



Федеральное государственное бюджетное образовательное учреждение
высшего образования

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
АРХИТЕКТУРНО-СТРОИТЕЛЬНЫЙ УНИВЕРСИТЕТ

Кафедра Информационных технологий

УТВЕРЖДАЮ
Начальник учебно-методического управления

«22» февраля 2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Технологии программирования

направление подготовки/специальность 09.04.03 Прикладная информатика

направленность (профиль)/специализация образовательной программы Разработка программного обеспечения для решения задач в сфере строительства

Форма обучения очная

1. Цели и задачи освоения дисциплины (модуля)

Обучение студентов систематизированному представлению о принципах построения и проектирования программных систем.

- познакомить обучающихся с основными понятиями и определениями, с классификацией программного обеспечения;
- дать представление об этапах создания программного продукта в рамках жизненного цикла, о современном состоянии технологий разработки программного продукта;
- познакомить обучающихся с существующими подходами к оценке качества процессов создания программного обеспечения ;
- дать обучающемуся практические навыки проектирования программного обеспечения и расчета его надежности.

2. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине, обеспечивающие достижение планируемых результатов освоения ОПОП
ОПК-1 Способен самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально- экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте	ОПК-1.3 Осуществляет выбор методов решения отдельных элементов нестандартной задачи	знает методы решения различных элементов нестандартных задач умеет уметь корректно подбирать методы решения для решения нестандартной задачи владеет навыками навыками применения методов для решения нестандартных задач
ОПК-1 Способен самостоятельно приобретать, развивать и применять математические, естественнонаучные, социально- экономические и профессиональные знания для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте	ОПК-1.5 Представляет результат решения отдельных элементов нестандартной задачи и задачи в целом	знает общие процедуры для решения нестандартных задач и представления результатов их решений умеет представлять результат решения отдельных элементов нестандартной задачи и задачи в целом владеет навыками приемами представления решений различных типов задач, в том числе нестандартных

ОПК-2 Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач	ОПК-2.1 Разрабатывает алгоритм решения задачи в соответствии с техническим заданием	знает методы алгоритмизации для написания корректных алгоритмов умеет корректно подбирать метод алгоритмизации опираясь на техническое задание владеет навыками навыками применения различных методов алгоритмизация
ОПК-5 Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем	ОПК-5.1 Разрабатывает программное или аппаратное обеспечение информационных или автоматизированных систем в соответствии с техническим заданием	знает современное программное и аппаратное обеспечение информационных и автоматизированных систем умеет разрабатывать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач в соответствии с техническим заданием владеет навыками методами разработки программного и аппаратного обеспечения информационных и автоматизированных систем для решения профессиональных задач в соответствии с техническим заданием
ОПК-5 Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем	ОПК-5.2 Модернизирует программное или аппаратное обеспечение информационных или автоматизированных систем в соответствии с техническим заданием	знает современное программное и аппаратное обеспечение информационных и автоматизированных систем умеет модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач в соответствии с техническим заданием владеет навыками методами модернизации программного и аппаратного обеспечения информационных и автоматизированных систем для решения профессиональных задач в соответствии с техническим заданием

3. Указание места дисциплины (модуля) в структуре образовательной программы

Данная дисциплина (модуль) включена в Блок «Дисциплины, модули» Б1.О.07 основной профессиональной образовательной программы 09.04.03 Прикладная информатика и относится к обязательной части учебного плана.

Знать:

- методики сбора и обработки информации;
- актуальные российские и зарубежные источники информации в сфере профессиональной деятельности;

-основные принципы структурного написания программ, конструкции языка высокого уровня и технологию создания программ, базовые средства языка и средства стандартных библиотек, технологию разработки алгоритмов и программ, методы отладки и решения задач на ЭВМ в различных режимах;

- методы и средства алгоритмизации и программирования.

Уметь:

- применять методики поиска, сбора и обработки информации;
- осуществлять критический анализ и синтез информации, полученной из разных источников;
- определять требования к создаваемой программе, выбирать среду программирования, выбирать или разрабатывать алгоритм решения задачи, реализовывать

программный код, выполнять отладку и проводить тестирование программы, ставить задачу и разрабатывать

алгоритм ее решения, использовать прикладные системы

программирования, разрабатывать основные программные документы, работать с современными системами программирования, включая объектноориентированные.

Владеть :

- методами поиска, сбора и обработки, критического анализа и синтеза информации;
- методикой системного подхода для решения поставленных задач;

методикой применения средств языка высокого уровня для создания программного обеспечения, языками процедурного и объектно-ориентированного программирования, навыками разработки и отладки программ не менее чем на одном из алгоритмических языков программирования высокого уровня.

№ п/п	Последующие дисциплины	Код и наименование индикатора достижения компетенции
1	Программирование в 1С	ОПК-6.1, ОПК-6.2
2	Прикладные информационные технологии	ОПК-1.1, ОПК-1.2, ОПК-6.1, ОПК - 6.2, ОПК-4.1
3	Выполнение и защита выпускной квалификационной работы	УК-1.1, УК-1.2, УК-1.3, УК-1.4, УК-1.5, УК-1.6, УК-2.1, УК-2.2, УК-2.3, УК-2.4, УК-2.5, УК-3.1, УК-3.2, УК-3.3, УК-3.4, УК-4.1, УК-4.2, УК-4.3, УК-4.4, УК-5.1, УК-5.2, УК-5.3, УК-6.1, УК-6.2, УК-6.3, ОПК-1.1, ОПК-1.2, ОПК- 1.3, ОПК-1.4, ОПК-1.5, ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-3.1, ОПК- 3.2, ОПК-3.3, ОПК-4.1, ОПК-4.2, ОПК-4.3, ОПК-5.1, ОПК-5.2, ОПК- 6.1, ОПК-6.2, ОПК-7.1, ОПК-7.2, ОПК-8.1, ОПК-8.2, ОПК-8.3, ПК- 1.1, ПК-1.2, ПК-1.3, ПК-2.1, ПК- 2.2, ПК-2.3, ПК-3.1, ПК-3.2, ПК-4.1, ПК-4.2, ПК-4.3, ПК(Ц)-1.1, ПК(Ц)-1.2, ПК(Ц)-1.3, ПК(Ц)-1.4, ПК(Ц)-1.5, ПК(Ц)-1.6

