

**Программа учебной дисциплины  
«Архитектура вычислительных систем»**

Утверждена  
Академическим советом ОП  
Протокол № \_\_\_\_\_ от \_\_\_\_\_ . \_\_\_\_\_ .20 \_\_\_\_\_

Разработчик	Баканов Валерий Михайлович, Профессор, Департамент программной инженерии
Число кредитов	4
Контактная работа (час.)	420
Самостоятельная работа (час.)	153
Курс, Образовательная программа	2 (Б) курс, Программная инженерия
Формат изучения дисциплины	Без использования онлайн курса

**1. Цель, результаты освоения дисциплины и пререквизиты**

Цели:

1. Формирование у студентов профессиональных компетенций, связанных с использованием теоретических знаний в области архитектур вычислительных систем
2. Ознакомление студентов с прогрессивными парадигмами развития архитектур вычислителей с целью заложить основы для последующих курсов, посвящённых созданию современных информационных систем
3. Получение практических навыков в области выбора архитектуры вычислительной системы, наилучшим образом раскрывающего потенциальные возможности заданного алгоритма с учётом заданных требований к программному обеспечению
4. Развитие умений, основанных на полученных теоретических знаниях, позволяющих на творческом и репродуктивном уровне применять и создавать эффективные алгоритмы для решения задач обработки информации применительно к данной архитектуре вычислительной системы
5. Получение студентами навыков самостоятельной исследовательской работы, предполагающей изучение специфических методов анализа архитектур вычислительных систем и функционирующего на них программного обеспечения, инструментов и средств, необходимых для реше

Планируемые результаты обучения (ПРО):

1. Разработка алгоритма решения полученной задачи с учетом использования программно-аппаратных возможностей микропроцессоров Intel т языка программирования Ассемблер, программирование и отладка программы на языке Ассемблер, оформление работы для сдачи на про
2. Экзамен по дисциплине: студент должен продемонстрировать профессиональные компетенции, связанные с использованием теоретических знаний в области архитектур вычислительных систем; студент должен иметь навыки самостоятельной исследовательской работы, п

Пререквизиты:

1. Дискретная математика
2. Программирование

**2. Содержание учебной дисциплины**

Тема (раздел дисциплины)	Объем	Планируемые результаты обучения (ПРО),	Формы
--------------------------	-------	--	-------

	<b>в</b>		<b>подлежащие контролю</b>	<b>контроля</b>
	<b>часах</b>			
	лк			
	см			
	onl/cp			
1. История машинного счёта.	2		<ul style="list-style-type: none"> <li>• Экзамен по дисциплине: студент должен продемонстрировать профессиональные компетенции, связанные с использованием теоретических знаний в области архитектур вычислительных систем; студент должен иметь навыки самостоятельной исследовательской работы, п</li> </ul>	
	2			
	6			
2. Определения понятия "Архитектура" в применении к вычислительным системам.	2		<ul style="list-style-type: none"> <li>• Экзамен по дисциплине: студент должен продемонстрировать профессиональные компетенции, связанные с использованием теоретических знаний в области архитектур вычислительных систем; студент должен иметь навыки самостоятельной исследовательской работы, п</li> </ul>	
	2			
	4			
3. Уровни управления процессом вычислений. Архитектура современных микропроцессоров. Конвейерность. Вычислители в позиционной системе счисления с основанием 3. Стековая машина.	0		<ul style="list-style-type: none"> <li>• Экзамен по дисциплине: студент должен продемонстрировать профессиональные компетенции, связанные с использованием теоретических знаний в области архитектур вычислительных систем; студент должен иметь навыки самостоятельной исследовательской работы, п</li> </ul>	
	2			
	6			
4. Общие требования к программному коду. Поточковые (DATA-FLOW) вычислители. Язык программирования Ассемблер.	2		<ul style="list-style-type: none"> <li>• Экзамен по дисциплине: студент должен продемонстрировать профессиональные компетенции, связанные с использованием теоретических знаний в области архитектур вычислительных систем; студент должен иметь навыки самостоятельной исследовательской работы, п</li> </ul>	
	2			
	8			
5. Недостаток процесса вычислений в позиционной системе счисления и альтернативные решения. Простейшие программы на Ассемблере.	2		<ul style="list-style-type: none"> <li>• Экзамен по дисциплине: студент должен продемонстрировать профессиональные компетенции, связанные с использованием теоретических знаний в области архитектур вычислительных систем; студент должен иметь навыки самостоятельной исследовательской работы, п</li> </ul>	
	2			
	8			

