

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение высшего  
образования  
"САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ"

Кафедра № 43

УТВЕРЖДАЮ  
Руководитель образовательной программы  
*с.я. прен.*  
(должность, уч. степень, звание)

Д.В. Куртяник  
(инициалы, фамилия)  
*Д.В. Куртяник*  
(подпись)  
«06» февраля 2025 г

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Технология программирования»  
(Наименование дисциплины)

Код направления подготовки/ специальности	09.03.01
Наименование направления подготовки/ специальности	Информатика и вычислительная техника
Наименование направленности	Компьютерные технологии, системы и сети
Форма обучения	заочная
Год приема	2025

Лист согласования рабочей программы дисциплины

Программу составил (а)

Ст. преподаватель *Соловьева* 05.02.2025 Н.А. Соловьева  
(должность, уч. степень, звание) (подпись, дата) (инициалы, фамилия)

Программа одобрена на заседании кафедры № 43  
«06» февраля 2025 г, протокол № 01/2025

Заведующий кафедрой № 43  
д.т.н., проф. *Охтилев* 06.02.2025 М.Ю. Охтилев  
(уч. степень, звание) (подпись, дата) (инициалы, фамилия)

Заместитель директора института №4 по методической работе  
доц., к.т.н. *Фоменкова* 06.02.2025 А.А. Фоменкова  
(должность, уч. степень, звание) (подпись, дата) (инициалы, фамилия)

## Аннотация

Дисциплина «Технология программирования» входит в образовательную программу высшего образования – программу бакалавриата по направлению подготовки/ специальности 09.03.01 «Информатика и вычислительная техника» направленности «Компьютерные технологии, системы и сети». Дисциплина реализуется кафедрой «№43».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

ОПК-2 «Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности»

Содержание дисциплины охватывает круг вопросов, связанных с анализом и проектированием больших программных комплексов или информационных систем (ИС), с использованием современных методологий и средств проектирования.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, самостоятельная работа обучающегося.

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости, промежуточная аттестация в форме экзамена.

Общая трудоемкость освоения дисциплины составляет 6 зачетных единиц, 216 часов.

Язык обучения по дисциплине «русский»

1. Перечень планируемых результатов обучения по дисциплине  
1.1. Цели преподавания дисциплины

Целью дисциплины «Технология программирования» является получение студентами необходимых знаний, умений и навыков в области проектирования информационных систем в различных прикладных областях с использованием современных методологий и технологических средств проектирования.

1.2. Дисциплина входит в состав обязательной части образовательной программы высшего образования (далее – ОП ВО).

1.3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОП ВО.

В результате изучения дисциплины обучающийся должен обладать следующими компетенциями или их частями. Компетенции и индикаторы их достижения приведены в таблице 1.

Таблица 1 – Перечень компетенций и индикаторов их достижения

Категория (группа) компетенции	Код и наименование компетенции	Код и наименование индикатора достижения компетенции
Общепрофессиональные компетенции	ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.3.1 знать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности ОПК-2.У.1 уметь выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности ОПК-2.В.1 владеть навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности

2. Место дисциплины в структуре ОП

Дисциплина может базироваться на знаниях, ранее приобретенных обучающимися при изучении следующих дисциплин:

- «Основы программирования»,
- Алгоритмы и структуры данных».

Знания, полученные при изучении материала данной дисциплины, имеют как самостоятельное значение, так и используются при изучении других дисциплин:

- «Базы данных».

### 3. Объем и трудоемкость дисциплины

Данные об общем объеме дисциплины, трудоемкости отдельных видов учебной работы по дисциплине (и распределение этой трудоемкости по семестрам) представлены в таблице 2.