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

Вид учебной работы	Всего часов	Из них часы на практическую подготовку	Семестр
			1
Контактная работа	32		32
Практические занятия (Пр)	32	0	32
Иная контактная работа, в том числе:			
консультации по курсовой работе (проекту), контрольным работам (РГР)			
контактная работа на аттестацию (сдача зачета, зачета с оценкой; защита курсовой работы (проекта); сдача контрольных работ (РГР))			
контактная работа на аттестацию в сессию (консультация перед экзаменом и сдача			
Часы на контроль	4		4
Самостоятельная работа (СР)	72		72
Общая трудоемкость дисциплины (модуля)			
часы:	108		108
зачетные единицы:	3		3

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

5.1. Тематический план дисциплины (модуля)

№	Разделы дисциплины	Семестр	Контактная работа (по учебным занятиям), час.						СР	Всего, час.	Код индикатора достижения компетенции
			лекции		ПЗ		ЛР				
			всего	из них на практическую подготовку	всего	из них на практическую подготовку	всего	из них на практическую подготовку			
1.	1 раздел. Технологии программирования										
1.1.	Основы программирования в среде C#	1			8				18	26	ОПК-1.3, ОПК-1.5, ОПК-2.1, ОПК-5.1, ОПК-5.2
1.2.	Основные понятия и подходы. Приемы обеспечения технологичности программных продуктов.	1			4				9	13	ОПК-1.3, ОПК-1.5, ОПК-2.1, ОПК-5.1, ОПК-5.2
1.3.	Проектирование программного обеспечения при структурном подходе к программированию	1			4				9	13	ОПК-1.3, ОПК-1.5, ОПК-2.1, ОПК-5.1, ОПК-5.2

1.4.	Тестирование и отладка программных продуктов при структурном подходе	1			4			9	13	ОПК-1.3, ОПК-1.5, ОПК-2.1, ОПК-5.1, ОПК-5.2
1.5.	Проектирование программного обеспечения при объектно-ориентированном подходе к программированию	1			4			9	13	ОПК-1.3, ОПК-1.5, ОПК-2.1, ОПК-5.1, ОПК-5.2
1.6.	Разработка пользовательских интерфейсов. Оценка качества программного обеспечения	1			4			9	13	ОПК-1.3, ОПК-1.5, ОПК-2.1, ОПК-5.1, ОПК-5.2
1.7.	Оценка качества программного обеспечения	1			4			9	13	ОПК-1.3, ОПК-1.5, ОПК-2.1, ОПК-5.1, ОПК-5.2
2.	2 раздел. Контроль									
2.1.	Зачет	1							4	ОПК-1.3, ОПК-1.5, ОПК-2.1, ОПК-5.1, ОПК-5.2

5.1. Практические занятия

№ разд	Наименование раздела и темы практических занятий	Наименование и содержание практических занятий
1	Основы программирования в среде C#	Основы программирования в среде C# Линейные программы. Циклические программы. Одномерные массивы. Двумерные массивы.
2	Основные понятия и подходы. Приемы обеспечения технологичности программных продуктов.	Основные понятия и подходы. Приемы обеспечения технологичности программных продуктов. Работа со строками класса String Работа со строками класса StringBuilder Регулярные выражения Оценка характеристик разработанных программ
3	Проектирование программного обеспечения при структурном подходе к программированию	Проектирование программного обеспечения при структурном подходе к программированию Работа с файлами. Оценка степени отлаженности разрабатываемых приложений
4	Тестирование и отладка программных продуктов при структурном подходе	Тестирование и отладка программных продуктов при структурном подходе Выполнить ручное тестирование разработанных приложений Выполнить структурное тестирование разработанных приложений Выполнить функциональное тестирование разработанных приложений

5	Проектирование программного обеспечения при объектно-ориентированном подходе к программированию	Проектирование программного обеспечения при объектно-ориентированном подходе к программированию Работа с классами в C# Наследование Разработка UML - диаграмм
6	Разработка пользовательских интерфейсов. Оценка качества программного обеспечения	Разработка пользовательских интерфейсов. Оценка качества программного обеспечения Разработать объектно-ориентированное, многооконное приложение с использованием интерфейсов, делегатов и событий. Разработка UML – диаграмм Обосновать тип пользовательского интерфейса и форму диалога Разработать граф диалога пользователя Разработать пользовательское меню
7	Оценка качества программного обеспечения	Оценка качества программного обеспечения Выполнить оценку качества разработанного приложения по ГОСТ 28195-89

5.2. Самостоятельная работа обучающихся

№ разд	Наименование раздела дисциплины и темы	Содержание самостоятельной работы
1	Основы программирования в среде C#	Основы программирования в среде C# Подготовка к тестированию Оформление и подготовка к защите практической работы
2	Основные понятия и подходы. Приемы обеспечения технологичности программных продуктов.	Основные понятия и подходы. Приемы обеспечения технологичности программных продуктов. Подготовка к тестированию Оформление и подготовка к защите практической работы
3	Проектирование программного обеспечения при структурном подходе к программированию	Проектирование программного обеспечения при структурном подходе к программированию Подготовка к тестированию Оформление и подготовка к защите практической работы
4	Тестирование и отладка программных продуктов при структурном подходе	Тестирование и отладка программных продуктов при структурном подходе Подготовка к тестированию Оформление и подготовка к защите практической работы
5	Проектирование программного обеспечения при объектно-ориентированном подходе к программированию	Проектирование программного обеспечения при объектно-ориентированном подходе к программированию Подготовка к тестированию Оформление и подготовка к защите практической работы
6	Разработка пользовательских интерфейсов. Оценка качества программного обеспечения	Разработка пользовательских интерфейсов. Оценка качества программного обеспечения Подготовка к тестированию Оформление и подготовка к защите практической работы

7	Оценка качества программного обеспечения	Оценка качества программного обеспечения Подготовка к тестированию Оформление и подготовка к защите практической работы
---	--	---

6. Методические материалы для самостоятельной работы обучающихся по дисциплине (модулю)

Программой дисциплины предусмотрено проведение практических занятий, предполагающих закрепление изученного материала и формирование у обучающихся необходимых знаний, умений и навыков. Кроме того, важнейшим этапом изучения дисциплины является самостоятельная работа обучающихся с использованием всех средств и возможностей современных образовательных технологий.