6. Архитектура параллельных вычислительных систем. Ассемблер - целочисленная арифметика.	2	<ul style="list-style-type: none"> <li>• Экзамен по дисциплине: студент должен продемонстрировать профессиональные компетенции, связанные с использованием теоретических знаний в области архитектур вычислительных систем; студент должен иметь навыки самостоятельной исследовательской работы, п</li> </ul>	
	2		
	8		
7. Суперкомпьютеры. Ассемблер - организация условных и безусловных переходов.	2	<ul style="list-style-type: none"> <li>• Экзамен по дисциплине: студент должен продемонстрировать профессиональные компетенции, связанные с использованием теоретических знаний в области архитектур вычислительных систем; студент должен иметь навыки самостоятельной исследовательской работы, п</li> </ul>	
	2		
	8		
8. Нейронные сети и нейрокомпьютеры.	2	<ul style="list-style-type: none"> <li>• Экзамен по дисциплине: студент должен продемонстрировать профессиональные компетенции, связанные с использованием теоретических знаний в области архитектур вычислительных систем; студент должен иметь навыки самостоятельной исследовательской работы, п</li> </ul>	
	2		
	6		
9. Ассемблер в архитектуре X86 - практика разработки программ. Интерактивные среды разработки. Макроассемблер.	2	<ul style="list-style-type: none"> <li>• Разработка алгоритма решения полученной задачи с учетом использования программно-аппаратных возможностей микропроцессоров Intel т языка программирования Ассемблер, программирование и отладка программы на языке Ассемблер, оформление работы для сдачи на про</li> <li>• Экзамен по дисциплине: студент должен продемонстрировать профессиональные компетенции, связанные с использованием теоретических знаний в области архитектур вычислительных систем; студент должен иметь навыки самостоятельной исследовательской работы, п</li> </ul>	
	2		
	8		
10. Транспьютеры. Отладка программ на языке Ассемблер.	2	<ul style="list-style-type: none"> <li>• Экзамен по дисциплине: студент должен продемонстрировать профессиональные компетенции, связанные с использованием теоретических знаний в области архитектур вычислительных систем; студент должен иметь навыки</li> </ul>	
	2		

	6	самостоятельной исследовательской работы, п	
11. Метакомпьютинг и концепция GRID. Использование ассемблерных программ при программировании на языках высокого уровня.	2	<ul style="list-style-type: none"> <li>• Экзамен по дисциплине: студент должен продемонстрировать профессиональные компетенции, связанные с использованием теоретических знаний в области архитектур вычислительных систем; студент должен иметь навыки самостоятельной исследовательской работы, п</li> </ul>	
	2		
	6		
12. Архитектура GPU фирмы NVIDIA и технология CUDA.	2	<ul style="list-style-type: none"> <li>• Экзамен по дисциплине: студент должен продемонстрировать профессиональные компетенции, связанные с использованием теоретических знаний в области архитектур вычислительных систем; студент должен иметь навыки самостоятельной исследовательской работы, п</li> </ul>	
	2		
	4		
13. Аналоговые вычислительные системы.	2	<ul style="list-style-type: none"> <li>• Экзамен по дисциплине: студент должен продемонстрировать профессиональные компетенции, связанные с использованием теоретических знаний в области архитектур вычислительных систем; студент должен иметь навыки самостоятельной исследовательской работы, п</li> </ul>	
	2		
	4		
14. Вычислители с программируемой архитектурой, пути усовершенствования вычислительных архитектур.	2	<ul style="list-style-type: none"> <li>• Экзамен по дисциплине: студент должен продемонстрировать профессиональные компетенции, связанные с использованием теоретических знаний в области архитектур вычислительных систем; студент должен иметь навыки самостоятельной исследовательской работы, п</li> </ul>	
	2		
	4		
15. Квантовые вычислители и системы передачи данных.	1	<ul style="list-style-type: none"> <li>• Экзамен по дисциплине: студент должен продемонстрировать профессиональные компетенции, связанные с использованием теоретических знаний в области архитектур вычислительных систем; студент должен иметь навыки самостоятельной исследовательской работы, п</li> </ul>	
	1		
	4		
16. Биологические компьютеры.	1	<ul style="list-style-type: none"> <li>• Экзамен по дисциплине: студент должен продемонстрировать профессиональные компетенции,</li> </ul>	

