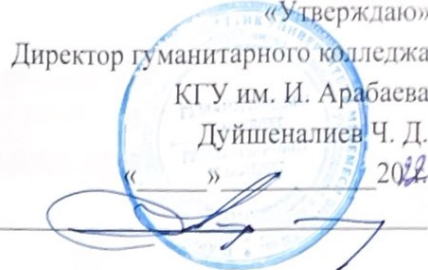


МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ
РЕСПУБЛИКИ

ГУМАНИТАРНЫЙ КОЛЛЕДЖ КЫРГЫЗСКОГО
ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА имени И. АРАБАЕВА

«Утверждаю»
Директор гуманитарного колледжа
КГУ им. И. Арабаева
Дуйшеналиев Ч. Д.
« » 2022



ТИПОВАЯ ПРОГРАММА

по дисциплине «Алгоритмические языки программирования»

Разработчик (должность) ст.преподаватель

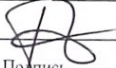
Ф.И.О. Тыналиева Ч.Т.

Заведующий отделением «Информатика и Дизайн»

Ф.И.О. Турганбаева Б. Т.

Принято на заседании отделения « » 2022г.

№ протокола _____


Подпись

Рекомендован
Пред. УМС Гум. Колледжа
КГУ им. И. Арабаева
ст. преп. Янгибаева Ж.Т.

« » 2022г.

г. Бишкек.

Типовая программа

Дисциплина "Алгоритмические языки программирования" _____
Направление(специальность) 220206 Автоматизированные системы
обработки информации и управление (по отраслям) _____
Форма обучения очное _____
Курс 2 Семестр 4 _____
Часов: всего 36, лекции 22, практических занятий 14 _____
СРС и виду индивидуальных занятий (курсовые работы) 24 _____
Обеспечивающее отделение «Информатика и Дизайн»

ОГЛАВЛЕНИЕ

1. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЙ РАЗДЕЛ.....	
1.1 Цели и задачи дисциплины	
1.2 Место дисциплины в структуре образовательной программы	
1.3 Требования к результатам освоения содержания дисциплины	
1.4 Объем дисциплины и виды учебной работы	
1.5 Критерий баллов – рейтинговой оценки знаний и умений студентов	
2. СОДЕРЖАНИЕ ПРОГРАММЫ УЧЕБНОГО КУРСА.....	
2.1 Содержание разделов дисциплины	
2.2 Задания для самостоятельной работы студентов.....	
3. ОЦЕНОЧНЫЕ СРЕДСТВА ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ...	
3.1 Вопросы к экзамену, модулям	
4. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ.....	
4.1 Учебно-методические: основная и дополнительная литература.....	
4.2 Интернет-ресурсы.....	
4.3 Материально-техническое обеспечение дисциплины.....	

1. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЙ РАЗДЕЛ

Назначение учебной дисциплины

Учебная дисциплина направлена на формирование у студентов знаний об основных принципах алгоритмизации и теории алгоритмов, программировании, а также формирование практических навыков создания прикладных программных продуктов на основе современных технологий программирования с использованием одного из наиболее распространенных алгоритмических языков. Расширение знаний учащихся в направлении изучения языков программирования, в частности, языка программирования Python.

1.1 Цель изучения дисциплины

Целью дисциплины «Алгоритмические языки программирования» является: изучение и освоение базовых понятий и приемов программирования, применяемых на всех основных этапах разработки программ; изучение методов программирования для овладения знаниями в области технологии программирования; подготовка к осознанному использованию как языков программирования, так и методов программирования. Воспитательной целью дисциплины является формирование у студентов научного, творческого подхода к освоению технологий, методов и средств производства программного обеспечения.

1.2 Задачи изучения дисциплины

- понять значение алгоритмизации как метода познания окружающего мира, принципы структурной алгоритмизации;
- развитие интереса учащихся к изучению программирования;
- овладеть базовыми понятиями теории алгоритмов при решении математических задач;
- научиться разрабатывать эффективные алгоритмы и программы для числовых рядов, прогрессий, значений многочленов, массивов, в области арифметики рациональных чисел;
- приобретение навыков работы в системе программирования Python;
- формирование самостоятельности и творческого подхода к решению задач с помощью средств современной вычислительной техники. формирование навыков алгоритмического и логического мышления; формирование навыков грамотной разработки программ.