В объем самостоятельной работы по дисциплине включается следующее:

- изучение теоретических вопросов по всем темам дисциплины;
- подготовка к практическим занятиям;
- подготовка к текущему контролю успеваемости студентов;
- подготовка к зачету.

Залогом успешного освоения дисциплины является обязательное посещение практических занятий, так как пропуск одного (тем более, нескольких) занятий может осложнить освоение разделов курса. На практических занятиях материал, изложенный на лекциях, закрепляется при выполнении практических заданий.

Приступая к изучению дисциплины, необходимо в первую очередь ознакомиться с содержанием РПД, а также методическими указаниями по организации самостоятельной работы.

При подготовке к практическим занятиям и в рамках самостоятельной работы по изучению дисциплины обучающимся необходимо:

- повторить законспектированный на лекционном занятии материал и дополнить его с учетом рекомендованной по данной теме литературы;
- при самостоятельном изучении теоретической темы сделать конспект, используя рекомендованные в РПД источники;
- выполнить практические задания в рамках изучаемой темы;
- ответить на контрольные вопросы по теме, используя материалы ФОС;
- подготовиться к тестированию, предусмотренному в контрольных точках;
- подготовиться к промежуточной аттестации.

При написании программ рекомендуется вначале написать программный код самостоятельно, а затем сравнить его с каким-либо примером. Такой подход позволяет студентам быстрее освоить методы, способы и стиль написания программ на языке C# и сократить время на его изучение.

Работы, выполняемые на практических занятиях, сдаются только лично на занятиях преподавателю, который ведет группу. Задания, выполняемые на компьютере, студенты сначала показывают только в электронном виде. При необходимости, в присутствии преподавателя дodelывают или исправляют ошибки. Если требуется распечатать выполненные работы и сдать их в бумажном виде, преподаватель говорит об этом на занятиях.

Итогом изучения дисциплины являются контрольная работа и зачет. Зачет проводится по расписанию. Форма проведения занятия может быть устная, письменная и в электронном виде. Студенты, не прошедшие аттестацию, должны ликвидировать задолженность в установленном порядке.

7. Оценочные материалы для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся по дисциплине (модулю)

7.1. Перечень компетенций с указанием этапов их формирования в процессе освоения дисциплины

№ п/п	Контролируемые разделы дисциплины (модуля)	Код и наименование индикатора контролируемой компетенции	Вид оценочного средства
1	Основы программирования в среде C#	ОПК-1.3, ОПК-1.5, ОПК- 2.1, ОПК-5.1, ОПК-5.2	Практические задания, тесты
2	Основные понятия и подходы. Приемы обеспечения технологичности программных продуктов.	ОПК-1.3, ОПК-1.5, ОПК- 2.1, ОПК-5.1, ОПК-5.2	Практические задания, тесты

3	Проектирование программного обеспечения при структурном подходе к программированию	ОПК-1.3, ОПК-1.5, ОПК- 2.1, ОПК-5.1, ОПК-5.2	Практические задания, тесты
4	Тестирование и отладка программных продуктов при структурном подходе	ОПК-1.3, ОПК-1.5, ОПК- 2.1, ОПК-5.1, ОПК-5.2	Практические задания, тесты
5	Проектирование программного обеспечения при объектно-ориентированном подходе к программированию	ОПК-1.3, ОПК-1.5, ОПК- 2.1, ОПК-5.1, ОПК-5.2	Практические задания, тесты
6	Разработка пользовательских интерфейсов. Оценка качества программного обеспечения	ОПК-1.3, ОПК-1.5, ОПК- 2.1, ОПК-5.1, ОПК-5.2	Практические задания, тесты
7	Оценка качества программного обеспечения	ОПК-1.3, ОПК-1.5, ОПК- 2.1, ОПК-5.1, ОПК-5.2	Практические задания, тесты
8	Зачет	ОПК-1.3, ОПК-1.5, ОПК- 2.1, ОПК-5.1, ОПК-5.2	

7.2. Типовые контрольные задания или иные материалы текущего контроля успеваемости, необходимые для оценки знаний, умений и навыков и (или) опыта профессиональной деятельности, характеризующих этапы формирования компетенций в процессе освоения дисциплины

Комплект тестов для проверки сформированности индикаторов достижений компетенций ОПК-1.3, 1.5, ОПК-2.1, ОПК-5.1, 5.2:

Тема 1. Основы программирования в среде С#

1. Для чего служат строки, начинающиеся с двух или трех косых черт:

С таких строк начинается описание класса; Служат для обозначения ключевых слов; Такие строки

являются комментариями и служат для документирования текста программы; Такие строки ограничивают блок операторов, выполняющихся в циклах или в операторах ветвления

2. Чему будет равно значение переменных a, b и c после выполнения следующего фрагмента программы?

```
{ int a = 0; int b = 0; int c = 0;
a = (b = 2 + 3) / 2 - 4 + (c = 5 % 2);
Console.WriteLine("a={0} b={1} c={2}", a, b, c);}
```

3. Сколько раз будет напечатано слово «Привет!»

```
static void Main(string[] args)
{ const int LIMIT = 10;
int i = 0;
while (i < LIMIT)
Console.WriteLine("Привет!");
i++; }
```

4. Сколько раз будет напечатано слово «Привет!»

```
static void Main(string[] args)
{ const int LIMIT = 10;
int i = 0;
while (i < LIMIT){
Console.WriteLine("Привет!");
i++;}}
```

5. Сколько раз будет напечатано слово «Привет!»

```
static void Main(string[] args)
{ const int LIMIT = 10;
uint i = 0;
while (i < LIMIT)
{ Console.WriteLine("Привет!");
i=i-1;}}
```

6. Сколько раз будет напечатано слово «Привет!»

```
static void Main(string[] args)
```

```
{ const int LIMIT = 10;
for (uint i = 0; i <= LIMIT; ++i)
Console.WriteLine("Привет!");}
```

7. Сколько раз будет напечатано слово «Привет!»

```
static void Main(string[] args)
{ const int LIMIT = 10;
uint i = 1;
do
{Console.WriteLine("Привет!");
i=i+1;
} while (i>LIMIT);}
```