	1	связанные с использованием теоретических знаний в области архитектур вычислительных систем; студент должен иметь навыки самостоятельной исследовательской работы, п
	4	
<b>Часов по видам учебных занятий:</b>	28	
	30	
	94	
<b>Итого часов:</b>	152	

### *Содержание разделов дисциплины:*

#### **1. История машинного счёта.**

Основные этапы развития автоматизированных вычислений (этапы - пальцевый счет, абак, палочки Непера, машины Паскаля, Лейбница, идея Лейбница о применении двоичной системы счисления, машина Беббеджа, машина Тьюринга, первые релейные машины, ламповые вычислители, транзисторная техника, квантовые вычислители, развитие программных средств).

#### **2. Определения понятия "Архитектура" в применении к вычислительным системам.**

Необходимость понятия "архитектура" в применении к сложным техническим системам. Вариации толкования понятия "архитектура". Первые вычислительные системы, к которым понятие "архитектура" обязательно в применении. Отдельные примеры применения понятия "архитектура". Внутренняя архитектура современных микропроцессоров. Классическая Принстонская архитектура (архитектура фон Неймана). Гарвардская архитектура, области применения. Многоядерность, закон Релэя. Кеширование, связанные с кешированием проблемы безопасности. Архитектура суперкомпьютеров. Принципы и системы классификации вычислительных систем. 4 уровня параллелизации при вычислениях.

#### **3. Уровни управления процессом вычислений. Архитектура современных микропроцессоров. Конвейерность. Вычислители в позиционной системе счисления с основанием 3. Стековая машина.**

Уровень языков сверхвысокого/высокого уровня, уровень машинных команд (инструкций) уровень микрокоманд. Программно/аппаратные средства конвертации между уровнями. Особенности машинных команд процессоров ARM, ITANIUM, ЭЛЬБРУС. Понятие машинного такта (тика) процессора. Суперскалярность. Конвейерная архитектура и Сергей Лебедев. Идеологемы CISC и RISC принципов построения вычислителей. Размерность и точность представления данных, стандарт IEEE 754. Николай Брусенцов и стековая машина на основе трехзначной логики СЕТУНЬ.

#### **4. Общие требования к программному коду. Поточковые (DATA-FLOW) вычислители. Язык программирования Ассемблер.**

Общие принципы определения последовательности выполнения машинных команд. Принципы "Control-flow" и "Data-flow". Ограничения идеологемы "Control-flow", современные технологические проблемы "Data-Flow". Необходимость использования ассоциативной памяти в "Data-flow" вычислительных системах. Статическая и динамическая архитектуры "Data-Flow", проблемы условного выполнения в "Data-Flow". Современное применение "Data-Flow". Язык программирования Ассемблер - применение, достоинства и недостатки. Компиляторы с Ассемблера. Форматы команд Ассемблеры, типы данных, виды адресации. Организация аппаратного стека.

#### **5. Недостаток процесса вычислений в позиционной системе счисления и альтернативные решения. Простейшие программы на Ассемблере.**

Принципиальная невозможность отойти от частично-последовательного характера операций над битами при вычислениях в позиционных системах счисления. Достоинства и ограничения табличных систем. Непозиционные системы счисления. Система остаточных классов (СОК) и ее применение при разработке процессоров. Место СОК в иерархии уровней параллелизации вычислений. Израиль Акушский, технические параметры построенных на СОК вычислителей. Современное состояние СОК при разработке процессоров. Простейшие программы на

Ассемблере - импорт процедур из системных библиотек, работа командной строкой, форматный вывод данных.

## 6. **Архитектура параллельных вычислительных систем. Ассемблер - целочисленная арифметика.**

SMP и MPP архитектуры, их достоинства и недостатки. Причина повышенной масштабируемости MPP. Разница в средствах программирования для SMP и MPP.

Арифметические инструкции в Ассемблере. Использование внутренних регистров процессора при реализации целочисленной арифметики. Регистры состояния (PSW) для индикации особых состояний при выполнении целочисленных команд.

## 7. **Суперкомпьютеры. Ассемблер - организация условных и безусловных переходов.**

тестирования производительности суперкомпьютеров, Джек Донгарра, тест LinPACK.

Альтернативные тесты производительности суперкомпьютеров. Потеря точности при супервычислениях. Технологии выявления параллелизма. ДНФ. Абстрактные модели параллельных вычислений, закон Брента. Двухстадийный процесс реализации условных переходов, использование PSW. "Короткие" и "длинные" переходы. Альтернативные способы реализации условного выполнения - предикация (архитектуры ARM, ITANIUM).

## 8. **Нейронные сети и нейрокомпьютеры.**

История - перцептрон. Формальная связь со структурами мозга (биологическими нейронами).

Функция активации, методика обучения при распознавании образов. Марвин Минский и "ледниковый период" в истории развития нейронных сетей. "Глубокие нейронные сети" (DNN).

"Проклятие размерности" при обучении нейронных сетей, сверточные сети. Нейронные сети с прямым распространением информации и с обратными связями. Методы задания информации для нейронных сетей. Современные аппаратные реализации нейронных сетей.

## 9. **Ассемблер в архитектуре X86 - практика разработки программ. Интерактивные среды разработки. Макроассемблер.**

Процессор x86 фирмы Intel, этапы его развития. Ассемблер для 32-х и 64-бит архитектур.

Ассемблер для процессоров архитектуры ARM. Поддержка операций над числами с плавающей точкой. Арифметический сопроцессор, кольцевой регистр при операциях с плавающей точкой, extended-формат обрабатываемых чисел. Приемы вычислений с плавающей точкой.

Использование макросов при работе с Ассемблером. Понятие пролога и эпилога. Стандартные соглашения о передаче параметров процедурам (stdcall, cdecl,fastcall). Двоично-десятичная арифметика (BCD) в процессорах x86. Инструкции для работы с цепочками (sequence) элементов (одномерными массивами).

## 10. **Транспьютеры. Отладка программ на языке Ассемблер.**

Транспьютеры как первая попытка реализации распределенных вычислений. Сэр Энтони Хоар, концепция взаимодействующих последовательных процессов CSP (Communicating Sequential Process), язык программирования ОККАМ. Язык Z спецификаций. Логика Хоара (Hoare Logic) - научная основа для конструирования корректных программ, используется для определения и разработки языков программирования. Отладчики (Debuggers) программ на Ассемблере.

## 11. **Метакомпьютинг и концепция GRID. Использование ассемблерных программ при программировании на языках высокого уровня.**

Архитектура систем поиска информации в сети InterNet. Создатели "всемирной паутины" Роберт Кайо и сэр Тимоти Джон Бернес-Ли. Закон Зипфа. Современные системы анализа и поиска данных. Эдвард Пейдж и Сергей Брин как авторы реализации поиска данным с учетом пользовательских предпочтений. Ричард Столлман - рупор концепции свободного программного обеспечения. Причины встраивания кодов Ассемблера в программы на языках высокого уровня. Программный реинжиниринг с использованием Ассемблера.

## 12. **Архитектура GPU фирмы NVIDIA и технология CUDA.**

Функции видеокарт как пример использования подхода SIMD (обработка множества данных единой последовательностью команд). История расширения функций видеокарт в сторону арифметических сопроцессоров. Фирма nVidia и аппаратно-программная технология CUDA. Основные параметры GPU, понятие вычислительных возможностей (Compute Capability). Принципы и примеры программирования GPU. Применение GPU при супервычислениях. Графические сопроцессоры фирмы AMD.

## 13. **Аналоговые вычислительные системы.**

Аналоговый и цифровой принципы представления информации. Область эффективности применения аналоговых вычислительных систем. "Рабочие тела" аналоговых вычислительных

систем - жидкости, газы, электричество. Примеры выполнения арифметических операций с помощью аналоговых вычислительных систем. Аналоговые вычислители в военном деле.

#### 14. **14. Вычислители с программируемой архитектурой, пути усовершенствования вычислительных архитектур.**

Принцип реконфигурации вычислительной системы. Изменчивость структуры вычислительной системы против статической системы. Реконфигурируемые вычислительные системы, программируемые логические интегральные схемы (ПЛИС). Этапы процесса подготовки к обработке информации по заданному алгоритму и собственно вычислениям. Эффективность вычислителей с программируемой архитектурой. Систематические матрицы (массивы). Технология сенсорных сетей ZigBee. Архитектура с тегированной памятью. Самоопределяемые данные. Реализация одноуровневой памяти.

#### 15. **15. Квантовые вычислители и системы передачи данных.**

Целесообразность построения квантовых вычислителей. Предел Ландауэра для необратимых вычислений. Проблема наблюдателя в квантовой теории. Кубит - аналог бита в традиционных вычислителях. Особенности кубита, принцип суперпозиции. Физическая связанность ("запутанность") состояний кубитов. Унитарные преобразования в квантовых схемах. Эффективность квантовых компьютеров на известных алгоритмах. Алгоритмы Шора, Гровера, Залки-Визнера, Дойча-Джоза. Будущее современных систем криптозащиты при реализации квантовых компьютеров. Квантовая криптография. Квантовая линия связи. "Квантовые деньги".

#### 16. **16. Биологические компьютеры.**

Физические параметры биологических объектов, позволяющие реализовывать хранение и обработку информации. Запись, считывание и обработка данных. "Инфузорное программирование". Клеточное программирование. Нейронные микросхемы - сращивание традиционных технологий с биологическими. "Живой микрочип". Жизнестойкость биокомпьютеров. Способность биокомпьютеров к копированию и усложнению.

### 3. **Оценивание**

- **ДР**, Не блокирующее, Домашнее задание  
Домашнее задание включает разработку, кодирование, тестирование и отладку программ реализации задачи на языке Ассемблера (по индивидуальному заданию), исследование и сравнительный анализ алгоритмов ее решения. По домашнему заданию оформляется отчет в электронном виде.
- **Экз**, Блокирующее, Экзамен (устный)  
Итоговый контроль: экзамен в конце 2-го модуля (2 час). Экзамен проводится в устной форме.

**Формула округления:** Стандартное арифметическое округление

**Вид формулы оценивания:** Линейная

**Формула оценивания:**

Окончательная оценка = Округление( $1/2 * ДР + 1/2 * Экз$ ) // Окончательная оценка - окончательная оценка за дисциплину, ДР - оценка за домашнюю работу, Экз - оценка на экзамене

