

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«Кубанский государственный университет»

Факультет компьютерных технологий и прикладной математики
Кафедра вычислительных технологий



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ Б1.В.11 «ВЕРИФИКАЦИЯ ПРОГРАММНЫХ СИСТЕМ»

Направление
подготовки/специальность 02.03.02 Фундаментальная информатика и
информационные технологии
(код и наименование направления подготовки/специальности)

Направленность (профиль) /
специализация Математическое и программное обеспечение компьютерных
технологий
(наименование направленности (профиля) специализации)

Программа подготовки академический бакалавриат
(академическая /прикладная)

Форма обучения очная
(очная, очно-заочная, заочная)

Квалификация выпускника бакалавр
(бакалавр, магистр, специалист)

Краснодар 2021

Рабочая программа дисциплины Б1.В.11 «ВЕРИФИКАЦИЯ ПРОГРАММНЫХ СИСТЕМ» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии.

Программу составил(а):

Жуков Сергей Александрович, доцент, к. ф.-м. н., доцент

Ф.И.О. ,должность, ученая степень, ученое звание

_____  _____
подпись

Рабочая программа дисциплины Б1.В.11 «ВЕРИФИКАЦИЯ ПРОГРАММНЫХ СИСТЕМ» утверждена на заседании кафедры Вычислительных Технологий протокол №6 от «20» мая 2021 г.

Заведующий кафедрой (разработчика) Вишняков Ю. М.

фамилия, инициалы

 _____
подпись

Утверждена на заседании учебно-методической комиссии факультета Компьютерных Технологий и Прикладной Математики протокол № 1 от «21» мая 2021 г

Председатель УМК факультета Коваленко А.В.

фамилия, инициалы

 _____
подпись

Рецензенты:

Гаркуша О.В., доцент кафедры информационных технологий ФБГОУ ВО «Кубанский государственный университет», кандидат физико-математических наук.

Схаляхо Ч.А., доцент КВВУ им.С.М.Штеменко, к.ф.-м.н., доцент

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1 Цели освоения дисциплины

Целью преподавания и изучения дисциплины «Верификация программных систем» является формирование у студентов способности понимать и составлять функционально-логические спецификации создаваемых программ и, а также знать основные методы проверки правильности таких программ.

1.2 Задачи дисциплины

Студент должен **знать** основные понятия, подходы и методы спецификации программных систем, методы и технологии верификации программных систем; **уметь** применять базовые методы верификации; **владеть** технологиями, способствующими верификации программных систем.

1.3 Место дисциплины в образовательной программе

Дисциплина «Верификация программных систем» относится к базовой части блока Б1 Дисциплины (модули).

Для изучения дисциплины необходимо знание основ алгебры, дискретной математики, теории алгоритмов и вычислительных процессов. Знания, получаемые при изучении верификации программных систем, используются при изучении других дисциплин профессионального цикла учебного плана бакалавра (“Современные концепции программирования”, “Облачные вычисления”, “Разработка технической документации”), а также при работе над выпускной работой.

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

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих **профессиональных компетенций**:

№ п.п	Индекс компетенции	Содержание компетенции (или ее части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	владеть

1.	ПК-3	Способен приобретать и использовать организационно-управленческие навыки в конкретной профессиональной и социальной деятельности; разрабатывать, реализовывать и управлять процессами жизненного цикла программных продуктов	фундаментальные концепции и подходы в области создания надежных программных систем, а также знания, стандарты качества программных систем	использовать углубленные теоретические и практические знания в области информационных технологий и прикладной логики, фундаментальные концепции и средства спецификации поведения сложных систем	методами верификации практических решений в области информационных технологий, инструментами проверки правильности архитектуры информационных систем
2.	ПК-5	Способен применять в профессиональной	фундаментальные концепции и	использовать углубленные	методами верификации

		деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии	подходы в области создания надежных программных систем, а также знания, стандарты качества программных систем	теоретические и практические знания в области информационных технологий и прикладной логики, фундаментальные концепции и средства специфицирования поведения сложных систем	практических решений в области информационных технологий, инструментами проверки правильности архитектуры информационных систем
--	--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 4 зач.ед. (144 часа), их распределение по видам работ представлено в таблице.