8. Чему будет равна переменная i после завершения работы цикла?

```
static void Main(string[] args)
{ const int LIMIT = 10;
uint i = 0;
for (i = 0; i <= LIMIT; ++i)
Console.WriteLine("Привет!");
Console.WriteLine("i={0}", i);}
```

9. Чему будет равна переменная a после завершения работы программы?

```
static void Main(string[] args)
{ bool i = false;
int a = 0;
if (i = true) a = 3;
else a = -3;
Console.WriteLine("a={0}", a);}
```

10. Чему будет равна переменная i после завершения работы программы?

```
static void Main(string[] args)
{ int i = -1;
if (i <0 && (i!=-1)) i = 3;
else i = -3;
Console.WriteLine("i={0}", i);}
```

11. Чему будет равна переменная i после завершения работы программы?

```
static void Main(string[] args)
{ int i = 0;
if ((i==i) || (i==2)) i = 3;
else i = -3;
Console.WriteLine("i={0}", i);}
```

12. Чему будет равна переменная i после завершения работы программы?

```
static void Main(string[] args)
{ int a=8; int b=12; int i =0;
i=((2*b-2*8)+2)/3;
if (i-3>0)
if (i==3) i=1;
else i = 2;
Console.WriteLine("i={0}", i);}
```

13. Чему будет равно значение переменной a после завершения работы программы?

```
static void Main(string[] args)
{ int a=8; int b=12; bool i =true;
if (i == a > b) a = 1;
else
if (i == !(a != b)) a = 2;
else if (i ==(a< b)) a = 3;
else a = 4;
Console.WriteLine("a={0}", a);}
```

14. Чему будет равно значение переменной a после завершения работы программы?

```
static void Main(string[] args)
```

```
{ int a=-10;
a = (a + 1 > 0) ? a : -a;
Console.WriteLine("a={0}", a);}
```

15. Чему будет равно значение переменной b после завершения работы программы?

```
static void Main(string[] args)
{ int b=-9;
b = (b + 1 >= 0) ? b : -b;
Console.WriteLine("b={0}", b);}
```

16. Чему будет равно значение переменной c после завершения работы программы?

```
static void Main(string[] args)
{ int c = -10; int a = 0;
a= (c + 1 >= 0) ? c : -c;
Console.WriteLine("c={0}", c);}
```

17. Чему будет равно значение переменной d после завершения работы программы?

```
static void Main(string[] args)
{ int d = -10;
d+= (d + 1 > 0) ? d : -d;
Console.WriteLine("d={0}", d);}
```

18. Чему будет равно значение переменной e после завершения работы программы?

```
static void Main(string[] args)
{ int e = -10;
e= (--e> 0) ? e : -e;
Console.WriteLine("e={0}", e);}
```

19. Чему будет равно значение переменной s после завершения работы программы?

```
static void Main(string[] args)
{ int i = 5;
float s = 1;
s*=(i+=5)/4;
Console.WriteLine("s={0}", s);}
```

20. Чему будет равно значение переменной s после завершения работы программы?

```
static void Main(string[] args)
{ int i = 5;
float s = 1;
s*=(float)(i+=5)/4;
Console.WriteLine("s={0}", s);}
```

21. Чему будет равно значение переменной i после завершения работы программы?

```
static void Main(string[] args)
{ int i = 0; int s = 10;
for (i = 0; s > 0; i++)
s /= 2;
Console.WriteLine("i={0}", i);}
```

22. Чему будет равно значение переменной j после завершения работы программы?

```
static void Main(string[] args)
{ int j = 0; int s = 10;
for (j = 0; s > 0 && (s /= 2) != 0; j++);
Console.WriteLine("j={0}", j);}
```

23. Чему будет равно значение переменной j после завершения работы программы?

```
static void Main(string[] args)
{ int j = 0; int s = 10;
for (j = 0; s > 0 && (s %= 2) == 0; j++);
Console.WriteLine("j={0}", j);}
```

24. Чему будет равно значение переменной k (в шестнадцатеричном формате) после завершения работы программы?

```
static void Main(string[] args)
{ int i = 0x1234; int k = 0;
```

```
k=i<<4;
```

```
Console.WriteLine("k={0:x}", k);}
```

25. Чему будет равно значение переменной j (в шестнадцатеричном формате) после завершения работы программы?

```
static void Main(string[] args)
```

```
{ int i = 0x1234; int j = 0;
```

```
j=i>>8;
```

```
Console.WriteLine("j={0:x}", j); }
```

26. Чему будет равно значение переменной r (в шестнадцатеричном формате) после завершения работы программы?

```
static void Main(string[] args)
```

```
{ int i = 0x45ff; int j = 0x00ff;
```

```
int r =0;
```

```
r=i^j;
```

```
Console.WriteLine("r={0:x}", r);}
```

27. Чему будет равно значение переменной r (в шестнадцатеричном формате) после завершения работы программы?

```
static void Main(string[] args)
```

```
{ int i = 0x45ff; int j = 0x00ff;
```

```
int r =0;
```

```
r=i|j;
```

```
Console.WriteLine("r={0:x}", r);}
```

28. Чему будет равно значение переменной r (в шестнадцатеричном формате) после завершения работы программы?

```
static void Main(string[] args)
```

```
{ int i = 0x45ff; int j = 0x00ff;
```

```
int r =0;
```

```
r=i&j;
```

```
Console.WriteLine("r={0:x}", r);}
```

29. Чему будет равно значение переменной a после завершения работы программы?

```
static void Main(string[] args)
```

```
{ int a = 5; int b = 2;
```

```
switch (b++)
```

```
{ case 1: a += 3; break;
```

```
case 2: a *= 3; break;
```

```
case 3: a = a * 2 + 10; break;
```

```
case 4: a %= 6; break; }
```

```
Console.WriteLine("a={0}", a);}
```

30. Чему будет равно значение переменных a и b после завершения работы программы?

```
static void Main(string[] args)
```

```
{ int a = 5; int b = 2; int c = 30; int d = 7;
```

```
a += ++b+c;
```

```
c -= d++ + ++b;
```

```
Console.WriteLine("a={0} c={1}", a,c);}
```