1.3 Требования к результатам освоения содержания дисциплины

В соответствии с целями ОПОП и задачами профессиональной деятельности, должен обладать следующими компетенциями:

а) универсальными:

общенаучными(ОК):

- владеть целостной системой научных знаний об окружающем мире, способен ориентироваться в ценностях жизни, культуры (ОК-1);

- способен использовать базовые положения математических/естественных/гуманитарных/экономических наук при решении профессиональных задач(ОК-2);
- способен приобретать новые знания, с большой степени самостоятельности, с использованием современных образовательных и информационных технологий(ОК-3);
- способен анализировать и оценивать социально-экономические и культурные последствия новых явлений в науке, технике и технологии, профессиональной сфере(ОК-4);
- способен на научной основе оценивать свой труд, оценивать, с большой степенью самостоятельности, результаты своей деятельности(ОК-5);

инструментальными(ИК):

- способен воспринимать, обобщать и анализировать информацию, ставить цели и выбирать пути ее достижения(ИК-1);
- способен логически верно, аргументировано и ясно строить свою и письменную речь на государственном и официальном языках(ИК-2);
- владеть одним из иностранных языков на уровне социального общения(ИК-3);
- владеть основными методами, способами и средствами получения, хранения и переработки, навыками работы с компьютером, как средством управления информацией, в том числе в глобальных компьютерных сетях и корпоративных информационных системах(ИК-4);

1.4 Объем дисциплины и виды учебной работы

Виды учебной работы	Всего часов	Семестр
Общая трудоемкость дисциплины	64	64
Аудиторные занятия	40	40
Лекции	22	22
Практические занятия	14	14

Прием и контроль	4	4
Самостоятельная работа	24	24
Вид итогового контроля (зачет, экзамен)	экзамен	экзамен

1.5 Критерий баллов – рейтинговой оценки знаний и умений студентов

Деятельность студентов в течение семестра оценивается следующим образом: работа на семинарах (50%), самостоятельные работы и реферат (20%), активность (25%), посещение занятий (5%).

Работа на семинарах (50%)

Чтение текстов и участие в дискуссиях являются важными составляющими работы на семинарах. Приветствуются вопросы по структуре и содержанию текста, комментарии, помогающие уяснить значение основных категорий и т.п.

Пропущенные семинары необходимо отработать письменно.

«Отработка» должна содержать основные моменты пропущенной темы занятия. Оценка за «отработки» не выставляется. Последний срок сдачи «отработок» - заключительное занятие по курсу (тем, кто не сможет присутствовать на заключительном занятии «отработку» необходимо принести заранее).

Неотработанные семинары являются основанием незачета по данному курсу.

Критерии оценки: регулярное присутствие и активное участие, уместность и глубина вопросов и комментариев, способность задавать живой импульс дискуссии и вовлекать других студентов в дебаты.

Оценки за активность на семинарах выставляются по 10-ти балльной шкале.

Критерии оценки работы студентов на семинарах следующие:

10 баллов – индивидуальный ответ, изложенный по существу структурно, логично, своими словами.

8-9 баллов – индивидуальный ответ, изложенный своими словами. Возможны мелкие проблемы с логикой изложения.

5-7 баллов – индивидуальный ответ, изложенный частично своими словами. Возможны мелкие проблемы с логикой изложения.

1-4 балла – индивидуальный ответ – уточнение (дополнение) по рассматриваемым вопросам семинарского занятия, задаваемые вопросы.

Самостоятельные работы и реферат (20%)

Самостоятельные работы выполняются на отдельном листочке письменно от руки. Указывается имя, фамилия, группа и дата сдачи работы.

Все письменные работы НЕ принимаются позже установленных сроков сдачи, за исключением документально подтвержденных случаев

отсутствия вследствие болезни или форс-мажорных обстоятельств.

Критерии оценки письменных работ следующие:

10 – выдающаяся работа на высоком уровне, присутствует логика и оригинальность изложения, выдвинут и доказан тезис, видно уверенное владение освоенным материалом.

8-9 – очень хорошая работа, продемонстрированы не только усвоенные знания по курсу, но навыки анализа материала и самостоятельного мышления. Возможны мелкие проблемы с логикой изложения.