### 4. **Примеры оценочных средств**

Тестовый пример - вычислений НОД двух чисел методом Эвклида с вычитанием

```
; Входные данные – файл GCD_EUCLID_32.ASM
;
format PE GUI 4.0 ; GUI-приложение для Windows (EXE-файл)
;
include "c:/emu8086/fasm/include/win32ax.inc" ; путь к подкаталогу /include/
;
.data ; секция данных
formats db "НОД чисел %d и %d суть %d (десятичные)",0
```

```

result db 256 dup(?) ; строка для форматного вывода
;
A dd 81d ; исходные данные (1-е число для нахождения НОД - Наиб. Общего Делителя)
B dd 54d ; 2-е число для того же
;
.code ; начало секция кода
start: ; точка старта программы
;
mov eax,dword[A] ; исходные данные присваиваем регистрам
mov ebx,dword[B]
;
call GCD_2S_Euclid_32 ; вызов процедуры GCD_2S_Euclid_32 (поиск НОД двух чисел)
;
invoke wsprintf,result,formats, [A],[B],eax ; форматирование строки для вывода
add esp,20 ; восстановим стек (соглашение о вызовах cdecl)
;
invoke MessageBox,0,result,"Данные расчёта",MB_OK ; EAX invoke ExitProcess,0 ; возврат в Windows
с кодом возврата 0
;
;
;
proc GCD_2S_Euclid_32 ; USES EAX,EBX; RESULT IN EAX
; фактически используется fastcall-соглашение (передача формальных
; параметров не посредством стека, а через регистры процессора
; ищет GCD от EAX,EBX по Эвклиду (при EAX==0 || EBX==0 возвращается 0)
; GCD is abbreviation of 'Greatest Common Divisor'
; эта процедура изменяет только регистры EAX и EBX
;
; while(EAX!=0 && EBX!=0) { if (EAX>EBX) EAX-=EBX else EBX-=EAX; } return EAX+EBX;
;
cmp eax,0 ; сравниваем EAX с 0
jne ax_jne ; переход, если EAX!=0
ret ; возвращаем EAX=0 как результат
ax_jne: ; вариант EAX!=0
; cmp eax,0 ; сравниваем EAX с 0 (повторять сравнение не надо)
jge ax_jge ; переход, если EAX>=0
neg eax ; изменили знак EAX (получили модуль EAX)
ax_jge: ; вариант EAX>=0
;
cmp ebx,0 ; сравниваем EBX с 0
jne bx_jne ; переход, если EAX!=0
mov eax,ebx ; готовим возврат EAX ret ; возвращаем EAX=0 (result)
bx_jne: ; вариант EBX!=0
; cmp ebx,0 ; сравниваем EBX с 0 (повторять сравнение не надо)
jge bx_jge ; переход, если EBX>=0
neg ebx ; изменили знак EBX (получили модуль EBX)
bx_jge: ; вариант EAX>=0
;
cycle: ; начинаем основной цикл метода Эвклида ;
cmp eax,ebx ; сравним EAX и EBX
je ended ; при EAX==EBX (ZF==1) выход с возвратом EAX
;
jl metka ; при EAX sub eax,ebx ; EAX jmp cycle ; безусловный переход на cycle
metka:
sub ebx,eax ; EBX jmp cycle ; безусловный переход на cycle
ended: ; EAX ;
ret ; возврат из процедуры GCD_2S_Euclid_32
;
endp ; конец процедуры GCD_2S_Euclid_32

```



.end start ; конец программы

Выходные данные

При входных данных EAX=81(10) и EBX=54(10) после выполнения процедуры GCD\_Euclid\_32 правильный ответ EAX=1B(16)=27(10)

Верно выполняется тестовый пример, приведённый выше - 3 балла.

Отсутствуют синтаксические ошибки в заданном примере - 4 балла.

Заданный пример выполняется не на всех входных данных - (+ до 2 баллов) (6-7 баллов).

Заданный пример выполняется на всех входных данных (+ до 2 баллов) (8-9 баллов).

## 5. Ресурсы

### 5.1. Рекомендуемая основная литература

п/п	Наименование
1	Э.Таненбаум, Т.Остин. Архитектура компьютера (издание 6)., — СПб.: Питер, 2014. — 811 с.
2	Боресков А.В., Харламов А.А., Марковский Н.Д. и др. Параллельные вычисления на GPU: архитектура и программная модель CUDA: учеб. пособие для вузов., — М.: Изд-во МГУ, 2012. — 333 с.

### 5.2. Рекомендуемая дополнительная литература

п/п	Наименование
1	Пирогов В.Ю. Ассемблер и дизассемблирование. , — СПб.: БХВ-Петербург, 2007. — 464 с.
2	Барский А.Б. Логические нейронные сети: учеб. пособие., — М.: Интернет-Университет Информационных Технологий: БИНОМ. Лаборатория знаний, 2007. — 352 с.
3	Сандерс, Дж., Кендрот Э. Технология CUDA в примерах: введение в программирование графических процессоров., — М.: ДМК Пресс, 2011. — 231 с.

### 5.3. Программное обеспечение

п/п	Наименование	Условия доступа/скачивания
1	Microsoft Windows 7 Professional RUS Microsoft Windows 8.1 Professional RUS Microsoft Windows 10	Из внутренней сети университета (договор)
2	Microsoft Office Professional Plus 2010	Из внутренней сети университета (договор)
3	FASM URL: <a href="https://flatassembler.net/">https://flatassembler.net/</a>	доступ свободный
4	FasmW URL: <a href="https://flatassembler.net/fasmw17306.zip">https://flatassembler.net/fasmw17306.zip</a>	доступ свободный
5	Fresh IDE URL: <a href="https://fresh.flatassembler.net/">https://fresh.flatassembler.net/</a>	доступ свободный
6	Fasm Editor 2.0 URL: <a href="http://asmworld.ru/instrumenty/fasm-editor-2-0/">http://asmworld.ru/instrumenty/fasm-editor-2-0/</a>	доступ свободный
7	Портал AsmWorld URL: <a href="http://asmworld.ru/">http://asmworld.ru/</a>	доступ свободный
8	NVIDIA: архитектура графических карт и технология CUDA URL: <a href="http://www.nvidia.ru/">http://www.nvidia.ru/</a>	доступ свободный

### 5.4. Профессиональные базы данных, информационные справочные системы, интернет-ресурсы (электронные образовательные ресурсы)

п/п	Наименование	Условия доступа/скачивания

<b>Профессиональные базы данных, информационно-справочные системы</b>		
1	Электронно-библиотечная система Юрайт	URL: <a href="https://biblio-online.ru/">https://biblio-online.ru/</a>
<b>Интернет-ресурсы (электронные образовательные ресурсы)</b>		
1	Открытое образование	URL: <a href="https://openedu.ru/">https://openedu.ru/</a>

#### 5.5. Материально-техническое обеспечение дисциплины

Учебные аудитории для лекционных по дисциплине обеспечивают использование и демонстрацию тематических иллюстраций, соответствующих программе дисциплины в составе:

- ПЭВМ с доступом в Интернет (операционная система, офисные программы, антивирусные программы);

- мультимедийный проектор с дистанционным управлением.

Учебные аудитории для семинарских и самостоятельных занятий по дисциплине оснащены ПЭВМ, с возможностью подключения к сети Интернет и доступом к электронной информационно-образовательной среде НИУ ВШЭ.

Компьютерные классы оборудованы ПЭВМ с доступом в Интернет, операционными системами и программным обеспечением, необходимыми для освоения дисциплины. При необходимости допускается замена оборудования его виртуальными аналогами.

#### **6. Особенности организации обучения для лиц с ограниченными возможностями здоровья и инвалидов**

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) а для инвалидов также в соответствии с индивидуальной программой реабилитации инвалида, могут предлагаться следующие варианты восприятия учебной информации с учетом их индивидуальных психофизических особенностей, в том числе с применением электронного обучения и дистанционных технологий:

6.1.1. *для лиц с нарушениями зрения:* в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); в печатной форме на языке Брайля; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации.

6.1.2. *для лиц с нарушениями слуха:* в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации.

6.1.3. *для лиц с нарушениями опорно-двигательного аппарата:* в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.