7.3. Система оценивания результатов обучения по дисциплине (модулю) при проведении текущего контроля успеваемости

<p>Оценка «отлично» (зачтено)</p>	<p>знания:</p> <ul style="list-style-type: none"> - систематизированные, глубокие и полные знания по всем разделам дисциплины, а также по основным вопросам, выходящим за пределы учебной программы; - точное использование научной терминологии, систематически грамотное и логически правильное изложение ответа на вопросы; - полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой по дисциплине (модулю) <p>умения:</p> <ul style="list-style-type: none"> - умеет ориентироваться в теориях, концепциях и направлениях дисциплины и давать им критическую оценку, используя научные достижения других дисциплин <p>навыки:</p> <ul style="list-style-type: none"> - высокий уровень сформированности заявленных в рабочей программе компетенций; - владеет навыками самостоятельно и творчески решать сложные проблемы и нестандартные ситуации; - применяет теоретические знания для выбора методики выполнения заданий; - грамотно обосновывает ход решения задач; - безупречно владеет инструментарием учебной дисциплины, умение его эффективно использовать в постановке научных и практических задач; - творческая самостоятельная работа на практических/семинарских/лабораторных занятиях, активно участвует в групповых обсуждениях, высокий уровень культуры исполнения заданий
<p>Оценка «хорошо» (зачтено)</p>	<p>знания:</p> <ul style="list-style-type: none"> - достаточно полные и систематизированные знания по дисциплине; - усвоение основной и дополнительной литературы, рекомендованной рабочей программой по дисциплине (модулю) <p>умения:</p> <ul style="list-style-type: none"> - умеет ориентироваться в основных теориях, концепциях и направлениях дисциплины и давать им критическую оценку; - использует научную терминологию, лингвистически и логически правильно излагает ответы на вопросы, умеет делать обоснованные выводы; - владеет инструментарием по дисциплине, умение его использовать в постановке и решении научных и профессиональных задач <p>навыки:</p> <ul style="list-style-type: none"> - самостоятельная работа на практических занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий; - средний уровень сформированности заявленных в рабочей программе компетенций; - без затруднений выбирает стандартную методику выполнения заданий; - обосновывает ход решения задач без затруднений

<p>Оценка «удовлетворительно» (зачтено)</p>	<p>знания: - достаточный минимальный объем знаний по дисциплине; - усвоение основной литературы, рекомендованной рабочей программой; - использование научной терминологии, стилистическое и логическое изложение ответа на вопросы, умение делать выводы без существенных ошибок умения: - умеет ориентироваться в основных теориях, концепциях и направлениях по дисциплине и давать им оценку; - владеет инструментарием учебной дисциплины, умение его использовать в решении типовых задач; - умеет под руководством преподавателя решать стандартные задачи навыки: - работа под руководством преподавателя на практических занятиях, допустимый уровень культуры исполнения заданий; - достаточный минимальный уровень сформированности заявленных в рабочей программе компетенций; - испытывает затруднения при обосновании алгоритма выполнения заданий</p>
<p>Оценка «неудовлетворительно» (не зачтено)</p>	<p>знания: - фрагментарные знания по дисциплине; - отказ от ответа (выполнения письменной работы); - знание отдельных источников, рекомендованных рабочей программой по дисциплине; умения: - не умеет использовать научную терминологию; - наличие грубых ошибок навыки: - низкий уровень культуры исполнения заданий; - низкий уровень сформированности заявленных в рабочей программе компетенций; - отсутствие навыков самостоятельной работы; - не может обосновать алгоритм выполнения заданий</p>

7.4. Теоретические вопросы и практические задания для проведения промежуточной аттестации обучающихся, необходимые для оценки знаний, умений и навыков и (или) опыта профессиональной деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

7.4.1. Теоретические вопросы для проведения промежуточной аттестации обучающихся

1. Технология программирования. Основные понятия. Этапы развития технологии программирования.
2. Жизненный цикл и этапы разработки программного обеспечения
3. Модели жизненного цикла программного обеспечения
4. Понятия эффективности и технологичности программного обеспечения. Модули и их свойства (сцепление и связность модулей)
5. Структурное программирование
6. Средства описания структурных алгоритмов (псевдокоды, схемы алгоритмов)
7. Средства описания структурных алгоритмов (Flow-формы, диаграммы Насси- Шнейдермана)
8. Правила оформления программ
9. Разработка технического задания
10. Классификация моделей разрабатываемого программного обеспечения
11. Структурный подход. Диаграммы переходов состояний
12. Структурный подход. Функциональные диаграммы
13. Структурный подход. Диаграммы потоков данных

14. Структурный подход. Структуры данных и диаграммы отношений компонентов данных
15. Структурный подход. Сетевая модель данных (Диаграммы «сущность-связь»)
16. Проектирование программного обеспечения при структурном подходе. Структурная и функциональная схемы
17. Структурный подход. Структурные карты Константайна
18. Проектирование структур данных. Методика Джексона
19. UML- стандартный язык описания разработки программных продуктов с использованием объектного подхода
20. Диаграммы вариантов использования
21. Диаграмма классов. Отношения между классами
22. Диаграмма последовательностей
23. Диаграммы деятельности
24. Диаграмма пакетов
25. Диаграммы состояний объекта
26. Диаграмма кооперации
27. Диаграмма компонентов
28. Диаграмма размещения
29. Структурное тестирование. Тестирование базового пути
30. Структурное тестирование. Тестирование условий
31. Структурное тестирование. Тестирование циклов
32. Структурное тестирование. Тестирование потоков данных
33. Функциональное тестирование. Разбиение на классы эквивалентности и анализ граничных значений
34. Функциональное тестирование. Анализ причинно-следственных связей
35. Классификация ошибок
36. Методы отладки программного обеспечения
37. Разработка пользовательского интерфейса. Классификация диалогов и общие принципы их работы
38. Разработка пользовательского интерфейса. Граф диалога с пользователем
39. Оценка качества программного обеспечения по ГОСТ 28195-89

7.4.2. Практические задания для проведения промежуточной аттестации обучающихся

Практические задания для проверки сформированности индикаторов достижений компетенций ОПК-1.3, 1.5, ОПК-2.1, ОПК-5.1, 5.2:

Одномерные массивы (примеры заданий)

Вариант 1

В одномерном массиве, состоящем из n вещественных элементов, вычислить

- сумму отрицательных элементов массива;
- произведение элементов массива, расположенных между максимальным и минимальным элементами.
- Упорядочить элементы массива по возрастанию.