Вид учебной работы	Всего часов	Семестры (часы)			
		7			
Аудиторные занятия (всего)	72,5	72,5			
В том числе:					
Занятия лекционного типа	34	34			
Лабораторные занятия	34	34			
КСР	4	4			
ИКР	0,5	0,5			
Самостоятельная работа (всего)	35,8	35,8			
В том числе:					
Проработка учебного (теоретического) материала	35,8	35,8			
Контроль	35,7	35,7			
Промежуточная аттестации	зачет, экзамен	зачет, экзамен			
Общая трудоемкость	час	144	144		
зач. ед.		4	4		

2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины. Разделы дисциплины, изучаемые в 7 семестре (очная форма).

№ Раздела	Наименование раздела	Количество часов	
		Аудиторная работа	Самостоятельная работа

		Всего	Лек	ЛР	КСР	ИКР	СРС	Контроль
--	--	-------	-----	----	-----	-----	-----	----------

1	Анализ понятия надежности программной системы	20	4	4			6	6
2	Прикладная теория логической правильности программы	54,2	12	12	2	0,2	14	14
3	Функциональный подход к спецификации программы	27,7	8	8			6	5,7
4	Подход Дейкстры по построению программы из доказательства ее правильности	24	6	6	2		5	5
5	Средства спецификации программ	18,1	4	4		0,3	4,8	5
6	Всего	144	34	34	4	0,5	35,8	35,7

2.3 Содержание разделов дисциплины

2.3.1 Занятия лекционного типа

№ раздела	Наименование раздела	Содержание раздела	Форма текущего контроля
1	2	3	4
1	Логические основы правильности программ	Высказывания. Аксиоматическая теория высказываний. Исчисление предикатов. Примеры прикладных теорий.	ЛР
2	Прикладная теория логической правильности программы	Подход Флойда к верификации алгоритмов. Частичная и тотальная правильность программы. Структурированные программы. Нотация и аксиоматика Хоара для доказательства частичной правильности. Эвристики построения инвариантов. Метод потенциальных функций для доказательства тотальной правильности.	ЛР
3	Верификация блок-схем по методу Флойда	Формальное определение блок-схемы. Логическая спецификация операторов и путей вычисления в блок-схеме.	ЛР
4	Аксиоматика тотальной правильности структурированных программ	Уточненная нотация Хоара для утверждений о тотальной правильности программы. Аксиоматика тотальной правильности в расширенной нотации Хоара. Подход к доказательству правильности на основе слабейшего предусловия. Свойства слабейшего предусловия. Аксиоматика Дейкстры для преобразователей предикатов.	ЛР
5	Средства спецификации программ	Язык спецификаций Z. Метод VDM и примеры его использования.	ЛР

2.3.2 Занятия семинарского типа

Учебным планом не предусмотрены.

2.3.3 Лабораторные занятия

№ работы	Наименование лабораторных работ	Форма текущего контроля
1	Анализ высказываний и построение высказываний	ЛР
2	Аксиоматика пропозиционального исчисления	ЛР
3	Построение предикатов и утверждений в предметных областях	ЛР
4	Анализ структуры предикатов и их преобразования	ЛР
5	Аксиоматика исчисления предикатов	ЛР
6	Построение логических спецификаций программ	ЛР
7	Доказательство частичной правильности линейных программ	ЛР
8	Доказательство частичной правильности программ условиями	ЛР
9	Доказательство частичной правильности программ циклами	ЛР
10	Доказательство частичной правильности сложно структурированных программ	ЛР
11	Построение потенциальных функций для циклических программ	ЛР
12	Примеры доказательств тотальной правильности программ	ЛР
13	Обзор отчетов по использованию методов дедуктивной верификации при построении программных систем, протоколов в различных предметных областях	ЛР
14	Построение слабейшего предусловия для линейных программ по заданному постусловию	ЛР
15	Построение слабейшего предусловия для программ с ветвлением по заданному постусловию	ЛР
16	Построение инвариантов циклов на основе анализа постусловия для программы	ЛР
17	Описание программных спецификаций средствами языка Z	ЛР
18	Обзор примеров программных спецификаций в VDM	ЛР

2.3.4 Самостоятельное изучение разделов дисциплины

Раздел 1. Язык логики предикатов. Выполнимость формул логики предикатов. Подход к тестированию программ и виды тестирования.

Раздел 2. Базовые понятия аксиоматической теории. Структурные программы. Правила вывода в аксиоматике частичной правильности Хоара. Подходы к построению потенциальных функций.

Раздел 3. Методы построения индуктивных утверждений. Подходы и методы анализа потоков управления в программах. Методики и средства аннотирования

программ. Обзор и анализ практики использования дедуктивных методов верификации в программных проектах.