Таблица 2 – Объем и трудоемкость дисциплины

Вид учебной работы	Всего	Трудоемкость по семестрам	
		№4	№5
1	2	3	4
<b>Общая трудоемкость дисциплины, ЗЕ/ (час)</b>	6/ 216	2/ 72	4/ 144
<b>Из них часов практической подготовки</b>			
<b>Аудиторные занятия, всего час.</b>	28	16	12
в том числе:			
лекции (Л), (час)	14	8	6
практические/семинарские занятия (ПЗ), (час)			
лабораторные работы (ЛР), (час)	14	8	6
курсовой проект (работа) (КП, КР), (час)			
экзамен, (час)	9		9
<b>Самостоятельная работа, всего (час)</b>	179	56	123
<b>Вид промежуточной аттестации:</b> зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.)	Дифф. Зач., Экз.	Дифф. Зач.	Экз.

### 4. Содержание дисциплины

4.1. Распределение трудоемкости дисциплины по разделам и видам занятий.

Разделы, темы дисциплины и их трудоемкость приведены в таблице 3.

Таблица 3 – Разделы, темы дисциплины, их трудоемкость

Разделы, темы дисциплины	Лекции (час)	ПЗ (СЗ)	ЛР (час)	КП (час)	СРС (час)
Семестр 4					
<b>Раздел 1.</b> Информационные системы, подходы и методологии проектирования.					
Тема 1.1. Этапы развития технологии программирования и средств создания программного обеспечения.	1				6
Тема 1.2. Жизненный цикл программного обеспечения, его основные этапы и модели.					
Тема 1.3. Тестирование программного обеспечения					

<p><b>Раздел 2.</b> Методология структурного анализа и проектирования программного обеспечения.</p> <p>Тема 2.1. Сущность и базовые принципы структурного подхода. Основные средства и диаграммы.</p> <p>Тема 2.2. Диаграммы потоков данных. Иерархия и декомпозиция диаграмм. Методы спецификации процессов. Построение словаря данных.</p> <p>Тема 2.3. Инфологическое моделирование ИС, ER-диаграммы. Нормализация ER-диаграмм, построение структуры базы данных ИС</p>	3		6		25
<p><b>Раздел 3.</b> Объектно-ориентированное программирование (ООП) и библиотека STL языка C++.</p> <p>Тема 3.1. Основные принципы и понятия ООП</p> <p>Тема 3.2. Библиотека STL языка C++: контейнеры, итераторы, алгоритмы.</p>	4		2		25
Итого в семестре:	8		8		56
Семестр 5					

<p><b>Раздел 4. Методологии объектно-ориентированного анализа и проектирования программного обеспечения.</b></p> <p>Тема 4.1. Основные принципы объектно-ориентированного проектирования и модели ИС. Универсальный язык моделирования UML, основные понятия, обозначения и диаграммы.</p> <p>Тема 4.2. Диаграмма вариантов использования. Назначение, связь с другими диаграммами, обозначения, рассмотрение примера.</p> <p>Тема 4.3. Разработка программы по принципам объектно-ориентированного программирования.</p> <p>Тема 4.4. Диаграмма классов. Назначение, связь с другими диаграммами, обозначения, рассмотрение примера.</p> <p>Тема 4.5. Диаграммы последовательностей, кооперации, состояния, пакетов и размещения.</p> <p>Тема 4.6. Автоматическая генерация кодов программ, обратное проектирование (реинжиниринг).</p> <p>Тема 4.7. Технологические средства разработки программного обеспечения. CASE-технологии и CASE-пакеты.</p>	4		6		80
<p><b>Раздел 5. Шаблоны проектирования.</b></p> <p>Тема 5.1. Шаблоны проектирования – основные понятия, назначение.</p> <p>Тема 5.2. Знакомство с примерами порождающих, структурных и поведенческих шаблонов.</p>	2				43
Итого в семестре:	6		6		123
Итого	14	0	14	0	179

Практическая подготовка заключается в непосредственном выполнении обучающимися определенных трудовых функций, связанных с будущей профессиональной деятельностью.

#### 4.2. Содержание разделов и тем лекционных занятий.

Содержание разделов и тем лекционных занятий приведено в таблице 4.