Вариант 2

В одномерном массиве, состоящем из n вещественных элементов, вычислить

- сумму положительных элементов массива;
- произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.
- Упорядочить элементы массива по убыванию.

Вариант 3

В одномерном массиве, состоящем из n целочисленных элементов, вычислить:

- произведение элементов массива с четными номерами;
- сумму элементов массива, расположенных между первым и последним нулевыми элементами.
- Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом — все отрицательные (элементы, равные нулю, считать положительными).

Двумерные «прямоугольные» массивы (примеры заданий)

Вариант 1

Дана целочисленная прямоугольная матрица. Определить:

- количество строк, не содержащих ни одного нулевого элемента;
- максимальное из чисел, встречающихся в заданной матрице более одного раза.

Вариант 2

Дана целочисленная прямоугольная матрица. Определить количество столбцов, не содержащих одного нулевого элемента.

Характеристикой строки целочисленной матрицы назовем сумму ее положительных четных элементов. Переставляя строки заданной матрицы, расположить их в соответствии с ростом характеристик.

Вариант 3

Дана целочисленная прямоугольная матрица. Определить:

- количество столбцов, содержащих хотя бы один нулевой элемент;
- номер строки, в которой находится самая длинная серия одинаковых элементов.

Работа со строками класса String и StringBuilder (примеры заданий)

Вариант 1

1. Написать программу, которая вводит текст, состоящий из нескольких предложений, и выводит каждое слово в обратном порядке.

2. Дана строка не более чем из шести произвольных различных символов. Разработать программу вывода всех возможных подстрок, составленных из символов данной строки

Вариант 2

1. Написать программу, которая вводит текст, состоящий из нескольких предложений, и выводит на экран только предложения, содержащие введенное с клавиатуры слово.

2. По правилам машинописи после запятой в тексте всегда ставится пробел. Составить программу исправления такого типа ошибок в тексте.

Вариант 3

1. Написать программу, которая вводит текст, состоящий из нескольких предложений, и выводит на экран только строки, содержащие двузначные числа.

2. Дан текст. Вывести слова, которые отличны от последнего слова и в них нет повторяющихся букв

Регулярные выражения (примеры заданий)

Используя регулярные выражения, напишите следующие приложения:

1. Дан текстовый файл f и строка s , Переписать в новый файл g все строки файла f , содержащие значения переменной s в качестве подстроки.

2. Дан текстовый файл. Вывести все слова, начинающиеся с согласных букв русского алфавита.

3. Дан текстовый файл. Вывести все слова, состоящие только из цифр

Тема 4

Вариант 1.

1. Сформировать файл последовательности 15 чисел, в которой каждый i -й компонент определяется по формуле

$$y = \sin(i * \pi / 8), \text{ если } i < 8;$$

$$y = 4 \cos(i(\pi + 1)/5), \text{ если } i > 8.$$

Определить количество положительных значений, содержащихся в сформированном файле.

2. Составить программу, определяющую правильность следования скобок в строке символов, используя для этой цели стек на основе файла.

Вариант 2.

1. Сформировать файл последовательности 20 чисел, в которой каждый i -й компонент

определяется по формуле
 $y = \sin(i \cdot \pi / 8)$, если $i < 8$;
 $y = 4 \cos(i(\pi + 1)/5)$, если $i > 8$.

Определить количество отрицательных значений, содержащихся в сформированном файле.

2. Подсчитать количество сдвоенных символов 'ee', 'nn', 'll' в тексте, расположенном в текстовом файле.

Вариант 3.

1. Сформировать файл из значений случайных величин:
0,324; 0,524; 0,789; 0,556; 0,761; 0,248; 0,345; 0,911; 0,216,

Определить для данной последовательности среднее арифметическое компонентов, значения которых меньше 0,5.

2. Разбить произвольный текст, находящийся в файле, на строки определенной длины. При переносе слова предусмотреть вывод дефиса.

Тема 5

Вариант 1

1. Даны цифры двух целых чисел: двузначного a_1a_2 и однозначного b , где a_1 – число единиц, a_2

число десятков, Получить цифры числа, равного разности заданных чисел (известно, что это двузначное число).

2. Даны N целых чисел x_1, x_2, \dots, x_N . Расставить между ними знак «+» и «-» так, чтобы значение получившегося выражения было равно заданному целому S .

Вариант 2

1. Даны натуральное число N и вещественные числа x_1, x_2, \dots, x_N . Выяснить, верно ли, что сумма тех вещественных чисел x_i , которые меньше 20,5 не превышает 50.

2. Даны N целых чисел x_1, x_2, \dots, x_N . Требуется вычеркнуть из них минимальное количество чисел так, чтобы оставшиеся шли в порядке возрастания.

3. Даны натуральные числа m, n_1, \dots, n_m ($m > 2$). Вычислить НОД(n_1, \dots, n_m), воспользовавшись для этого соотношением $\text{НОД}(n_1, \dots, n_k) = \text{НОД}(\text{НОД}(n_1, \dots, n_{k-1}), n_k)$ ($k = 3, \dots, m$) и алгоритмом

Евклида

Вариант 3

1. Даны натуральное число N и вещественные числа x_1, x_2, \dots, x_N . Выяснить, верно ли, что сумма тех вещественных чисел x_i , которые не превышают 10 кратна 3.

2. Вывести все простые числа от M до N включительно.

3. Даны натуральное число n , целые числа $a_1, \dots, a_{25}, b_1, \dots, b_n$. Среди a_1, \dots, a_{25} нет повторяющихся

чисел, нет их и среди b_1, \dots, b_n

. Построить пересечение последовательностей a_1, \dots, a_{25} и b_1, \dots, b_n

(т. е. получить в каком-нибудь порядке все числа, принадлежащие последовательности a_1, \dots, a_{25} и последовательности b_1, \dots, b_n одновременно).

Тема 6

Работа с классами в C#

Вариант 1

Описать класс, реализующий десятичный счетчик, который может увеличивать или уменьшать свое значение на единицу в заданном диапазоне.