Раздел 4. Составление программ в нотации Дейкстры. Построение слабейшего предусловия по заданной программе в нотации Дейкстры. Методика синтеза программы на основе анализа доказательства истинности его слабейшего предусловия.

Раздел 5. Анализ средств языка Z и его использование для структурированных программ. Использование методики VDM для построения программных спецификаций.

2.3.4 Примерная тематика курсовых работ (проектов)

Учебным планом не предусмотрены.

3. Образовательные технологии

При проведении занятий по дисциплине используются следующие образовательные технологии:

- технология разноуровневого обучения (дифференцированное обучение);
- технология коллективного взаимодействия (организованный диалог, коллективный способ обучения).

Технология адаптивного обучения (индивидуализированное обучение).

Семестр	Вид занятия (Л, ПР, ЛР)	Используемые интерактивные образовательные технологии	Количество часов
7	Л	Компьютерные презентации и обсуждение	34
	ЛР	Разбор конкретных ситуаций (задач) с использованием штатного ПО, выполнение тестов на знание терминологии, сведений из области верификации программных систем, программирование и аннотирование алгоритмов	34
Итого:			68

4. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

4.1 Фонд оценочных средств для проведения текущего контроля

Фонд оценочных средств дисциплины состоит из средств текущего контроля выполнения заданий, лабораторных работ, средств итоговой аттестации (зачет и экзамен в 7 семестре).

Оценка успеваемости осуществляется по результатам:

- выполнения лабораторных работ;
- ответов на теоретические вопросы при сдаче лабораторных работ;
- ответа на зачете и экзамене (для выявления знания и понимания теоретического материала дисциплины и практических навыков).

4.1.1 Примеры типовых заданий

1. Доказать частичную правильность программы;
{ $n > 0$ }
 $i := 0; S := 0; p := 1;$
while $i \leq n$ do begin $S := S + p; p := 2 * p; i := i + 1$ end; $S := S + 1$

$$\{ S = 2^n \}$$

2. Доказать частичную правильность программы:

```
{ n > 0 } i := 1; S := 0; while i ≤ n do begin S := S + i3; i := i + 1 end
```

$$\{ S = (n^2 * (n + 1)^2) / 4 \}$$

3. Доказать завершимость программы определением потенциальных функций: для любых натуральных x, y, n

```
while n > 0 do begin
  while x ≤ y do x := x - 1;
  1;
```

```
    if y < n then n := n - 1 else while y ≤ x do y := y - 1
  end;
```

4. Доказать завершимость программы определением потенциальных функций: для любых натуральных x, y, n

```
while n > 0 do begin
  while x ≤ n / 2 do x := x - 1;
  while y ≤ n / 2 do y := y - 1;
  while n ≤ x + y do n := n - 1;
  1;
```

```
end;
```

4.2 Фонд оценочных средств для проведения промежуточной аттестации

4.2.1 Перечень вопросов к зачету

1. Сформулировать определение частичной и тотальной правильности программы.
2. Какова связь между частичной правильностью и тотальной правильностью программ.
3. В чем смысл тройки Хоара? Сформулировать аксиомы и правила вывода для доказательства частичной правильности.
4. Сформулировать назначение и свойства потенциальной функции.
5. Сформулировать основные шаги в доказательстве правильности программы методом Флойда.
6. Сформулировать основные свойства слабейшего предусловия.
7. Правила вывода преобразователей предикатов для последовательной и условной композиции.
8. Правила вывода преобразователя предикатов для циклической композиции.
9. Основные элементы нотации языка спецификаций Z.
10. Основные составляющие методики VDM для спецификации программной системы.

4.2.2 Перечень задач к зачету

1. Пусть есть два предиката $A(x)$ и $B(x)$, на множестве натуральных чисел N . Записать на языке исчисления предикатов, что множество элементов, на которых выполняются оба предиката одновременно, является бесконечным.

2. Для предиката $A(x)$, от одной свободной индивидуальной переменной x , принимающей натуральные значения, сформулировать утверждение, что $A(x)$ справедлив для всех x , за исключением, возможно, конечного множества элементов.
3. Определить формальную спецификацию для программы, которая по трем входным вещественным значениям, таким, что первое не равно нулю, вычисляет три

результатирующих значения, первые два из которых являются корнями квадратного уравнения, в котором коэффициентами при старшем члене, линейном члене и свободном члене являются эти три входных значения, а третье результирующее значение равно true, если это квадратное уравнение имеет корни, и оно равно false, если квадратное уравнение корней не имеет.

