

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Никишкина Татьяна Юрьевна
Должность: Директор
Дата подписания: 30.01.2026 13:58:15
Уникальный программный ключ:
0a44cc3e5f5bb746f50173cba89718ffd301d635

ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА

**ОМСКИЙ ИНСТИТУТ ВОДНОГО ТРАНСПОРТА
(ФИЛИАЛ)**

**ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО
ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ВОДНОГО ТРАНСПОРТА»**

УТВЕРЖДАЮ

Заместитель директора по УиНРВО
ОИВТ (филиал) ФГБОУ ВО «СГУВТ»

Калекин В.В.

« _____ » _____ 20 _____ г.

Шифр дисциплины: 2024.09.03.02.01 . Б1.В.08
(шифр ОПОП) (шифр дисциплины из учебного плана)

Рабочая программа дисциплины (модуля)

Наименование дисциплины (модуля):

Современное программирование

(полное наименование дисциплины (модуля), в строгом соответствии с учебным планом)

Омск

2024
(год набора)

Составлена _____ доцентом _____ кафедры _____ ЕНиИТ _____
(должность) (сокращенное наименование кафедры)
к.ф.-м.н. _____, _____ Р.О. Карелиной _____
(ученая степень) (ученое звание) (И.О. Фамилия)

_____ Р.О. Карелина « _____ » 20 _____ г.
(подпись) (И.О. Фамилия) число месяц год

Рассмотрена и одобрена на заседании кафедры естественных наук
и информационных технологий

(наименование кафедры полностью)

от « _____ » 20 _____ г., Протокол № _____
число месяц год

Зав. кафедрой _____ ЕНиИТ _____ О.Л. Курнявко _____
(сокращенное наименование кафедры) (подпись) (И.О.Фамилия)

Согласована:

Руководитель ОПОП _____ к.т.н. _____ Т.Л.Иванова _____
(ученая степень, ученое звание) (подпись) (И.О. Фамилия)

Заведующий библиотекой ОИВТ _____ И.Д. Скоробогатова _____
(подпись) (И.О. Фамилия)

1 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

1.1 Цели дисциплины

Целью данной дисциплины является формирование у обучаемых понимания информационной сущности решаемых профессиональных задач, роли и места компьютера и телекоммуникационных систем в будущей профессиональной деятельности, выработки навыков и умений в определении основных этапов постановки и решения профессиональных задач с помощью современных информационных технологий.

Освоение различных языков и стилей программирования позволяет студентам свободно себя чувствовать в любой сфере ИТ и не бояться выйти за пределы традиционных подходов.

1.2 Перечень формируемых компетенций

В результате освоения дисциплины (модуля) у обучающегося должны сформироваться следующие компетенции, выраженные через результат обучения по дисциплине (модуля), как часть результата освоения образовательной программы (далее – ОП):

1.2.1 Универсальные компетенции (УК)

Дисциплина не формирует универсальные компетенции.

1.2.2 Общепрофессиональные компетенции (ОПК)

Дисциплина не формирует общепрофессиональные компетенции.

1.2.3 Профессиональные компетенции (ПК)

Компетенция		Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине
Шифр	Содержание		
ПК-2	Способность выполнять интеграцию программных модулей и компонент.	ПК-2.2. Знать методы и средства сборки и интеграции программных модулей и компонент. ПК-2.3. Уметь использовать выбранную среду программирования для разработки процедур интеграции программных модулей. ПК-2.5. Иметь навыки разработки процедур сборки модулей и компонент программного обеспечения.	Знать теоретические основы технологий и принципов процесса эффективной разработки программного обеспечения, основные подходы к интегрированию программных модулей. Уметь применять языки и среды программирования, средства пакетного выполнения процедур. Иметь навыки документирования контрольных примеров и тестовых наборов данных.