Предусмотреть инициализацию счетчика значениями по умолчанию и произвольными значениями. Счетчик имеет два метода: увеличения и уменьшения, — и свойство, позволяющее получить его текущее состояние.

При выходе за границы диапазона выбрасываются исключения. Написать программу, демонстрирующую все разработанные элементы класса.

Создать дочерний класс Счетчик1, который может увеличивать или уменьшать свое значение на любое введенное число в заданном диапазоне.

Вариант 2

Описать класс, реализующий шестнадцатеричный счетчик, который может увеличивать или уменьшать свое значение на единицу в заданном диапазоне.

Предусмотреть инициализацию счетчика значениями по умолчанию и произвольными значениями. Счетчик имеет два метода: увеличения и уменьшения, — и свойство, позволяющее получить его текущее состояние.

При выходе за границы диапазона выбрасываются исключения.

Написать программу, демонстрирующую все разработанные элементы класса.

Создать дочерний класс, который может увеличивать или уменьшать свое значение на любое введенное число в заданном диапазоне

Вариант 3

Описать класс, представляющий треугольник. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и вращения на заданный угол.

Описать свойства для получения состояния объекта.

При невозможности построения треугольника выбрасывается исключение.

Написать программу, демонстрирующую все разработанные элементы класса.

Создать дочерний класс Прямоугольник, который может увеличивать или уменьшать свое значение на любое введенное число в заданном диапазоне

Тема 7

Построить иерархию классов в соответствии с вариантом задания. Предусмотреть, по крайней мере, два интерфейса и обработку событий (предусмотреть хотя бы одно событие)

1. Ученик, бакалавр, магистр
2. Студент, преподаватель, заведующий кафедрой
3. Студент, группа, институт
4. Служащий, рабочий, инженер
5. Рабочий, цех, предприятие
6. Рабочий, инженер, администрация
7. Деталь, узел, механизм
8. Организация, завод, страховая компания
9. Журнал, книга, учебник
10. Тест, экзамен, выпускной экзамен
11. Город, область, страна
12. Водоем, озеро, море
13. Растение, куст, дерево
14. Растение, дерево, лесной массив
15. Молочный продукт, сметана, сыр
16. Квитанция, накладная, счет
17. Транспортное средство, автомобиль, поезд
18. Двигатель, двигатель внутреннего сгорания, реактивный двигатель
19. Государство, республика, монархия
20. Животное, млекопитающее, парнокопытное
21. Корабль, парусник, пароход
22. Летательный аппарат, планер, самолет
23. Спортсмен, мастер спорта, заслуженный мастер спорта

7.4.3. Примерные темы курсовой работы (проекта) (при наличии)

Курсовые работы в данной рабочей программе не предусмотрены учебным планом

7.5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта профессиональной деятельности, характеризующие этапы формирования компетенций

Процедура проведения промежуточной аттестации и текущего контроля успеваемости регламентируется локальным нормативным актом, определяющим порядок организации и проведения текущего контроля успеваемости и промежуточной аттестации обучающихся.

Процедура оценивания формирования компетенций при проведении текущего контроля приведена в п. 7.2.

Типовые контрольные задания или иные материалы текущего контроля приведены в п. 7.3.

Промежуточная аттестация по дисциплине проводится в форме зачета.

7.6. Критерии оценивания сформированности компетенций при проведении промежуточной аттестации

Критерии оценивания	Уровень освоения и оценка			
	Оценка «неудовлетворительно»	Оценка «удовлетворительно»	Оценка «хорошо»	Оценка «отлично»
	«не зачтено»	«зачтено»		
	Уровень освоения компетенции «недостаточный». Компетенции не сформированы. Знания отсутствуют, умения и навыки не сформированы	Уровень освоения компетенции «пороговый». Компетенции сформированы. Сформированы базовые структуры знаний. Умения фрагментарны и носят репродуктивный характер. Демонстрируется низкий уровень самостоятельности практического навыка.	Уровень освоения компетенции «продвинутой». Компетенции сформированы. Знания обширные, системные. Умения носят репродуктивный характер, применяются к решению типовых заданий. Демонстрируется достаточный уровень самостоятельности устойчивого практического навыка.	Уровень освоения компетенции «высокий». Компетенции сформированы. Знания аргументированные, всесторонние. Умения успешно применяются к решению как типовых, так и нестандартных творческих заданий. Демонстрируется высокий уровень самостоятельности, высокая адаптивность практического навыка

знания	<p>Обучающийся демонстрирует:</p> <ul style="list-style-type: none"> -существенные пробелы в знаниях учебного материала; -допускаются принципиальные ошибки при ответе на основные вопросы билета, отсутствует знание и понимание основных понятий и категорий; -непонимание сущности дополнительных вопросов в рамках заданий билета. 	<p>Обучающийся демонстрирует:</p> <ul style="list-style-type: none"> -знания теоретического материала; -неполные ответы на основные вопросы, ошибки в ответе, недостаточное понимание сущности излагаемых вопросов; -неуверенные и неточные ответы на дополнительные вопросы. 	<p>Обучающийся демонстрирует:</p> <ul style="list-style-type: none"> -знание и понимание основных вопросов контролируемого объема программного материала; - знания теоретического материала -способность устанавливать и объяснять связь практики и теории, выявлять противоречия, проблемы и тенденции развития; -правильные и конкретные, без грубых ошибок, ответы на поставленные вопросы. 	<p>Обучающийся демонстрирует:</p> <ul style="list-style-type: none"> -глубокие, всесторонние и аргументированные знания программного материала; -полное понимание сущности и взаимосвязи рассматриваемых процессов и явлений, точное знание основных понятий, в рамках обсуждаемых заданий; -способность устанавливать и объяснять связь практики и теории, -логически последовательные, содержательные, конкретные и исчерпывающие ответы на все задания билета, а также дополнительные вопросы экзаменатора.
умения	<p>При выполнении практического задания билета обучающийся продемонстрировал недостаточный уровень умений. Практические задания не выполнены. Обучающийся не отвечает на вопросы билета при дополнительных наводящих вопросах преподавателя.</p>	<p>Обучающийся выполнил практическое задание билета с существенными неточностями. Допускаются ошибки в содержании ответа и решении практических заданий. При ответах на дополнительные вопросы было допущено много неточностей.</p>	<p>Обучающийся выполнил практическое задание билета с небольшими неточностями. Показал хорошие умения в рамках освоенного учебного материала. Предложенные практические задания решены с небольшими неточностями. Ответил на большинство дополнительных вопросов.</p>	<p>Обучающийся правильно выполнил практическое задание билета. Показал отличные умения в рамках освоенного учебного материала. Решает предложенные практические задания без ошибок. Ответил на все дополнительные вопросы.</p>

владение навыками	<p>Не может выбрать методику выполнения заданий. Допускает грубые ошибки при выполнении заданий, нарушающие логику решения задач. Делает некорректные выводы. Не может обосновать алгоритм выполнения заданий.</p>	<p>Испытывает затруднения по выбору методики выполнения заданий. Допускает ошибки при выполнении заданий, нарушения логики решения задач. Испытывает затруднения с формулированием корректных выводов. Испытывает затруднения при обосновании алгоритма выполнения заданий.</p>	<p>Без затруднений выбирает стандартную методику выполнения заданий. Допускает ошибки при выполнении заданий, не нарушающие логику решения задач. Делает корректные выводы по результатам решения задачи. Обосновывает ход решения задач без затруднений.</p>	<p>Применяет теоретические знания для выбора методики выполнения заданий. Не допускает ошибок при выполнении заданий. Самостоятельно анализирует результаты выполнения заданий. Грамотно обосновывает ход решения задач.</p>
-------------------	--	---	---	--

Оценка по дисциплине зависит от уровня сформированности компетенций, закрепленных за дисциплиной, и представляет собой среднее арифметическое от выставленных оценок по отдельным результатам обучения (знания, умения, владение навыками).

Оценка «отлично»/«зачтено» выставляется, если среднее арифметическое находится в интервале от 4,5 до 5,0.

Оценка «хорошо»/«зачтено» выставляется, если среднее арифметическое находится в интервале от 3,5 до 4,4.

Оценка «удовлетворительно»/«зачтено» выставляется, если среднее арифметическое находится в интервале от 2,5 до 3,4.

Оценка «неудовлетворительно»/«не зачтено» выставляется, если среднее арифметическое находится в интервале от 0 до 2,4.

8. Учебно-методическое и материально-техническое обеспечение дисциплины (модуля)

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

№ п/п	Автор, название, место издания, издательство, год издания учебной и учебно-методической литературы	Количество экземпляров/электронный адрес ЭБС
<u>Основная литература</u>		
1	Кирсяев А. Н., Теория и технология программирования. Программное обеспечение вычислительной математики, Санкт-Петербург: Санкт-Петербургский политехнический университет Петра Великого, 2017	http://www.iprbookshop.ru/83309.html
2	Терехов А. Н., Технология программирования, Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017	http://www.iprbookshop.ru/67370.html
3	Громов Ю. Ю., Иванова О. Г., Беляев М. П., Минин Ю. В., Технология программирования, Тамбов: Тамбовский государственный технический университет, ЭБС АСВ, 2013	http://www.iprbookshop.ru/63910.html
<u>Дополнительная литература</u>		

1	Буч Г., Рамбо Д., Якобсон И., Язык UML. Руководство пользователя, Москва: ДМК Пресс, 2008	http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=1246
2	Павловская Т. А., Программирование на языке высокого уровня С#, Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016	http://www.iprbookshop.ru/73713.html
3	Биллиг В. А., Основы программирования на С#, Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016	http://www.iprbookshop.ru/73695.html

Обучающиеся из числа инвалидов и лиц с ОВЗ обеспечиваются печатными и (или) электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.

8.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)

Наименование ресурса сети «Интернет»	Электронный адрес ресурса
ГОСТ 19.xxx Единая система программной документации (ЕСПД)	http://www.rugost.com/index.php
Единая система программной документации (ЕСПД).ГОСТ 19.xxx	http://standartgost.ru/0/2870/2880-edinaya_sistema_programmnoy_dokumentatsii

8.3. Перечень современных профессиональных баз данных и информационных справочных систем

Наименование	Электронный адрес ресурса
Научная электронная библиотека eLIBRARY.RU	Научная электронная библиотека eLIBRARY.RU
Электронно-библиотечная система издательства "IPRbooks"	http://www.iprbookshop.ru/
Электронно-библиотечная система издательства "ЮРАЙТ"	https://www.biblio-online.ru/
Электронно-библиотечная система издательства "Лань"	https://e.lanbook.com/
Система дистанционного обучения СПбГАСУ Moodle	https://moodle.spbgasu.ru/

8.4. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного и свободно распространяемого программного обеспечения

Наименование	Способ распространения (лицензионное или свободно распространяемое)
Microsoft Office 2016	Договор № Д32009689201 от 18.12.2020г Программные продукты Майкрософт, договор № Д32009689201 от 18.12.2020 с АО "СофтЛайн Трейд": Windows 10, Project Professional 2016, Visio Professional 2016, Office 2016.
Microsoft Visual Studio 2017	Договор № Д32009689201 от 18.12.2020г Программные продукты Майкрософт, договор № Д32009689201 от 18.12.2020 с АО "СофтЛайн Трейд": Windows 10, Project Professional 2016, Visio Professional 2016, Office 2016.

8.5. Материально-техническое обеспечение дисциплины

Сведения об оснащённости учебных аудиторий и помещений для самостоятельной работы

Наименование учебных аудиторий и помещений для самостоятельной работы	Оснащённость оборудованием и техническими средствами обучения
47. Учебные аудитории для проведения практических занятий, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	Учебная аудитория для проведения практических занятий, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации – комплект мультимедийного оборудования (персональный компьютер, мультимедийный проектор, экран, аудио-система), доска маркерная белая эмалевая, комплект учебной мебели, подключение к компьютерной сети СПбГАСУ, выход в Интернет.
47. Компьютерный класс	Рабочие места с ПК (стол компьютерный, системный блок, монитор, клавиатура, мышь), стол рабочий, подключение к компьютерной сети СПбГАСУ, выход в Internet.

Для инвалидов и лиц с ОВЗ обеспечиваются специальные условия для получения образования в соответствии с требованиями нормативно-правовых документов.