4. Определить формальную спецификацию для программы, которая принимает в качестве входных данных одномерный числовой массив из n элементов (n – натуральное число, не превышающее сотни) и выполняет его сортировку по неубыванию. Результат программы – отсортированный таким образом массив.

5. Доказать частичную правильность программы:

```
{ x ≥ 0 & y ≥ 0 & ( x , y – целые числа )
  } z := 0; u := x; v := y;

  while u > 0 do begin if odd(u) then z := z + v; u := u div 2; v := 2*v; end;

  { z = x * y }
```

6. Доказать частичную правильность программы:

```
{ n > 0 }
  i := 2; found := false;
  while ( i < n ) and not found do begin
    p := 1; while p < n do p := p * i; if n = p then found := true;
    i := i + 1
  end
  { found ⇐ n – степень некоторого числа }
```

4.2.3 Перечень вопросов к экзамену

1. Смысл программной верификации. Отличие верификации программы от ее тестирования. Недостатки тестирования. Место верификации и тестирования в жизненном цикле программы.
2. Определение частичной и тотальной правильности программ. Необходимость выделения частичной правильности.
3. Смысл спецификации программы, состав спецификации. Сохранение частичной и тотальной правильности программы при замене входного предиката на более сильный, а выходного предиката на более слабый.
4. Смысл спецификации программы, состав спецификации. Связь между конъюнкциями/дизъюнкциями утверждений о частичной/тотальной правильности программы и конъюнкциями/дизъюнкциями постусловий в утверждениях о частичной/тотальной правильности программы.
5. Связь между тотальной и частичной правильностью. Проблема разрешения частичной правильности программ.
6. Метод индуктивных утверждений Флойда: назначение метода, объект его применения, точки сечения, классификация базовых путей, смысл индуктивного утверждения.

7. Метод индуктивных утверждений Флойда: смысл предиката допустимости пути и функции пути, их примеры. Метод построения предикта допустимости пути и функции пути.

8. Метод индуктивных утверждений Флойда: смысл и вид условий верификации. Пример условия верификации.
9. Метод индуктивных утверждений Флойда: общая схема выполнения метода. Способ доказательства завершимости программы.
10. Нотация Хоара для утверждений о частичной правильности. Аксиома и базовые правила вывода для утверждений о частичной правильности.
11. Производные правила вывода в аксиоматике частичной правильности.
12. Пример доказательства частичной правильности. Обоснование справедливости производного правила базовой операции через базовую аксиоматику (аксиому и базовые правила вывода).
13. Эвристики ослабления постулов (теория воздушного шарика). Примеры использования.
14. Подход к доказательству завершимости программы с помощью потенциальных функций. Пример программы и обоснование ее завершимости с помощью потенциальных функций.
15. Аксиоматика тотальной правильности.

16. Основные шаги в доказательстве правильности программы методом Флойда.
17. Классификация путей выполнения блок-схемы в методе Флойда.
18. Способ построения предиката выполнимости пути и функции пути в методе Флойда.
19. Условный язык описания программы в методе Дейкстры.
20. Основные свойства слабейшего предусловия.
21. Правила вывода преобразователей предикатов для последовательной и условной композиции.
22. Правила вывода преобразователя предикатов для циклической композиции.
23. Основные элементы нотации языка спецификаций Z.
24. Основные составляющие методики VDM для спецификации программной системы.

4.2.4 Критерии оценивания к зачету

Оценка “зачтено” - практические задания выполнены в срок в объеме не менее 80%. Студент демонстрирует правильные, уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при аргументации ответов на вопросы при защите лабораторных.

Оценка «не зачтено» -практические задания не выполнены либо предоставлены не в срок в объеме менее 60%, Студент демонстрирует наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электробрамного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4.2.5 Образцы билетов