Таблица 4 – Содержание разделов и тем лекционного цикла

Номер раздела	Название и содержание разделов и тем лекционных занятий
1	<p><b>Информационные системы, подходы и методологии проектирования.</b></p> <p><b>Лекция 1.1.</b> Этапы развития вычислительной техники, задач и средств Технологии программирования. Понятие информационной системы. Цели и задачи проектирования. Организация проекта программного обеспечения.</p> <p><b>Лекция 1.2.</b> Жизненный цикл программного обеспечения. Линейная, каскадная и спиралевидная модели жизненного цикла программного обеспечения. Основные этапы жизненного цикла и инструментальные средства поддержки проекта.</p> <p><b>Лекция 1.3</b> Тестирование программного обеспечения: классификация видов, области применения, инструментальные средства.</p>
2	<p><b>Методология структурного анализа и проектирования программного обеспечения.</b></p> <p><b>Лекция 2.1.</b> Сущность и базовые принципы структурного подхода. Основные этапы и средства структурного анализа и проектирования информационных систем. Основные диаграммы.</p> <p><b>Лекция 2.2.</b> Диаграммы потоков данных. Декомпозиция ДПД, построение словаря проекта. Построение словаря данных. Атрибуты описания потоков данных. БНФ-определение. Примеры описания потоков данных. Методы спецификации процессов. Структурированный естественный язык (псевдокод), визуальные языки проектирования, FLOW-формы и диаграммы Насси-Шнейдермана (структурограммы).</p> <p><b>Лекция 2.3.</b> Инфологическое моделирование. ИС. Проектирование структуры базы данных ИС. ER-диаграммы. Базовые элементы и их свойства. Нормализация ER-диаграмм. Пример построения ER-диаграммы. Примеры нормализации, приведение диаграмм к 1НФ, 2НФ, 3НФ, устранение связей типа М:М.</p>
3	<p><b>Объектно-ориентированное программирование (ООП) и библиотека STL языка C++.</b></p> <p><b>Лекция 3.1.</b> Основные принципы и понятия ООП: класс, объект, поля, методы, уровни доступа, наследование.</p>

	<b>Лекция 3.2.</b> Библиотека STL языка C++: контейнеры, итераторы, алгоритмы.
4	<p><b>Методологии объектно-ориентированного анализа и проектирования программного обеспечения.</b></p> <p><b>Лекция 4.1.</b> Основные принципы объектно-ориентированных методов проектирования. Инкапсуляция, наследование, полиморфизм. Статическая и динамическая модели информационной системы. Универсальный язык моделирования UML. Структура языка, основные понятия и обозначения.</p> <p><b>Лекция 4.2.</b> Основные диаграммы языка UML. Взаимосвязь диаграмм. Диаграмма вариантов использования. Основные понятия и обозначения. Пример построения диаграммы вариантов использования.</p> <p><b>Лекция 4.3.</b> Принципы объектно-ориентированного программирования. Разработка программы с классами.</p> <p><b>Лекция 4.4.</b> Диаграмма классов. Основные понятия и обозначения. Пример построения диаграммы классов.</p> <p><b>Лекция 4.5.</b> Диаграммы последовательностей и кооперации. Рассмотрение примеров.</p> <p><b>Лекция 4.6.</b> Диаграммы состояния и видов деятельности. Рассмотрение примеров,</p> <p><b>Лекция 4.7.</b> Диаграммы пакетов и размещения. Рассмотрение примеров,</p> <p><b>Лекция 4.8.</b> Автоматическая генерация кодов программ, обратное проектирование (реинжиниринг).</p> <p><b>Лекция 4.9.</b> Технологические средства разработки программного обеспечения ПО. CASE-пакеты. Структура типового CASE-средства.</p>
5	<p><b>Шаблоны проектирования.</b></p> <p><b>Лекция 5.1.</b> Шаблоны проектирования – основные понятия, назначение.</p> <p><b>Лекция 5.2.</b> Знакомство с примерами порождающих, структурных и поведенческих шаблонов.</p>

#### 4.3. Практические (семинарские) занятия

Темы практических занятий и их трудоемкость приведены в таблице 5.

Таблица 5 – Практические занятия и их трудоемкость

№ п/п	Темы практических занятий	Формы практических занятий	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
Учебным планом не предусмотрено					



#### 4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 6.

Таблица 6 – Лабораторные занятия и их трудоемкость

№ п/п	Наименование лабораторных работ	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
Семестр 4				
1	Разработка диаграммы потоков данных для предметной области.	3	2	2
2	Нормализация диаграммы «Сущность-связь»	3	2	2
3	Объектно-ориентированное программирование	2	2	3
Семестр 5				
1	Язык UML. Разработка диаграммы вариантов использования.	2	2	4
2	Разработка диаграммы классов	2	2	4
3	Автоматическая генерация кода программы по диаграмме классов и обратное проектирование	2	2	4
Всего		14		

#### 4.5. Курсовое проектирование/ выполнение курсовой работы

Учебным планом не предусмотрено

#### 4.6. Самостоятельная работа обучающихся

Виды самостоятельной работы и ее трудоемкость приведены в таблице 7.

Таблица 7 – Виды самостоятельной работы и ее трудоемкость

Вид самостоятельной работы	Всего, час	Семестр 4, час	Семестр 5, час
1	2	3	4
Изучение теоретического материала дисциплины (ТО)	60	20	40
Подготовка к текущему контролю успеваемости (ТКУ)	60	20	40
Контрольные работы заочников (КРЗ)	59	16	43
Всего:	179	56	123

#### 5. Перечень учебно-методического обеспечения

для самостоятельной работы обучающихся по дисциплине (модулю)

Учебно-методические материалы для самостоятельной работы обучающихся указаны в п.п. 7-11.

## 6. Перечень печатных и электронных учебных изданий

Перечень печатных и электронных учебных изданий приведен в таблице 8.

Таблица 8– Перечень печатных и электронных учебных изданий

Шифр/ URL адрес	Библиографическая ссылка	Количество экземпляров в библиотеке (кроме электронных экземпляров )
URL: <a href="https://e.lanbook.com/book/176670">https://e.lanbook.com/book/176670</a>	Волк, В. К. Базы данных. Проектирование, программирование, управление и администрирование : учебник для вузов / В. К. Волк. — 2-е изд., стер. — Санкт-Петербург : Лань, 2021. — 244 с. — ISBN 978-5-8114-8412-6. — Текст : электронный // Лань : электронно-библиотечная система Режим доступа: для авториз. пользователей.	
URL: <a href="https://e.lanbook.com/book/175503">https://e.lanbook.com/book/175503</a>	Маран, М. М. Программная инженерия : учебное пособие для вузов / М. М. Маран. — 2-е изд., стер. — Санкт-Петербург : Лань, 2021. — 196 с. — ISBN 978-5-8114-8367-9. — Текст : электронный // Лань : электронно-библиотечная система. Режим доступа: для авториз. пользователей.	
URL: <a href="https://e.lanbook.com/book/111721">https://e.lanbook.com/book/111721</a>	Забродин, А. В. Основы проектирования информационных систем с помощью языка UML : учебное пособие / А. В. Забродин, В. П. Бубнов. — Санкт-Петербург : ПГУПС, 2018. — 46 с. — ISBN 978-5-7641-1133-9. — Текст : электронный // Лань : электронно-библиотечная система. Режим доступа: для авториз. пользователей.	
URL: <a href="https://e.lanbook.com/book/168336">https://e.lanbook.com/book/168336</a>	Гринченко, Н. Н. Разработка моделей информационных систем на языке UML : учебное пособие / Н. Н. Гринченко, Ю. В. Конкин. — Рязань : РГРТУ, 2015. — 48 с. — Текст : электронный // Лань : электронно-библиотечная система. Режим доступа: для авториз. пользователей.	

## 7. Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины приведен в таблице 9.

Таблица 9 – Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

URL адрес	Наименование
<a href="https://intuit.ru/studies/courses/3632/874/lecture/14297">https://intuit.ru/studies/courses/3632/874/lecture/14297</a>	Лекция «Жизненный цикл программных систем»
<a href="https://www.lucidchart.com/pages/ru/%D0%B4%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0-dfd">https://www.lucidchart.com/pages/ru/%D0%B4%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0-dfd</a>	Ресурс по ДПД (Что такое диаграмма DFD и как ее создать?)
<a href="https://intuit.ru/studies/courses/508/364/lecture/8647">https://intuit.ru/studies/courses/508/364/lecture/8647</a>	Ег-диаграмма

## 8. Перечень информационных технологий

8.1. Перечень программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине.

Перечень используемого программного обеспечения представлен в таблице 10.

Таблица 10– Перечень программного обеспечения

№ п/п	Наименование
1	Редактор для создания диаграмм
2	Компилятор языка C++

8.2. Перечень информационно-справочных систем,используемых при осуществлении образовательного процесса по дисциплине

Перечень используемых информационно-справочных систем представлен в таблице 11.

Таблица 11– Перечень информационно-справочных систем

№ п/п	Наименование
	Не предусмотрено

## 9. Материально-техническая база

Состав материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине, представлен в таблице12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Лекционная аудитория	
2	Компьютерный класс	

## 10. Оценочные средства для проведения промежуточной аттестации

10.1. Состав оценочных средствдля проведения промежуточной аттестации обучающихся по дисциплине приведен в таблице 13.

Таблица 13 – Состав оценочных средств для проведения промежуточной аттестации

Вид промежуточной аттестации	Перечень оценочных средств
Экзамен	Список вопросов к экзамену; Тесты.
Дифференцированный зачёт	Список вопросов; Тесты

10.2. В качестве критериев оценки уровня сформированности (освоения) компетенций обучающимися применяется 5-балльная шкала оценки сформированности компетенций, которая приведена в таблице 14. В течение семестра может использоваться 100-балльная шкала модульно-рейтинговой системы Университета, правила использования которой, установлены соответствующим локальным нормативным актом ГУАП.

Таблица 14 –Критерии оценки уровня сформированности компетенций

Оценка компетенции	Характеристика сформированных компетенций
5-балльная шкала	
«отлично» «зачтено»	– обучающийся глубоко и всесторонне усвоил программный материал; – уверенно, логично, последовательно и грамотно его излагает; – опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью направления; – умело обосновывает и аргументирует выдвигаемые им идеи;

Оценка компетенции 5-балльная шкала	Характеристика сформированных компетенций
	<ul style="list-style-type: none"> <li>– делает выводы и обобщения;</li> <li>– свободно владеет системой специализированных понятий.</li> </ul>
«хорошо» «зачтено»	<ul style="list-style-type: none"> <li>– обучающийся твердо усвоил программный материал, грамотно и по существу излагает его, опираясь на знания основной литературы;</li> <li>– не допускает существенных неточностей;</li> <li>– увязывает усвоенные знания с практической деятельностью направления;</li> <li>– аргументирует научные положения;</li> <li>– делает выводы и обобщения;</li> <li>– владеет системой специализированных понятий.</li> </ul>
«удовлетворительно» «зачтено»	<ul style="list-style-type: none"> <li>– обучающийся усвоил только основной программный материал, по существу излагает его, опираясь на знания только основной литературы;</li> <li>– допускает несущественные ошибки и неточности;</li> <li>– испытывает затруднения в практическом применении знаний направления;</li> <li>– слабо аргументирует научные положения;</li> <li>– затрудняется в формулировании выводов и обобщений;</li> <li>– частично владеет системой специализированных понятий.</li> </ul>
«неудовлетворительно» «не зачтено»	<ul style="list-style-type: none"> <li>– обучающийся не усвоил значительной части программного материала;</li> <li>– допускает существенные ошибки и неточности при рассмотрении проблем в конкретном направлении;</li> <li>– испытывает трудности в практическом применении знаний;</li> <li>– не может аргументировать научные положения;</li> <li>– не формулирует выводов и обобщений.</li> </ul>

### 10.3. Типовые контрольные задания или иные материалы.

Вопросы (задачи) для экзамена представлены в таблице 15.