6-7 – хорошая работа, продемонстрированы не только усвоение фактических знаний по курсу и основные навыки аргументации, но изложение не вполне закончено с точки зрения обоснования тезиса и раскрытия вопроса.

4-5 – средняя работа, неполное усвоение фактических знаний по курсу, слабая логика изложения и обоснования.

2-3 – плохая работа, отрывочные знания по курсу, слабая логика изложения и обоснования.

1 – отсутствие каких-либо знаний.

0 – доказанный случай плагиата.

Темы рефератов студенты выбирают согласно нумерации по учебному журналу.

Реферативная работа оформляется письменно от руки. Допускается печатное исполнение титульного листа, списка литературы, графических и табличных приложений.

Студенты, вовремя не сдавшие реферат, защищают свою работу на консультации или в дополнительно отведенное время.

Своевременное выполнение работ является предпосылкой к обоснованию возможности допуска студента к зачету (экзамену).

Проверка уровня усвоения лекционных занятий, включая теоретических СРС и СРСП, проводится тестированием по рейтинго-модульной системе. Каждый тест включает 15 вопросов, где правильный ответ на 1 вопрос оценивается на 1 балл.

Результаты практических работ, включая, практических СРС и СРСП принимаются в виде графических и контрольных работ, рефератов и собеседования.

Система оценки знаний

№	Этапы проверки	Вид средства проверки	Баллы	Сроки
1	1 модуль	Тестирование	35	Согласно графику учебного процесса
2	2 модуль	Проверка заданий	35	Согласно графику учебного процесса
3	Практические СРС	Контрольные и графические работы, рефераты.	10	В течение семестра, до итогового контроля

		собеседование		
4	Поощрительные баллы за активность		7	В конце семестра, до итогового контроля
5	Посещение занятий		3	В течение семестра
6	Итоговый контроль	Письменный или устный	10	Согласно графику учебного процесса
	Итого:		100	

2. СОДЕРЖАНИЕ ПРОГРАММЫ УЧЕБНОГО КУРСА

2.1 Содержание разделов дисциплины

План изучения дисциплины (лекции):

№	Темы дисциплины	Количество часов
		лекции
	Глава 1. Теоретические основы алгоритмизации и программирования	
1.	Тема 1.1. Алгоритм и его свойства. Тема 1.2. Формы записи алгоритмов	1
2.	Тема 1.3. Данные и их типы Тема 1.4. Основные этапы решения задач	1
3.	Тема 1.5. Основные структуры алгоритмов	1
4.	Тема 1.6. Язык программирования Тема 1.7. Классификация языков программирования	1
5.	Тема 1.8. Методы и принципы программирования	1
6.	Тема 1.9. Виды программного обеспечения. Общие принципы разработки ПО	1
	Глава 2. Язык программирования Python	
7.	Тема 2.1. Введение в язык программирования Python. Среда Python.	1
8.	Тема 2.2. Основные элементы языка Python	1
9.	Тема 2.3. Типы данных в Python	1
10.	Тема 2.4. Ввод и вывод данных	1
11.	Тема 2.5. Операторы и выражения, Модуль math, Модуль random	1

	Глава 3. Основные алгоритмические инструкции языка Python	
12.	Тема 3.1. Линейный алгоритм (последовательность действий)	1
13.	Тема 3.2. Разветвляющиеся алгоритм: Простой условный оператор. Сокращенный условный оператор. Составной условный оператор.	1
14.	Тема 3.3. Многозначные ветвления. Алгоритмы поиска максимального и минимального элементов.	1
15.	Тема 3.4. Циклический алгоритм : Оператор цикла WHILE .	1
16.	Тема 3.5. Оператор цикла FOR. Функции range(). Вложенные циклы	1
	Глава 4. Коллекции в Python	
17.	Тема 4.1. Строки и операции с ними. Методы строк.	1
18.	Тема 4.2. Сравнение и сортировка строк. Форматирование строк	1
19.	Тема 4.3. Множества в Python	1
20.	Тема 4.4. Списки в Python	1
21.	Тема 4.5. Кортежи в Python	1
22.	Тема 4.6. Словари в Python	1
	Всего:	22

1.6. План изучения дисциплины (практические)

№	Наименование тем	Кол. часов
1.	Организация ввода-вывода данных	1
1.	Запись арифметических выражений, линейный алгоритм	2
2.	Многозначные ветвления в программах	2
3.	Программирование алгоритмов циклических структур	2
	Вложенные условные операции и циклы	1
4.	Строки и операции с ними, использование срезов для обработки строк, сравнение и сортировка строк	1
5.	Форматирование строк	1

6.	Работа с кортежами.	1
7.	Работа с множествами.	1
8.	Работа со списками в Python	1
9.	Работа со словарями в Python	1
	Всего:	14

2.2 Задания для самостоятельной работы студентов

СРС:

1. Разработка программного обеспечения сегодня
2. Алгоритм: способы представления и свойства
3. Языки программирования
4. Классификация языков программирования
5. Выбор языка программирования
6. Почему Python стоит изучать?
7. Алфавит языка Python
8. Основные типы данные в Python.
9. Методы ввода и вывода данных
10. Линейные алгоритмы. Операции над целочисленными данными
11. Запись математических функций, модуль MATH
12. Разветвляющиеся алгоритм
13. Основные алгоритмические инструкции языка Python
14. Многозначные ветвления
15. Циклический алгоритм, оператор цикла FOR
16. Циклический алгоритм, оператор цикла While
17. Вложенные условные операции и циклы
18. Кортежи в Python
19. Методы работы со строками
20. Базовые алгоритмы обработки строк
21. Словари в Python
22. Множества в Python
23. Списки в Python
24. Коллекции в Python, их классификация.

3. ОЦЕНОЧНЫЕ СРЕДСТВА ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

3.1 Вопросы к экзамену, модулям

Контрольные вопросы

1. Этапы решения задач на ЭВМ.
2. Алгоритм. Свойства алгоритма. Формы записи алгоритма.
3. Данные и их типы. Методы сортировки данных.
4. Основные структуры алгоритмов.
5. Логические высказывания и операции.
6. Таблицы истинности. Свойства логических операций.
7. Язык программирования. Поколения, классификация, элементы языков программирования.
8. Система программирования. Классификация СП.
9. Структура системы программирования. Методы программирования.
10. Виды программного обеспечения. Общие принципы разработки ПО. Жизненный цикл ПО.
11. Язык программирования Python. Среда программирования Wing IDE.
12. Основные элементы языка. Типы данных в Python. Операторы и выражения.
13. Для каких целей используются комментарии в программах? Как можно закомментировать участок программного кода в Python?
14. Каково назначение операторов print и input? Приведите примеры использования таких операторов.
15. Программирование алгоритмов линейной структуры.
16. Программирование алгоритмов ветвления. Вложенные условия.
17. Программирование алгоритмов циклической структуры. True и False, break и continue. Вложенные циклы.
18. Расскажите о работе оператора цикла while. Приведите примеры. Расскажите о работе цикла с оператором for по убывающим значениям параметра цикла.
19. Коллекции в Python, их классификация.
20. Строки. Функции и методы строк.
21. Множества. Операции над множествами.
22. Списки. Методы списков. Вложенные списки.
23. Напишите синтаксис объявления списков.
24. Кортежи. Преобразование коллекций.
25. Напишите синтаксис объявления кортежей.
26. Словари. Методы словаря.
27. Напишите синтаксис создания словаря.

Тест за 1 модуль:

1. Алгоритм, записанный на понятном компьютеру языке, называется:
 - a. исполнителем;
 - b. программой;
 - c. блок-схемой;
 - d. системой команд исполнителя

е. псевдокодом.

2. Где записана команда присваивания?:

- a. $X+Y:=X$ b. $F=G$
c. $X:=X+Y$ d. $A>D$

3. Выберите тип величины, который следует использовать для обозначения количества учеников в классе:

- a. числовой целый; b. числовой вещественный;
c. строковый; d. логический.

4. Какое арифметическое выражение записано правильно?

- a. $A1+B1*50$; b. $6A-23B$;
c. $b2 - 4ac$ d. $67*A2-30*B$.

5. Укажите логические выражения:

- a. $X+7$; b. $X+7>=0$; c. $X:=7$; d. $N=10$.