Билет №1

1. Смысл спецификации программы, состав спецификации. Сохранение частичной правильности и тотальной правильности программ при замене входного предиката на более сильный, а выходного предиката на более слабый.
2. Аксиоматика тотальной правильности.
3. Доказать частичную правильность программы:

$$\{ n > 0 \} \quad i := 1; S := 0; \text{ while } i \leq n \text{ do begin } S := S + i \quad ; i := i + 1 \text{ end}$$

$$\{ S = (n * (n + 1)) / 4 \}$$

Билет №2

1. Смысл программной верификации. Отличие верификации программы от ее тестирования. Недостатки тестирования. Место верификации и тестирования в жизненном цикле программы.
2. Производные правила вывода в аксиоматике частичной правильности.
3. Доказать частичную правильность программы:

$$\{ x \geq 0 \ \& \ y \geq 0 \ \& \ (x, y - \text{целые числа})$$

$$\} z := 0; u := x; v := y;$$

$$\text{ while } u > 0 \text{ do begin if odd}(u) \text{ then } z := z + v; u := u \text{ div } 2; v := 2 * v; \text{ end;}$$

$$\{ z = x * y \}$$

4.2.6 Критерии оценивания к экзамену

Оценка «отлично»: точные формулировки алгоритмов, теорем и правильные доказательства; точные определения математических объектов и ясные и правильные определения объектов, характеризующихся неформализованными понятиями.

Оценка «хорошо»: при ответе на один вопрос даны точные формулировки алгоритмов, теорем и правильные доказательства; точные определения математических объектов и ясные и правильные определения объектов, характеризующихся

неформализованными понятиями; при ответе на второй вопрос имеются неточности формулировки алгоритмов, теорем или пробелы в правильных доказательствах; недостаточно точные определения математических объектов или неясные и не совсем правильные определения объектов, характеризующихся неформализованными понятиями.

Оценка «удовлетворительно»: при ответе на оба вопроса имеются неточности формулировки алгоритмов, теорем или пробелы в правильных доказательствах; недостаточно точные определения математических объектов или неясные и не совсем правильные определения объектов, характеризующихся неформализованными понятиями.

Оценка «неудовлетворительно»: отсутствует ответ хотя бы на один из вопросов или имеются существенные неточности в формулировках алгоритмов, теорем, приведены неправильные доказательства; неверные определения математических объектов и неправильные определения объектов, характеризующихся неформализованными понятиями.

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

5.1 Основная литература

1. Гурин Р.Е., Рудаков И.В., Ребриков А.В. Методы верификации программного обеспечения. – Наука и образование. МГТУ им. Баумана. Электрон. журнал 2015, № 10, с. 235-251.

5.2 Дополнительная литература

2. Дейкстра Э. Дисциплина программирования. – М.: Мир, 1978.– 276 с.
3. Грис Д. Наука программирования. – М.: Мир, 1984.– 416 с
4. Андерсон Р. Доказательство правильности программ. – М.: Мир, 1982.– 165 с.
5. Алагич С, Арбиб М. Проектирование корректных структурированных программ. – М.: Радио и связь, 1984.– 265 с.

6. Методические указания для обучающихся по освоению дисциплины (модуля)

По курсу предусмотрено проведение лекционных занятий, на которых дается основной систематизированный материал, лабораторных работ, контрольной работы, зачета и экзамена.

Важнейшим этапом курса является самостоятельная работа по дисциплине с использованием указанных литературных источников и методических указаний автора курса.

Виды и формы СР, сроки выполнения, формы контроля приведены выше в данном документе.

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

7. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю)

7.1 Перечень информационных технологий

- Проверка домашних заданий и консультирование посредством электронной почты.
- Использование электронных презентаций при проведении лекций и практических

занятий.

7.2 Перечень необходимого программного обеспечения

1. Программы для демонстрации и создания презентаций («MicrosoftPowerPoint»).

7.3 Перечень информационных справочных систем:

1. Электронная библиотечная система eLIBRARY.RU (<http://www.elibrary.ru/>)

8. Методические указания по выполнению лабораторных работ

Лабораторные работы выполняются, как правило, в компьютерном классе. Отдельные работы могут выполняться в аудитории при наличии у студентов портативных компьютеров. На лабораторных занятиях осуществляется проработка и закрепление методов и инструментария для верификации программ учебного характера.

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине (модулю)

№	Вид работ	Материально-техническое обеспечение дисциплины(модуля) и оснащенность
1.	Лекционные занятия	Лекционная аудитория, оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук) и соответствующим программным обеспечением (ПО)PowerPoint. ауд. 129, 131, А305.
2.	Лабораторные занятия	Лаборатория, укомплектованная специализированными техническими средствами обучения – компьютерный класс, с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета (лаб. 102-106.).
3.	Групповые (индивидуальные) консультации	Аудитория, (кабинет) – компьютерный класс
4.	Текущий контроль, промежуточная аттестация	Аудитория, приспособленная для письменного ответа при промежуточной аттестации.
5.	Самостоятельная работа	Кабинет для самостоятельной работы, оснащенный компьютерной техникой с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета.