разработал:  
Профессор, д.т.н., доц. С.А. Молодяков

# **1. Цели и планируемые результаты изучения дисциплины**

## **Цели освоения дисциплины**

1. Формирование специалистов, умеющих обоснованно и результативно использовать современную вычислительную технику, грамотно оценивать и осваивать новые элементы и системы.
2. Подготовка специалистов, обладающих знаниями, связанными с основными понятиями вычислительной техники, концепциями механизмов выполнения команд, принципами построения компьютеров и систем обработки информации.
3. Подготовка специалистов, обладающих умениями: эффективно программировать на низком уровне все элементы процессоров и построенных на их основе компьютеров; оценивать и разрабатывать наилучшие по производительности программы; программировать мультимедийные системы.

## **Результаты обучения выпускника**

<b>Код</b>	<b>Результат обучения (компетенция) выпускника ООП</b>
<b>ОПК-6</b>	<b>Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов</b>
ИД-6 ОПК-6	Разрабатывает и тестирует программы на уровне ассемблера
ИД-7 ОПК-6	Создает программное обеспечение для обработки аудио и видео информации
<b>ОПК-7</b>	<b>Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой</b>
ИД-1 ОПК-7	Применяет особенности архитектуры компьютера при решении профессиональных задач

## **Планируемые результаты изучения дисциплины**

### **знания:**

- Знает команды языка Ассемблера, коды команд, программную модель процессоров
- Знает принципы построения процессоров и мультипроцессорных систем
- Знает принципы построения и организацию процессоров и компьютеров

**умения:**

- Умеет разрабатывать программы с элементами ассемблера, тестировать выполнение программ на регистрах процессора
- Умеет разрабатывать программы с элементами векторности и многопотковости
- Умеет программировать вычислительные и медиийные системы на низком (регистровом) уровне

**навыки:**

- Владеет методами разработки медиийных приложений
- Владеет навыками разработки приложений с использованием внешних программных библиотек и встраиваемого ассемблера

## **2. Место дисциплины в структуре ООП**

В учебном плане дисциплина «Архитектура компьютера» относится к модулю «Модуль цифровых компетенций».

Изучение дисциплины базируется на результатах освоения следующих дисциплин:

- Алгоритмизация и программирование
- Системное программное обеспечение GNU/Linux

### **3. Распределение трудоёмкости освоения дисциплины по видам учебной работы и формы текущего контроля и промежуточной аттестации**

#### **3.1. Виды учебной работы**

Виды учебной работы	Трудоемкость по семестрам
	Очная форма
Лекционные занятия	60
Практические занятия	38
Самостоятельная работа	100
Промежуточная аттестация (зачет с оценкой)	12
Курсовое проектирование	6
<b>Общая трудоемкость освоения дисциплины</b>	216, ач
	6, зет

#### **3.2. Формы текущего контроля и промежуточной аттестации**

Формы текущего контроля и промежуточной аттестации	Количество по семестрам
	Очная форма
<b>Текущий контроль</b>	
Курсовые работы, шт.	1
<b>Промежуточная аттестация</b>	
Зачеты с оценкой, шт.	2

### **4. Содержание и результаты обучения**

#### **4.1 Разделы дисциплины и виды учебной работы**

№ раздела	Разделы дисциплины, мероприятия текущего контроля	Очная форма		
		Лек, ач	Пр, ач	СР, ач
1.	Базовая организация ЭВМ	4	2	0
2.	Кодирование данных. Системы команд. Ассемблер.	6	10	40

3.	Процессор. Основы построения.	4	2	0
4.	Иерархическая структура памяти. Системная организация ЭВМ	8	2	3
5.	Организация ввода-вывода	6	4	4
6.	<b>СОВРЕМЕННЫЕ ПРОЦЕССОРЫ И МУЛЬТИПРОЦЕССОРНЫЕ СИСТЕМЫ.</b>			
6.1.	Современные процессоры. Параллельность уровня команд. Зависимости структурные, по данным и по управлению	6	2	0
6.2.	Процессорные ядра линейки Pentium.	4	2	4
6.3.	Семейства микропроцессоров. Структурные особенности.	2	0	3
6.4.	Мультипроцессорные архитектуры. Параллельность уровня потоков команд. Программирование.	6	4	12
7.	<b>МЕДИЙНЫЕ И ПЕРИФЕРИЙНЫЕ УСТРОЙСТВА ЭВМ</b>			
7.1.	Классы периферийных устройств. Погрешности. Эффекты дискретизации	4	0	0
7.2.	Звуковой канал. Программирование	2	6	15
7.3.	Системы и процессоры обработки изображений. Видеоканал. Программирование.	4	4	15
7.4.	Магнитные и системы хранения. Принтеры. Интерфейсы периферийных устройств.	4	0	4
<b>Итого по видам учебной работы:</b>		60	38	100
Зачеты с оценкой, ач				0
<b>Часы на контроль, ач</b>				0
<b>Курсовое проектирование</b>				6
<b>Промежуточная аттестация (зачет с оценкой)</b>				12
<b>Общая трудоёмкость освоения: ач / зет</b>				216 / 6

## **4.2. Содержание разделов и результаты изучения дисциплины**

<b>Раздел дисциплины</b>	<b>Содержание</b>
<b>1. Базовая организация ЭВМ</b>	Эволюция развития ЭВМ. Классическая структура ЭВМ - организация Фон-Неймановской машины, ее отличия от современных ЭВМ. Шинная архитектура. Микропроцессоры, как основа построения компьютера. “Закон Мура”. Программная модель процессора и ЭВМ. Регистровые файлы. Типовая структура машинной команды в пространстве и во времени. CISC и RISC идеологии. Методы адресации и организация памяти. Конвейер. Пути повышения производительности. Ассемблер.
<b>2. Кодирование данных. Системы команд. Ассемблер.</b>	Системы счисления. Форматы представления. Выполнение арифметических операций с числами с фиксированной и плавающей запятой. Особые случаи. Представление символьной информации и команды работы. Таблицы кодировки ASCII, Unicode. Представление текстов. Представление графической информации. Система команд процессора Pentium. Ассемблер. MMX-, SSE-, AVX- технологии. Иерархическая структура при построении программ: подпрограммы, макросы. Обращение к ПП и возврат. Использование стека. Требования к командам работы с подпрограммами. Команды вызова прерываний. Понятие контекста
<b>3. Процессор. Основы построения.</b>	Основные компоненты процессора и их назначение. Элементы схемотехники. Структура микрокоманды. Что такое такт процессора. Структура процессора с тремя внутренними шинами. VLIW-, EPIC-, Hyper-threading технологии.

<b>4. Иерархическая структура памяти. Системная организация ЭВМ</b>	<p>Адресная организация памяти. Стековая организация памяти. Ассоциативная память. Иерархия памяти. Основные характеристики разных "слоев" памяти. Организация динамической памяти. Разбиение ОЗУ на банки. Микросхемы динамической памяти. Временные диаграммы обращения к ОЗУ. Регенерация, виды регенерации. Обнаружение и исправление ошибок. Виртуальная память. Принципы организации КЭШ памяти. Многоуровневая КЭШ память. Защищенный режим и страничный механизм. Схема вычисления физического адреса в защищенном режиме. Принципы аппаратного обеспечения защиты задач друг от друга. Уровни привилегий и аппаратная поддержка защиты операционной системы. Аппаратная поддержка страничного механизма. Мультизадачный режим. Диспетчерирование и планирование. Аппаратная поддержка отладки. Внутренние прерывания при отладке. Расширенная отладка.</p>
<b>5. Организация ввода-вывода</b>	<p>Канал (магистраль). Алгоритмы обмена - синхронный и асинхронный. Временная диаграмма. Элементы протокола обмена в PCI. Распределение ресурсов, Plug and Play. Прерывание как способ реакции на асинхронные события. Типы событий, вызывающих прерывания: аппаратные (hardware int), исключения (exceptions). Контроллер прерываний, схема, модель для программиста. Прямой доступ к памяти. Контроллер ПДП. Управление вводом/выводом: проблемы унификации. Системные устройства ввода-вывода. Таймер. Клавиатура. Программные модели последовательного (COM) и параллельного (SPP, EPP) интерфейсов. Загрузка и тестирование ЭВМ.</p>
<b>6. СОВРЕМЕННЫЕ ПРОЦЕССОРЫ И МУЛЬТИПРОЦЕССОРНЫЕ СИСТЕМЫ.</b>	
<b>6.1. Современные процессоры. Параллельность уровня команд. Зависимости структурные, по данным и по управлению</b>	<p>Тестирование быстродействия. Тесты Spec и Linpack. Конвейер. Суперскалярность. Векторность. Зависимости структурные, по данным и управлению. Способы преодоления зависимостей. Статическая и динамическая оптимизация исполнения команд. Предсказание переходов, разворачивание циклов, параллельное выполнение команд из окна выполнения. Переименование регистров. Динамическое исполнение. Методы статического и динамического предсказания переходов. Логические и физические ресурсы микропроцессора. Способы переименования ресурсов. Работа с памятью, спекулятивное обращение к памяти.</p>

<b>6.2. Процессорные ядра линейки Pentium.</b>	Мультискалярные процессоры Pentium. Основные структурные особенности, обеспечивающие повышение производительности. Режимы пониженного энергопотребления.
<b>6.3. Семейства микропроцессоров. Структурные особенности.</b>	Семейства микропроцессоров Motorola PowerPC, IBM Power, Alpha, SPARC. Архитектурные пути повышения производительности и нетрадиционные архитектуры. Процессор Itanium AI-64. Командные связки. Регистровые окна. Механизм предикатных команд и его аппаратная поддержка.
<b>6.4. Мультипроцессорные архитектуры. Параллельность уровня потоков команд. Программирование.</b>	Уровни параллелизма. Гранулярность. Закон Амдала. Закон Густафсона. Классы параллельных архитектур. Классификация Флинна. Мультикомпьютерные и мультипроцессорные вычислительные системы. Матричные и векторные процессоры. Cuda процессоры. PVP Parallel Vector Process - параллельная архитектура с векторными процессорами. Шинная архитектура с однородным доступом для задач симметричной обработки(SMP). Мультипроцессоры с неоднородным доступом (NUMA). Нейронная сеть. Кластерные архитектуры. МРР архитектура (massive parallel processing). Сетевая архитектура. Топологии и метрики сетей. Суперкомпьютеры. Отказоустойчивые вычислительные системы. RAID технологии. Элементы программирования OpenMP и MPI.
<b>7. МЕДИЙНЫЕ И ПЕРИФЕРИЙНЫЕ УСТРОЙСТВА ЭВМ</b>	
<b>7.1. Классы периферийных устройств. Погрешности. Эффекты дискретизации</b>	Классы периферийных устройств. Стандартизация. Погрешности статические, динамические и восстановления. Теорема Котельникова. Преобразование Фурье. Эффекты дискретизации.
<b>7.2. Звуковой канал. Программирование</b>	Классы преобразователей. Возможности подключения датчиков к компьютеру. Базовые схемы АЦП и ЦАП. Sound-blaster. Форматы WAV, MP3. Библиотеки DirectSound, FMOD.
<b>7.3. Системы и процессоры обработки изображений. Видеоканал. Программирование.</b>	Системы технического зрения. Video-blaster. Фотоприемники на приборах с зарядовой связью (ФПЗС). КМОП-фотоприемники. Видеосистема. Видеокамеры. Аппаратные средства обработки изображений. 2D и 3D изображения. Библиотека компьютерного зрения OpenCV.
<b>7.4. Магнитные и системы хранения. Принтеры. Интерфейсы периферийных устройств.</b>	Магнитные диски. Технология записи. Разметка диска. Структура дисковода. Приводы управления считывающей головкой. Интерфейсы винчестеров. Логическое и физическое форматирование. Системы хранения. Принтеры. Методы повышения разрешения. Интерфейсы периферийных устройств.

## **5. Образовательные технологии**

Лекции в сочетании с практическими занятиями. Применение электронных технологий (два дистанционных курса ).

## **6. Лабораторный практикум**

Не предусмотрено

## **7. Практические занятия**

№ раздела	Наименование практических занятий (семинаров)	Трудоемкость, ач
		Очная форма
1.	Введение в низкоуровневое программирование. Встроенный отладчик IDE. Встроенный Ассемблер.	2
2.	Система команд процессора, ее связь с кодами команд.	2
3.	Ассемблер и функции BIOS. Макросы.	2
4.	Способы адресации и сегментная организация памяти	2
5.	Подпрограммы и передача параметров	4
6.	Подпрограммы, программные прерывания и особые случаи	4
7.	FPU. Кодирование чисел с плавающей запятой. Особые численные значения. Особые случаи	2
8.	Отладчик. Отладочные прерывания	2
9.	Обмен ЭВМ с клавиатурой	2
10.	Мультизадачность. Сохранение и восстановление контекста задачи	2
11.	BMP конвертор форматов графических файлов	2
12.	Векторное программирование. Технология MMX-SSE-AVX	2
13.	Программирование для многопоточных платформ. OpenMP	2
14.	3D звук. Библиотека FMod	2
15.	Проигрывание и изменение заголовков звуковых файлов. Библиотека DirectSound	2
16.	Потоковое видео. Библиотека OpenCV	2
17.	Технология CUDA	2
18.	Потоковая обработка отсчетов звукового сигнала в процессоре BlackFin	0
<b>Итого часов</b>		38

## **8. Организация и учебно-методическое обеспечение самостоятельной работы**

Предполагается самостоятельная работа студентов над курсовой работой в первом семестре и над итоговой работой во втором семестре. Кроме того предполагается самостоятельная работа студентов по следующим темам: кодирование команд, современные процессоры, ассемблеры современных процессоров, современные фотоприемники, медийные библиотеки.

Курсовая работа посвящена разработке программ на языке Ассемблер и включает написание двух программ. Темы №1 курсовых работ (первый семестр) :

1.Упорядочить список группы из 10 человек, состоящий из строк "ФАМИЛИЯ И. О." по возрастанию инициала имени. Выведите на экран начальный и по команде конечный список

2.Дан список из 20 слов по 10 символов в каждом. Напечатать его в обратном алфавитном порядке, предварительно удалив из него повторяющиеся слова. При сортировке игнорировать высоту букв (Например, A = a).

3.Разработать набор процедур работы с очередью, а именно: - включение нового элемента; - исключение очередного элемента (со сдвигом очереди). Разработать демо-программу.

4.В файле хранится каталог файлов в формате команды DIR. Разработать программу переупорядочения каталога - по имени, - по расширениям, - по размерам, а также программу выдачи его на экран.

5.Дано арифметическое выражение. Разработать программу проверки правильности по следующим критериям: - допустимые знаки операций ("+", "-", "\*", "/"); - допустимые константы (целые без знака); - допустимые переменные (до пяти латинских букв); - скобки (только круглые, они должны быть парными). Продумать диагностику по разным типам ошибок.

6.Разработать игру: кто придумает больше слов из символов заданной строки. Проверять: - на допустимость по символам; - по словарю, есть ли такие слова. Предусмотреть возможность дополнения словаря и игнорирование высоты букв ( A = a).

7.Разработать программу поиска модели в тексте (см. любой текстовый редактор). С клавиатуры вводим модель поиска.

8.Разработать процедуру слияния двух упорядоченных списков вида "ФАМИЛИЯ И. О." с сохранением алфавитного порядка. Исходные и конечный списки вывести на экран.

9.Разработать процедуру, показывающую в зависимости от указания пользователя либо первые 15, либо последние 15 строк текстового файла. При переключении очищать экран.

10. Организовать печатание произвольного текстового файла на экране. Вертикальный размер окна (количество строк) может меняться. Для печатания желательно использовать PgUP и PgDOWN.

11. Составить "электронную зачетную ведомость" для вашей группы по ТТО:

---

---

N |Фамилия И.О.|Номер зачет. книжки |Отметка о зачете|Дата

п/п| | | |

---

Предусмотреть возможность включения слова "зачет" и даты получения зачета.

12. Составить программу построения диаграммы повторяемости букв в тексте. Формат произвольный. Использовать для текста:

"Fools! You have no perception! The stakes you are gambling Are frighteningly high! We must crush him completely So like John before him This Jesus must die!"

13. Составить программу склонения мужской фамилии в родительный падеж.

14. Составить программу склонения женской фамилии в родительный падеж.

15. Написать программу для преобразования целого числа в римские цифры. Проверить в пределах 1-100.

16. Написать программу, записывающую однобайтовое целое число без знака прописью, например, 128 = сто двадцать восемь

17. В списке людей "ФАМИЛИЯ ИМЯ ОТЧЕСТВО" необходимо найти всех людей с заданным именем. Выводить на экран в виде: ИМЯ ..... количество повторений

18. Из текст выбрать слова, начинающиеся буквой из первой половины алфавита. Образец текста:

The end is just a little harder When brought about by friends! For all who cares this wine could be my blood! For all who cares this bread could be my body! The end! This is my blood you drink, this is my body you eat!

19. Составить программу повторяемости слов в тексте. Использовать для текста:

See my eyes I can hardly see See me stand I can hardly walk I believe you can make me whole  
See my tongue I can hardly talk See my skin I'm a mass of blood See my legs I can hardly stand I believe you can make me well See my purse I'm a poor poor man.

20. Разработать программу, которая выводит строки текста в указанном с клавиатуры порядке.

Темы №2 курсовых работ :

1. Программа "Будильник1". Задайте с клавиатуры время относительно текущего времени (например, 2 минуты). Перехватите прерывание клавиатуры и при повторном наборе (нажатии клавиш). Ваша программа сообщает "Ошибка". При наступлении заданного времени выдайте звуковой сигнал с использованием 2-го канала таймера. Звук должен быть многоголосый.
2. Программа "Будильник2". Будильник должен прозвенеть в абсолютное время (11:45). Перехватите прерывание клавиатуры (номер 9) и при повторном наборе (нажатии клавиш). Ваша программа сообщает "Ошибка". При наступлении заданного времени выдайте звуковой сигнал.
3. Программа "Будильник3". Задайте с клавиатуры время относительно текущего времени (например, 2 минуты). На экран выводится время, оставшееся до звонка. При наступлении заданного времени выдайте звуковой сигнал.
4. Программа "Будильник4". Задайте с клавиатуры время относительно текущего времени (например, 2 минуты). Перехватите прерывание клавиатуры (номер 9) и при повторном наборе (нажатии клавиш). Ваша программа сообщает "Ошибка". При наступлении заданного времени выдайте звуковой сигнал. Определите свободные векторы внешних прерываний, выведите на экран их количество.
5. Определите экспериментально время обработки прерывания от канала 0 таймера. Напишите программу, которая перехватывает прерывание таймера и обеспечивает мигание индикаторов клавиатуры (или выдает звуковой сигнал) через каждую 1 сек.
6. Напишите программу для измерения времени выполнения одной тестовой подпрограммы тремя способами с разной точностью (используя ячейку 46С, повышение частоты прерываний системного времени, нулевой канал таймера). Изменение частоты прерываний от канала 0 исказит значение системного времени. Исключите это искажение. Распечатайте содержимое ячейки, используемой для счета импульсов времени 0046С,Д,Е,Ф.
7. Подмена клавиш 1. Замените клавишу А на В. Напишите программу, обработки прерывания клавиатуры, которая заменяет скан-коды клавиш на одно системное прерывание. Замедлите системные часы в 100 раз путем прореживания прерываний таймера.

8. Подмена клавиш 2. Замените клавишу F1 на F2. Напишите программу, обработки прерывания клавиатуры, которая заменяет скан-коды клавиш. При каждом нажатии выдавайте звуковой сигнал с использованием 2-го канала таймера.
9. Подмена клавиш 3. Замените клавишу Del на Ins. Напишите программу, обработки прерывания клавиатуры, которая заменяет скан-коды клавиш. При каждом нажатии выдавайте звуковой сигнал с использованием порта 061h.
10. Работа с клавиатурой 1. Напишите программу, обработки прерывания клавиатуры, которая выводит на экран скан-код и код, который соответствует нажатой клавише из буфера клавиатуры.
11. Работа с клавиатурой 2. Напишите программу работы с портом клавиатуры, который используется для чтения скан-кодов и для управления клавиатурой. Выведите на экран скан-коды нажатой клавиши, выполните функции управления клавиатурой.
12. Отладчик 1. Постройте программу, которая содержит два фрагмента, один из которых играет роль отладчика, а второй имитирует отлаживаемую программу. Используйте режим трассировки (флаг T). По нажатию кнопки отладчик должен распечатывать содержимое регистра и обеспечивать мигание индикаторов клавиатуры (или выдавать звуковой сигнал).
13. Отладчик 2. Постройте программу, которая содержит два фрагмента, один из которых играет роль отладчика, а второй имитирует отлаживаемую программу. Используйте режим трассировки (флаг T). Имитатор должен обеспечивать вывод на экран адрес отлаживаемой программы и звуковой сигнал с использованием 2-го канала таймера.
14. Определите свободные векторы внешних прерываний, выведите их на экран. По окончании выдавайте звуковой сигнал с использованием 2-го канала таймера
15. Мультизадачность 1. Организуйте мультизадачную среду (3 и более задач) с выводом на экран номера задачи . Переключение происходит по нажатию клавиши мышки.
16. Мультизадачность 2. Организуйте мультизадачную среду (3 задачи) с выводом на экран номера задачи. Переключение происходит по алгоритму кругового планирования с выделенными квантами времени (используйте прерывание таймера).
17. Мультизадачность 3. Организуйте мультизадачную среду (3 и более задач) с выводом на экран номера задачи. Переключение происходит по алгоритму адаптивного планирования. Для переключения задач используйте прерывание таймера.
18. Музыкальный инструмент1. Напишите программу, в которой имитируется музыкальный инструмент. При нажатии на кнопки клавиатуры выводится звуковой сигнал, высота которого зависит от нажатой кнопки (используйте 2-й канал таймера).
19. Музыкальный инструмент2. Напишите программу, в которой имитируется музыкальный инструмент. Запоминается последовательность нажатия на кнопки клавиатуры и затем проигрывается звуковой фрагмент. Мелодия зависит от последовательности нажатых кнопок.

20. Напишите любую другую программу с указанными и другими внешними элементами.

Схема программы: Нажата кнопка – узнали скан-код из порта 60 – узнали ASCII-код – перепрограммировали таймер – загрузили по каналу ПДП Sound Blaster – запустили музыкальный фрагмент через Sound Blaster с разной скоростью воспроизведения.

Темы итоговой работой, написанной на СИ с применением функций медийных библиотек (второй семестр):

1. Программа превращения USB-камеры в камеру для задач измерений. Используется графический интерфейс, библиотека DirectShow. Выводятся на экран указанные строки/столбец, указанная часть видео кадров, записывается потоковое видео (avi-файл) и статические/динамические(с указанным интервалом) строки/столбец в текстовом файле.
2. Многофункциональный проигрыватель видео файлов с графическим интерфейсом и с использованием библиотеки DirectShow (со звуком).
3. Захват, воспроизведение и запись видео (используется камера) с функциями zoom и subarray (запись в avi-файл). Библиотека DirectShow или Media Foundation.
4. Программа превращения USB-камеры в камеру для задач измерений. Используется графический интерфейс, библиотека OpenCV. Выводятся на экран указанные строки/столбец. Данные записываются в текстовый файл.
5. Обработка видео (не менее трех операций) в графическом процессоре CUDA с использованием CUDA-функций. Вывод исходной и обработанной картинок.
6. Обработка видео в графическом процессоре CUDA и в нескольких ядрах компьютера с использованием библиотеки OpenCL. Вывод исходной и обработанной картинок.
7. Применение векторного программирования (SIMD) для ускорения работы с изображением (поиск в потоке видео тестового изображения).
8. Воспроизведение видео-файлов с возможностью вывода заданной строки/столбца, режимов zoom и subarray.
10. . Многофункциональный интерфейс с функциями OpenCV.

Описание: C:\Users\Alexander\Pictures\Без имени-1.png

11. Конвертор BMP с графическим интерфейсом. Программа по выбору конвертирует изображения из 24 бит на пиксель в 1, 4, 8, 16, 24 бит на пиксель.

12. Конвертор BMP в другие форматы. Данная программа с графическим интерфейсом конвертирует изображения в различные форматы (такие как JPEG, BMP, PNG и др.), а так же позволяет просмотреть записанные изображения.
13. OpenCV или Media Foundation. Поиск образца в потоке изображений с камеры (путем сравнения контуров). Картинка-образец задается в диалоге.
14. Фильтры в OpenCV или Media Foundation. Программа имеет графический интерфейс, который позволяет: запустить поток видео с веб-камеры, и применять различные фильтры к этому потоку в реальном времени.
15. Игра в OpenCV. Работа осуществляется с веб-камерой. Формируется мячик и платформа. Платформой управляем через веб-камеру. При контакте с границами игрового поля или с платформой, управляемой игроком, мячик меняет траекторию передвижения и отскакивает, в зависимости от места столкновения. При столкновении с блоками он так же отскакивает, но при этом блок уничтожается, а игрок получает очко. Задача игрока набрать как можно большее количество очков за наименьшее время.
16. Игра в OpenCV. Ловим мячики.
17. Рисование в OpenCV. Считывание жестов пользователя с веб-камеры и рисование соответствующих объектов в другом рабочем окне. Необходима возможность редактирования кисти, с помощью которой осуществляется рисование.
16. Сравнение в OpenCV. Сравнение изображений и генерация картинки отличий.
17. Наложение (совмещение) изображений (второе прозрачное). Оценка смещения и изменения изображения в выделенной области.
18. Мультимедиа плеер с графическим интерфейсом в DirectShow. Программа проигрывает файл с расширением .avi. Пользователь может выбрать файл для воспроизведения и остановить или продолжить показ видео.
19. Запись движения в OpenCV. Программа считывает и отображает данные с камеры. Если программа находит отличия между текущим и предыдущим кадром, то она пишет видео в avi файл.
20. Проигрыватель FMOD с графическим интерфейсом. Программа воспроизводит mp3 файл и дает возможность поставить воспроизведение на паузу, возобновить, и изменить громкость звука.

21. 3D проигрыватель в FMOD с графическим интерфейсом. Программа позволяет не только воспроизводит mp3 и др. файл в 3D-режиме, но и задать параметры 3D-режима, и выбрать функции (эхо, реверберация...).

22. Многофункциональный проигрыватель DirectSound с графическим интерфейсом.

## Примерное распределение времени самостоятельной работы студентов

Вид самостоятельной работы	Примерная трудоемкость, ач
	Очная форма
<b>Текущая СР</b>	
работа с лекционным материалом, с учебной литературой	2
опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	0
самостоятельное изучение разделов дисциплины	0
выполнение домашних заданий, домашних контрольных работ	6
подготовка к лабораторным работам, к практическим и семинарским занятиям	6
подготовка к контрольным работам, коллоквиумам	12
<b>Итого текущей СР:</b>	26
<b>Творческая проблемно-ориентированная СР</b>	
выполнение расчётно-графических работ	0
выполнение курсового проекта или курсовой работы	34
поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме	0
работа над междисциплинарным проектом	0
исследовательская работа, участие в конференциях, семинарах, олимпиадах	0
анализ данных по заданной теме, выполнение расчётов, составление схем и моделей на основе собранных данных	0
выполнение итоговой лабораторной работы	40
<b>Итого творческой СР:</b>	74
<b>Общая трудоемкость СР:</b>	100

## 9. Учебно-методическое обеспечение дисциплины

### 9.1. Адрес сайта курса

<https://dl.spbstu.ru/course/view.php?id=188>

## **9.2. Рекомендуемая литература**

### **Основная литература**

<b>№</b>	<b>Автор, название, место издания, издательство, год (годы) издания</b>	<b>Год изд.</b>	<b>Источник</b>
1	Молодяков С.А. Архитектура ЭВМ, 2019. URL: <a href="http://elib.spbstu.ru/dl/2/s19-116.pdf">http://elib.spbstu.ru/dl/2/s19-116.pdf</a>	2019	ЭБ СПбПУ
2	Молодяков С.А. Архитектура ЭВМ, 2019. URL: <a href="http://elib.spbstu.ru/dl/2/s19-117.pdf">http://elib.spbstu.ru/dl/2/s19-117.pdf</a>	2019	ЭБ СПбПУ

### **Дополнительная литература**

<b>№</b>	<b>Автор, название, место издания, издательство, год (годы) издания</b>	<b>Год изд.</b>	<b>Источник</b>
1	Дао Л., Беляев М.Б., Гельман М.М. Программирование микропроцессора 8088: Москва: Мир, 1988.	1988	ИБК СПбПУ
2	Молодяков С.А. Архитектура компьютера. Мультимедийное программирование, 2024. URL: <a href="https://elib.spbstu.ru/dl/5/tr/2024/tr24-68.pdf">https://elib.spbstu.ru/dl/5/tr/2024/tr24-68.pdf</a>	2024	ЭБ СПбПУ
3	Молодяков С.А., Петров А.В. Архитектура ЭВМ. Низкоуровневое программирование: лабораторный практикум: Санкт-Петербург: ПОЛИТЕХ-ПРЕСС, 2019. URL: <a href="http://elib.spbstu.ru/dl/2/i19-273.pdf">http://elib.spbstu.ru/dl/2/i19-273.pdf</a>	2019	ЭБ СПбПУ

### **Ресурсы Интернета**

1. Дистанционный курс: <https://dl.spbstu.ru/course/view.php?id=188>
2. Дистанционный курс: <https://dl.spbstu.ru/course/view.php?id=167>

## **9.3. Технические средства обеспечения дисциплины**

Используются шаблоны программ лабораторных заданий. В 1-м семестре в компьютерном классе инструментом исследования ЭВМ являются языки Ассемблер, Си. Во 2-м семестре в классе технических средств используются: Ассемблер и Си для сигнальных процессоров, Visual Studio с медийными библиотеками, видеокамеры, средства функционирования и исследования лабораторных установок.

## **10. Материально-техническое обеспечение дисциплины**

При выполнении лабораторных работ используется компьютерный класс персональных ЭВМ (первый семестр) и класс технических средств (второй семестр). Класс технических средств имеет не менее 10 рабочих мест, которые оборудованы компьютером, исследуемой и измерительной аппаратурой (отладочными модулями сигнальных процессоров, видеокамерами и др.).

## **11. Критерии оценивания и оценочные средства**

### **11.1. Критерии оценивания**

Для дисциплины «Архитектура компьютера» формой аттестации является зачёт с оценкой. Дисциплина реализуется с применением системы индивидуальных достижений.

#### **Текущий контроль успеваемости**

Максимальное значение персонального суммарного результата обучения (ПСРО) по приведенной шкале - 100 баллов

Максимальное количество баллов приведенной шкалы по результатам прохождения двух точек контроля - 80 баллов.

Подробное описание правил проведения текущего контроля с указанием баллов по каждому контрольному мероприятию и критериев выставления оценки размещается в СДО в навигационном курсе дисциплины.

#### **Промежуточная аттестация по дисциплине**

Максимальное количество баллов по результатам проведения аттестационного испытания в период промежуточной аттестации – 20 баллов приведенной шкалы.

Промежуточная аттестация по дисциплине проводится в соответствии с расписанием.

Экзаменационная оценка выводится с учетом успешности сдачи экзамена, всех лабораторных работ и курсовой работы. Зачет с оценкой выводится также только при выполнении всех лабораторных работ и индивидуальных бесед по темам лекций.

При дистанционном обучении также как и при очном обучении проводится экзамен (зачет с оценкой) в форме бесед со студентами по представленным студентам экзаменационным вопросам (билетам).

Результаты промежуточной аттестации, определяются на основе баллов, набранных в рамках применения, СИД

Баллы по приведенной шкале в рамках применения СИД (ПСРО+ ПА)	Оценка по результатам промежуточной аттестации
	Экзамен/диф.зачет/зачет
0 - 60 баллов	Неудовлетворительно/не зачтено

<b>Баллы по приведенной шкале в рамках применения СИД (ПСРО+ ПА)</b>	<b>Оценка по результатам промежуточной аттестации</b>
	<b>Экзамен/диф.зачет/зачет</b>
61 - 75 баллов	Удовлетворительно/зачтено
76 - 89 баллов	Хорошо/зачтено
90 и более	Отлично/зачтено

## **11.2. Оценочные средства**

Оценочные средства по дисциплине представлены в фонде оценочных средств, который является неотъемлемой частью основной образовательной программы и размещается в электронной информационно-образовательной среде СПбПУ на портале etk.spbstu.ru

## **12. Методические рекомендации по организации изучения дисциплины**

Успешное изучение курса требует посещения лекций, активной работы на лабораторных работах, выполнения всех учебных заданий преподавателя, ознакомления с основной и дополнительной литературой. Полезным для студента является ознакомление с материалами порталов дистанционных технологий обучения.

Лабораторные работы составляют важную часть профессиональной подготовки студентов. Они направлены на подтверждение теоретических положений и формирование учебных и профессиональных практических умений. Значительное внимание следует уделять выполнению студентами самостоятельной работы, прежде всего итоговых индивидуальных заданий: в первом семестре - это курсовая работы по теме "разработка программы на языке Ассемблера", а во втором семестре - это индивидуальное задание "разработка медийного приложения с элементами графического интерфейса".

При подготовке к экзамену в дополнение к изучению конспектов лекций и учебных пособий, необходимо пользоваться учебной литературой, рекомендованной в настоящей программе.

При очно-заочном обучении большое внимание следует уделять самостоятельной работе студентов. Тесты, размещенные на сайте, заставляют студентов более глубоко изучить материал.

## **13. Адаптация рабочей программы для лиц с ОВЗ**

Адаптированная программа разрабатывается при наличии заявления со стороны обучающегося (родителей, законных представителей) и медицинских показаний (рекомендациями психолога-

медицинской комиссии). Для инвалидов адаптированная образовательная программа разрабатывается в соответствии с индивидуальной программой реабилитации.