6. Выберите верные утверждения:

- a. одна величина может иметь несколько типов;
b. значение переменной может изменяться в процессе выполнения алгоритма;
c. величина логического типа может принимать всего два значения;
d. при присваивании переменной какого-либо значения предыдущее её значение сохраняется автоматически.

7. Как выглядит комментарий в Python:

- a. "это пустая программа" b. #это пустая программа
c. 'это пустая программа'

8. Что такое "переменная":

- a. это величина, имеющая имя, тип и значение. Значение переменной нельзя изменять во время работы программы.
b. это знак проверки условия
c. это величина, значение которой нельзя менять во время работы программы.
d. это величина, имеющая имя, тип и значение. Значение переменной можно изменять во время работы программы.

9. Какие типы переменных используются в Python: Строчные

- d. Кинетическое число
a. Символьная строка e. Вещественное число
c. Целое число f. Логическая

10. Команда для присваивания нового значения переменной:

- a. Оператор Конъюнкция b. Оператор присваивания
c. Условный оператор

11. Что будет в результате работы программы:

```
a=input()
b=input()
c=a+b
print (c)
```

Входные данные a=21 b=33.

- a. 54 b. 21 c. 33 d. 2133

12. Как ввести целое число с клавиатуры:

- a. a=input() b. a=int(input())
c. a=10 d. a=str(input())

13. Что будет в результате выполнения программы (чему равно a и b):

```
a=5
b=a+2
a=(a+2)*(b-3)
b=b+1.
```

- a. a=2 b=20 b. a=8 b=28
c. a=28 b=8

14. Возведение в степень в Python:

- a. // b. **
c. % d. + e. e^

15. Целочисленное деление:

- a. // b. % c. / d. *

16. Что будет в результате такого деления:

```
a=1234
d=a%10
print(d).
```

- a. 5 b. 4 c. 3 d. 2

17. Что выведет вторая d:

```
a = 1234
d = a % 10; print( d )
a = a // 10
d = a % 10; print( d ).
```

- a. 3 b. 4 c. 5 d. 6

18. Как импортировать библиотеку случайных чисел:

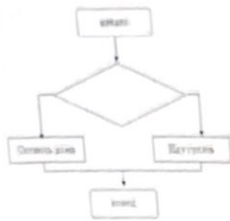
- a. from random import b. random.randint
c. random.random d. import random

19. Знак равенства в Python, варианты ответов:

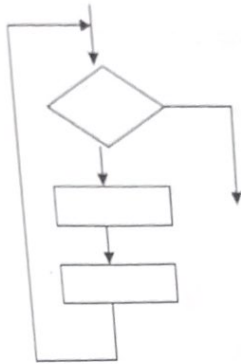
- a. != b. >< c. == d. <=

20. Как получить данные от пользователя?

- a. Использовать метод read() b. Использовать метод get()



- а. Линейный
 б. Разветвляющийся
 в. Циклический
 г. Смешанный
3. Назовите тип алгоритма:



- а. Разветвляющийся
 б. Линейный
 в. Смешанный
 г. Циклический
4. Что хранит в себе переменная?
- а. Имя
 б. Значение
 в. Тип
 г. Длину своего значения
5. Какой оператор здесь используется?
- If $n < 100$:
 $b = n + a$
- а. Условный оператор
 б. Оператор присваивания
 в. Оператор сложения
 г. Оператор умножения
6. Что лучше использовать для множественного ветвления?
- а. if – elif – else
 б. Много if
 в. if – else – elif
 г. while
7. Оператор цикла в языке Python:
- а. while
 б. for
 в. if
 г. print
8. Для чего нужен оператор break?
- а. Для завершения программы
 б. Для выхода из цикла
 в. Для поломки компьютера
 г. Для удаления программы
9. Сколько раз программа напишет слово «Пока»?

```
k=0
```

```
while k<10:
```

```
    print(«Привет»)
```

```
    k += 1
```

a. 9

b. 0

c. 10

d. Бесконечно

10. Что такое «else»?

a. Так как

b. Иначе

c. Если

d. Потому что

11. Выберите циклический алгоритм

a. k = 0

```
while k < 10:
```

```
    print("Привет")
```

```
    k += 1
```

b. a = int(input())

```
b = int(input())
```

```
c = int(input())
```

```
s = a+b+c
```

```
print(c)
```

c. a = int(input())

```
if a > 0:
```

```
    print(a)
```

```
else:
```

```
    print(a)
```

12. Создатель языка программирования Python

a. Гвидо Ван Россум

b. Дэвид Паттерсон

c. Эрвин Дональд Кнут

d. Джеймс Артур Гослинг

13. На каких операционных системах может работать Python?

a. Windows

b. Linux

c. macOS

d. Ничего из этого

14. Что делает функция len()?

a. Возвращает длину строки

c. Возвращает номер символа

b. Возвращает случайное число

d. Возвращает модуль числа

15. Где находятся параметры, а где аргументы функции?

a. Параметры пишутся при объявлении функции, аргументы при вызове

b. Аргументы пишутся при объявлении функции, параметры при вызове

c. Это одно и то же!

d. У функции есть только параметры

16. Ниже представлены утверждения о списках. Какие из них верны?

a. Один и тот же объект может появляться в списке несколько раз

b. Размеры списка четко не определены

c. Эти два списка одинаковы:

```
['a', 'b', 'c']
```


['c', 'a', 'b']

d. Все элементы в списке должны быть одного типа

17. Допустим, объявлен список — `a = ['foo', 'bar', 'baz', 'qux', 'quux', 'corge']`.

Ниже представлены несколько программ. В каких из них вывод указан правильно?

a. `print(a[4::-2])`

['quux', 'baz', 'foo']

c. `a[:]` is a

b. `max(a[2:4] + ['gault'])`

'qux'

d. `print(a[-5:-3])`

['bar', 'baz']

18. Объявлен список — `a = [1, 2, 3, 4, 5]`.

Ниже представлены строки кода, удаляющие элемент. Какие из них в результате дадут список [1, 2, 4, 5]?

a. `del a[2]`

b. `a[2:2] = []`

c. `a[2:3] = []`

d. `a.remove(3)`

e. `a[2] = []`

19. Что такое кортеж?

a. Неупорядоченная неизменяемая коллекция объектов произвольных типов

b. Упорядоченная изменяемая коллекция объектов произвольных типов

c. Упорядоченная неизменяемая коллекция объектов произвольных типов

d. Упорядоченная неизменяемая коллекция объектов одного типа

20. Что из следующего ниже верно о кортежах?

a. Кортежи можно складывать

b. Кортеж занимает больше памяти чем список

c. Кортеж может быть ключом словаря

d. У кортежа есть метод `append()`

e. Кортеж занимает меньше памяти чем список

21. Выберите верные утверждения:

a. Словари изменяемы

b. Словари могут быть любой «глубины»

c. Словарь может содержать объект любого типа, кроме другого словаря

d. Доступ к элементам словаря осуществляется с помощью ключа

22. Ниже представлены несколько вариантов кода. Какой из них удалит элемент с ключом 'baz' из словаря?

a. `del d['baz']`

b. `d['baz']`

c. `del.d(baz)`

d. `del d(baz)`

23. Начинаем с простого. Что такое множество в Python?

a. Это любая коллекция элементов

b. Это список, содержащий в себе только функции

4.2 Интернет-ресурсы

1. Studref - Студенческие реферативные статьи и материалы (info{at}studref.com) © 2017 - 2021
2. nsportal.ru - Образовательная социальная сеть.
3. Основы программирования на Python. Учебное пособие для СПО.-Электронная книга...Дата выхода на ЛитРес: 01 сентября 2021
Дата написания: 2021,Объем: 287 стр
4. Korilkaurokov.ru - сайт для учителей.

4.3 Материально-техническое обеспечение дисциплины

МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ) Специальные помещения представляют собой учебные аудитории для проведения всех занятий по дисциплине, предусмотренных учебным планом и содержанием РПД. Помещения укомплектованы специализированной мебелью и техническими средствами обучения согласно требованиям. Помещение для проведения лекционных занятий укомплектованы специализированной мебелью и техническими средствами для представления учебной информации студентам, включая проекционное оборудование и интерактивную доску; помещения для проведения лабораторных работ укомплектованы компьютерным оборудованием, столами, стульями.