ПК-5	Способность выполнять работы по созданию (модификации) и сопровождению информационных систем.	ПК-5.1. Знать современные стандарты информационного взаимодействия систем, возможности, архитектуру, устройство и функционирование современных ИС. ПК-5.2. Знать современные методики тестирования разрабатываемых ИС, инструменты и методы модульного тестирования. ПК-5.4. Уметь создавать пользовательскую документацию к модифицированным элементам типовой ИС в соответствии с трудовым заданием. ПК-5.6. Иметь навыки работы с источниками информации, необходимой для профессиональной деятельности.	Знать современные методы, модели, принципы и средства эффективной разработки и сопровождения информационных систем. Уметь выполнять разработку прототипов типовой ИС в соответствии с трудовым заданием. Владеть навыками анализа и выявления первоначальных требований заказчика к типовой ИС.
ПК-6	Способен осуществлять безопасное техническое использование, техническое обслуживание судовой компьютерной информационной системы в соответствии с международными и национальными требованиями	ПК-6.1. Умеет осуществлять безопасное техническое использование компьютерной информационной системы в соответствии с международными и национальными требованиями; ПК-6.2. Умеет осуществлять безопасное техническое обслуживание судовой компьютерной информационной системы в соответствии с международными и национальными требованиями;	Знать: методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий. Уметь: применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий Владеть: Навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач

1.2.4 Профессиональные компетенции профиля или специализации (ПКС)

Дисциплина не формирует профессиональные компетенции профиля или специализации.

1.2.5 Компетенции МК ПДНВ (КМК)

Дисциплина не формирует компетенции МК ПДНВ (КМК).

2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина (модуль) реализуется в рамках

<u>вариативной</u>	части
(базовой, вариативной или факультативной)	

основной профессиональной образовательной программы.

3 Объем дисциплины (модуля) в зачетных единицах (ЗЕ) с указанием количества академических или астрономических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Для очной формы обучения:
(очной, очно-заочной или заочной)

Форма контроля						з.е.	-	Итого акад. часов					Курс 3									Курс 4																
													Семестр 6									Семестр 7																
Экзамен	Зачет	Зачет с оц.	КП	КР	Контр.	Экспертное	Факт	Часов в з.е.	Экспертное	По плану	Конт. раб.	СР	Контроль	з.е.	Итого	Лек	Лаб	Пр	Контр. раб.	Курс. пр-е	ИКР	СР	Контроль	з.е.	Итого	Лек	Лаб	Пр	Контр. раб.	Курс. пр-е	ИКР	СР	Контроль					
6		7	7			9	9	36	324	324	111	177	36	5	180	24	30				4	86	36	4	144	20	24				9	91						
в том числе тренажерная подготовка:																																						

Для заочной формы обучения:
(очной, очно-заочной или заочной)

Форма контроля						з.е.		-	Итого акад. часов						Курс 3										Курс 4																
															Летняя сессия										Летняя сессия																
Экзамен	Зачет	Зачет с оц.	КП	КР	Контр.	Экспертное	Факт	Часов в з.е.	Экспертное	По плану	Конт. раб.	СР	Контроль	з.е. на курсе	Итого	Лек	Лаб	Пр	Контр. раб.	Курс. пр-е	ИКР	СР	Контроль	Формы контр.	з.е. на курсе	Итого	Лек	Лаб	Пр	Контр. раб.	Курс. пр-е	ИКР	СР	Конт роль	Формы контр.						
3		4	4			9	9	36	324	324	44	262	18	4	144	6	8				4	108	18	э	5	180	8	10				8	154		оп						
в том числе тренажерная подготовка:																																									

4 Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1 Разделы и темы дисциплины (модуля) и трудоемкость по видам учебных занятий (в академических часах)

№	Разделы и темы дисциплины (модуля)	Виды учебных занятий, включая СРС											
		Лекции			Лаб			Пр			СРС		
		О	ОЗ	З	О	ОЗ	З	О	ОЗ	З	О	ОЗ	З
1	Язык программирования C++	16			20		8				52		80
2	Основы Linux	8			10						34		38
3	Функциональное программирование	4		4	8						12		18
4	Проектирование высоконагруженных систем	4		4	8		2				12		18
5	Программная инженерия	2		4	8		8				12		18
6	Параллельное программирование	2		2							12		18
7	Архитектура JVM	2									12		18
8	Введение в метавычисления	2									11		18
9	Выпуклая оптимизация	2									10		18
10	Трёхмерное компьютерное зрение	2									10		18
Итого		44		14	54		18				177		262

Примечания: О – очная форма обучения, ОЗ – очно-заочная форма обучения, З – заочная форма обучения.

4.2 Содержание разделов и тем дисциплины

Тема 1. Язык программирования C++

Разработка объектно-ориентированных программ на C++. Обработка ошибок (в том числе, исключения). Разработка библиотек с помощью обобщенного программирования (шаблоны) и метапрограммирования. Обзор стандартной библиотеки STL. Элементы многопоточного программирования. Элементы проектирования. Обзор последних версий стандарта C++.

Тема 2. Основы Linux

Работа в командной строке: доступ к файловой системе, управление процессами, запуск программ и их взаимодействие, переменные окружения, важнейшие утилиты командной строки. Программирование на языке bash: переменные, ввод-вывод, управляющие инструкции, функции. Элементы DevOps: сборка ПО, контейнеризация, интеграция и доставка.

Тема 3. Функциональное программирование

Лямбда-исчисление: термы, редукция, нормальная форма, стратегии редукции, теорема Черча-Россера, теорема о неподвижной точке и рекурсия. Типы: простая система типов, версии Карри и Черча, проверка, вывод, обитаемость, алгоритм Хиндли-Милнера.

Изучение языка программирования Haskell. Язык Haskell: нестрогая семантика, алгебраические типы данных, сопоставление с образцом; полугруппы и моноиды; свертки и траверсы; функторы, аппликативные функторы и монады; трансформеры монад.

Тема 4. Проектирование высоконагруженных систем

Программирование на Go. Web-сервисы. Распределенные системы. Сервисная архитектура. • Мониторинг. Поставка приложения.

Тема 5. Программная инженерия

Управление программными проектами Инструменты и средства командной разработки Проектирование и архитектура программных систем Принципы построения высоконагруженных распределенных систем Сервисы и микросервисная архитектура Работа с базами данных, ORM системы Юнит тестирование и интеграционное тестирование Рефакторинг и чистый код.

Работа в командах по Agile . Использование современных инструменты командной разработки (Jira, Azure DevOps, Git) . Проектирование, разработка и интегрирование отдельных компонент в общий проект. Настройка процесса CI/CD.

Тема 6. Параллельное программирование

Процессы/потoki: критерии выбора архитектуры, жизнь в ядре ОС. Синхронизация: примитивы, алгоритмы, lock-free, wait-free, транзакционная память, консенсус. Шаблоны параллельного программирования: базовые шаблоны, асинхронный ввод/вывод. Иные технологии повышения производительности: компиляторы, SSE, OpenCL. Профилирование и поиск ошибок. Технологии: потоки ОС, OpenMP, Intel TBB, Java.util.concurrent, Coroutines.

Тема 7. Архитектура JVM

Обзор архитектуры JVM. Байт-код, классы и методы. Модель памяти, вызовы методов. Интерпретатор и JIT-компилятор. Сборка мусора.

Тема 8. Введение в метавычисления

Представление о теоретических и практических аспектах метавычислений — области Computer Science, которая изучает методы преобразования программ.

Тема 9. Выпуклая оптимизация

Основные концепции математической оптимизации и базовые методы для решения различных прикладных задач. Квадратичные функции. Метод множителей Лагранжа и его обобщения. Основные разновидности градиентного спуска. Метод Ньютона и метод внутренней точки.

Тема 10. Трёхмерное компьютерное зрение

Задачи извлечения информации из изображений и видео, возникающие во множестве областей: от промышленности до кино. Идеи и классические алгоритмы компьютерного зрения. Азы обработки изображений, линейные фильтры и свертки. Ключевые точки: детекторы, дескрипторы, сопоставление, оптический поток.

4.3 Содержание лабораторных работ

Раздел (тема) дисциплины	Тема лабораторной работы
Язык программирования C++	Основные конструкции, возможности C++.
Основы Linux	Работа в командной строке: доступ к файловой системе, управление процессами, запуск программ и их взаимодействие, переменные окружения, важнейшие утилиты командной строки. Программирование на языке bash: переменные, ввод-вывод, управляющие инструкции, функции.
Функциональное программирование	Разработка программного обеспечения на языке Haskell
Проектирование высоконагруженных систем	Программирование на Go
Программная инженерия	Работа в командах по Agile. Современные инструменты командной разработки (Jira, Azure DevOps, Git)

4.4 Содержание практических занятий

Практические занятия не предусмотрены.

4.5 Курсовая работа

Курсовая работа не предусмотрена.

4.6 Курсовой проект

Дисциплина не предусматривает курсовой проект.

4.7 Самостоятельная работа. Контроль самостоятельной работы

В самостоятельную работу студента входит подготовка к занятиям по дисциплине путём изучения теоретического материала по источникам, указанным в перечне основной и дополнительной литературы, выполнение лабораторных работ, оформление отчетов по лабораторным работам и подготовка к их защите.

Контроль самостоятельной работы осуществляется в ходе лабораторных занятий.

4.8 Интерактивные формы обучения

В процессе освоения данной учебной дисциплины интерактивные формы используются при изучении следующих тем:

Раздел (тема) дисциплины	Количество часов		
	О	ОЗ	З
Язык программирования C++	8		8
Итого	8		8

5 Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)

5.1 Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Показатели оценивания	Описание показателя
Оценка «отлично» (зачтено)	Обучаемый демонстрирует способность к полной самостоятельности (допускаются консультации с преподавателем по сопутствующим вопросам) в выборе способа решения неизвестных или нестандартных заданий с использованием знаний, умений и навыков, полученных в ходе проведения освоения дисциплины, следует считать компетенцию сформированной на высоком уровне. Присутствие сформированной компетенции на высоком уровне означает способность студента к ее дальнейшему саморазвитию и высокой адаптивности практического применения к изменяющимся условиям профессиональной задачи.
Оценка «хорошо» (зачтено)	Обучаемый демонстрирует самостоятельное применение знаний, умений и навыков при решении заданий, аналогичных тем, которые предоставлял преподаватель при потенциальном формировании компетенции, подтверждает наличие сформированной компетенции, причем на более высоком уровне. Наличие сформированной компетенции на повышенном уровне самостоятельности со стороны обучаемого при ее практической демонстрации в ходе решения аналогичных заданий следует оценивать как положительное и устойчиво закрепленное в практическом навыке.
Оценка	Обучаемый демонстрирует самостоятельность в применении знаний, умений и

Показатели оценивания	Описание показателя
«удовлетворительно» (зачтено)	навыков к решению учебных заданий в полном соответствии с образцом, данным преподавателем, по заданиям, решение которых было показано преподавателем, следует считать, что компетенция сформирована, но ее уровень недостаточно высок. Поскольку выявлено наличие сформированной компетенции, ее следует оценивать положительно, но на низком уровне.
Оценка «неудовлетворительно» (не зачтено)	Обучаемый не способен самостоятельно продемонстрировать наличие знаний при решении заданий, которые были представлены преподавателем вместе с образцом их решения, отсутствие самостоятельности в применении умения к использованию методов проведения преддипломной практики и неспособность самостоятельно проявить навык повторения решения поставленной задачи по стандартному образцу свидетельствуют об отсутствии сформированной компетенции.

5.2 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений и (или) навыков, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Качественный вопрос: Чему будет равна переменная `a` после выполнения этого кода :

```
int a; for(a = 0; a < 10; a++) {}?
```

- ☐ 9
- ☐ 1
- ☐ 10

Задача: Какой из следующих операторов - оператор сравнения двух переменных?

- ☐ `:=`
- ☐ `=`
- ☐ `equal`
- ☐ `==`

Задача: Чтобы подключить заголовочный файл в программу на C++, например, `iostream`, необходимо написать:

- ☐ `include (iostreamh)`
- ☐ `include #iostream,h;`
- ☐ `#include <>;` с `iostream.h` внутри скобок
- ☐ `#include <>` с `iostream` внутри скобок

5.3 Перечень вопросов для итогового контроля по дисциплине

1. Разработка объектно-ориентированных программ на C++.
2. Обработка ошибок (в том числе, исключения).
3. Разработка библиотек с помощью обобщенного программирования (шаблоны) и метапрограммирования.
4. Обзор стандартной библиотеки STL.
5. Элементы многопоточного программирования.
6. Элементы проектирования.
7. Обзор последних версий стандарта C++.

8. Работа в командной строке: доступ к файловой системе, управление процессами, запуск программ и их взаимодействие, переменные окружения, важнейшие утилиты командной строки.
9. Программирование на языке `bash`: переменные, ввод-вывод, управляющие инструкции, функции.
10. Элементы DevOps: сборка ПО, контейнеризация, интеграция и доставка.
11. Лямбда-исчисление: термы, редукция, нормальная форма, стратегии редукции, теорема Черча-Россера, теорема о неподвижной точке и рекурсия.
12. Типы: простая система типов, версии Карри и Черча, проверка, вывод, обитаемость, алгоритм Хиндли-Милнера.
13. Изучение языка программирования Haskell.
14. Язык Haskell: нестрогая семантика, алгебраические типы данных, сопоставление с образцом; полугруппы и моноиды; свертки и траверсы; функторы, аппликативные функторы и монады; трансформеры монад.
15. Программирование на Go. Web-сервисы. Распределенные системы. Сервисная архитектура. Мониторинг. Поставка приложения.
16. Управление программными проектами. Инструменты и средства командной разработки
17. Проектирование и архитектура программных систем. Принципы построения высоконагруженных распределенных систем.
18. Сервисы и микросервисная архитектура. Работа с базами данных, ORM системы.
19. Юнит тестирование и интеграционное тестирование. Рефакторинг и чистый код.
20. Работа в командах по Agile. Использование современных инструментов командной разработки (Jira, Azure DevOps, Git). Проектирование, разработка и интегрирование отдельных компонент в общий проект. Настройка процесса CI/CD.
21. Процессы/потoki: критерии выбора архитектуры, жизнь в ядре ОС.
22. Синхронизация: примитивы, алгоритмы, `lock-free`, `wait-free`, транзакционная память, консенсус.
23. Шаблоны параллельного программирования: базовые шаблоны, асинхронный ввод/вывод.
24. Иные технологии повышения производительности: компиляторы, SSE, OpenCL.
25. Профилирование и поиск ошибок.
26. Технологии: потоки ОС, OpenMP, Intel TBB, `Java.util.concurrent`, `Coroutines`.
27. Обзор архитектуры JVM. Байт-код, классы и методы.
28. Модель памяти, вызовы методов. Интерпретатор и JIT-компилятор. Сборка мусора.
29. Представление о теоретических и практических аспектах метавычислений — области Computer Science, которая изучает методы преобразования программ.
30. Основные концепции математической оптимизации и базовые методы для решения различных прикладных задач.
31. Квадратичные функции. Метод множителей Лагранжа и его обобщения. Основные разновидности градиентного спуска. Метод Ньютона и метод внутренней точки.
32. Задачи извлечения информации из изображений и видео, возникающие во множестве областей: от промышленности до кино.
33. Идеи и классические алгоритмы компьютерного зрения. Азы обработки изображений, линейные фильтры и свертки.
34. Ключевые точки: детекторы, дескрипторы, сопоставление, оптический поток.

5.4 Методические рекомендации к проведению итогового контроля по дисциплине

На итоговый контроль выносится учебный материал, рассмотренный на лекционных и практических занятиях, а также темы, которые были предложены для самостоятельного изучения.

Формой проведения итогового контроля по данной дисциплине является экзамен, который проводится в устной форме с использованием билетов, содержащих теоретическую и практическую часть. Также допустимо проведение итогового контроля в письменной форме. О форме проведения итогового контроля студенты должны быть оповещены заблаговременно.

Во время подготовки к ответу на зачете использование справочной литературы не допускается.

6 Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

6.1 Основная учебная литература

Шлее М. Ш68 Qt 5.10. Профессиональное программирование на C++. — СПб.: БХВ-Петербург, 2018. — 1072 с.: ил. — (В подлиннике). <https://monster-book.com/reader/22716>

6.2 Дополнительная учебная литература

Энтони Уильямс Параллельное программирование на C++ в действии. Практика разработки многопоточных программ. Пер. с англ. Слинкин А. А. – М.: ДМК Пресс, 2012. – 672с.: ил

6.3 Периодические издания

1. Вестник Омского университета [Текст]: ежекварт. науч. журнал/ Мин-во образ.и науки РФ, Омский гос. ун-т. – Омск, 2013-2016
2. Известия ТулГУ. Естественные науки [Текст]: ежемес. науч. журнал/ Мин-во образ.и науки РФ, Тульский гос. ун-т. - Тула, 2008-2015
3. Информатика и образование [Текст] : научно-метод. журнал/ Российская Академия образования. - М. : Образование и информатика , 1986 - . - Выходит ежемесячно. - ISSN 0234-0453

7 Перечень ресурсов информационно-телекоммуникационной сети "Интернет" (далее - сеть "Интернет"), необходимых для освоения дисциплины (модуля)

1. Электронная библиотечная система «Лань» [Электронный ресурс]. – Режим доступа:<https://e.lanbook.com/>
2. Научная электронная библиотека [Электронный ресурс]. – Режим доступа: <https://elibrary.ru/>
3. Математический портал «EqWorld» [Электронный ресурс]. – Режим доступа: <http://eqworld.ipmnet.ru/ru/info/mathwebs.htm>
4. Электронный каталог НГАВТ [Электронный ресурс]. – Режим доступа: <http://libcat.nsawt.ru/>

8 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

1. ОС Windows-7/8/10
2. MS Word
3. MS Excel
4. Microsoft Office Power Point
5. Microsoft Visual Studio – среда разработки программных приложений.

9 Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

№ п/п	Наименование вида образования, уровня образования, профессии, специальности, направления подготовки (для профессионального образования), подвида дополнительного образования	Наименование оборудованных учебных кабинетов, объектов для проведения практических занятий, объектов физической культуры и спорта с перечнем основного оборудования	Адрес (местоположение) учебных кабинетов, объектов для проведения практических занятий, объектов физической культуры и спорта (с указанием номера помещения в соответствии с документами бюро технической инвентаризации)
15.16	Информатика	Учебная аудитория для проведения занятий лекционного типа: ученическая парта - не менее 42 шт., стол преподавательский - не менее 1 шт., доска учебная - не менее 1 шт., экран проекционный - не менее 1 шт., стул - не менее 84 шт., рамки портретные - не менее 23 шт.	644099, г. Омск, ул. Ивана Алексеева, д. 2 (№№ 31,32 на поэтажном плане (1П), 4 этаж)
		Лаборатория вычислительной техники: операционная система Windows, пакет программ MicrosoftOffice, программы: VisualBasicforApplication, Mathcad, MatLab, КОМПАС-3D, СПС Консультант ПЛЮС, ГАРАНТ, доска - не менее 2 шт., экран - не менее 2 шт., шкаф - не менее 2 шт, стенды - не менее 10 шт., стол - не менее 43 шт., стул - не менее 48 шт.	644099, г. Омск, ул. Ивана Алексеева, д. 2 (№№ 52, 53,54,55 на поэтажном плане (1П), 3 этаж)
		Учебная аудитория для проведения групповых и индивидуальных консультаций: ученическая парта - не	

		менее 21 шт., стол преподавательский - не менее 1 шт., доска учебная - не менее 1 шт., экран проекционный - не менее 1 шт., стул - не менее 43 шт., рамки портретные - не менее 6 шт., плакаты - не менее 5 шт.	
		Учебная аудитория для проведения текущего контроля и промежуточной аттестации: операционная система Windows, пакет программ MicrosoftOffice, программы: VisualBasicforApplication, Mathcad, MatLab, КОМПАС-3D, СПС Консультант ПЛЮС, ГАРАНТ, доска - не менее 2 шт., экран - не менее 2 шт., шкаф - не менее 2 шт., стенды - не менее 10 шт., стол - не менее 43 шт., стул - не менее 48 шт.	
		Помещение для самостоятельной работы: операционная система Windows, пакет программ MicrosoftOffice, программы: VisualBasicforApplication, Mathcad, MatLab, КОМПАС-3D, СПС Консультант ПЛЮС, ГАРАНТ, доска - не менее 2 шт., экран - не менее 2 шт., шкаф - не менее 2 шт., стенды - не менее 10 шт., стол - не менее 43 шт., стул - не менее 48 шт.	

\