Таблица 15 – Вопросы (задачи) для экзамена

№ п/п	Перечень вопросов (задач) для экзамена	Код индикатора
1	Объектно-ориентированный подход к разработке ПО. Основные понятия, принципы, особенности и достоинства.	ОПК-2.3.1 ОПК-2.У.1
2	Объектно-ориентированный анализ. Основные модели	ОПК-2.3.1 ОПК-2.У.1
3	Объектно-ориентированное проектирование. Основные этапы и средства.	ОПК-2.3.1 ОПК-2.У.1
4	Язык UML. Основные понятия, обозначения, диаграммы. Взаимосвязь диаграмм.	ОПК-2.3.1 ОПК-2.У.1
5	Диаграмма вариантов использования. Основные элементы и их свойства. Пример.	ОПК-2.В.1
6	Диаграмма классов. Основные элементы и их свойства. Пример.	ОПК-2.В.1
7	Диаграмма последовательностей. Основные элементы и их свойства. Пример	ОПК-2.В.1

8	Диаграмма кооперации. Основные элементы и их свойства. Пример.	ОПК-2.В.1
9	Диаграммы состояний и видов деятельности. Основные элементы и их свойства. Пример.	ОПК-2.В.1
10	Диаграмма пакетов. Основные элементы и их свойства. Пример.	ОПК-2.В.1
11	Диаграмма размещения. Основные элементы и их свойства. Пример.	ОПК-2.В.1
12	Автоматическая генерация кодов программ. Возможности и ограничения. Пример реализации.	ОПК-2.3.1 ОПК-2.У.1
13	Реинжиниринг. Примеры использования. Пример реализации.	ОПК-2.3.1 ОПК-2.У.1
14	CASE-технологии. Сравнительный анализ.	ОПК-2.3.1 ОПК-2.У.1
15	CASE-средства. Классификация CASE-средств. Примеры современных CASE- средств и их возможности.	ОПК-2.3.1 ОПК-2.У.1
16	Пример структуры типового CASE-средства	ОПК-2.3.1 ОПК-2.У.1
17	Обзор шаблонов проектирования, назначение	ОПК-2.3.1 ОПК-2.У.1
18	Порождающие шаблоны для языка C++	ОПК-2.3.1 ОПК-2.У.1
19	Структурные шаблоны для языка C++	ОПК-2.3.1 ОПК-2.У.1
20	Поведенческие шаблоны для языка C++	ОПК-2.3.1 ОПК-2.У.1

Вопросы (задачи) для зачета / дифф. зачета представлены в таблице 16.

Таблица 16 – Вопросы (задачи) для зачета / дифф. зачета

№ п/п	Перечень вопросов (задач) для зачета / дифф. зачета	Код индикатора
1	Эволюция вычислительной техники и задач Технологии программирования. Этапы развития методологий проектирования ПО.	ОПК-2.3.1 ОПК-2.У.1
2	Жизненный цикл программного обеспечения. Модели ЖЦ ПО.	ОПК-2.3.1 ОПК-2.У.1
3	Информационные системы. Определение, основные задачи и цели создания.	ОПК-2.3.1 ОПК-2.У.1
4	Классификация современных методологий анализа и проектирования ИС.	ОПК-2.3.1 ОПК-2.У.1
5	Сущность структурного подхода к разработке ПО ИС.	ОПК-2.3.1 ОПК-2.У.1
6	Структурный анализ. Определения, основные этапы и средства структурного анализа.	ОПК-2.3.1 ОПК-2.У.1
7	Диаграммы потоков данных. Определения, основные элементы, этапы разработки.	ОПК-2.3.1 ОПК-2.У.1
8	Построение словаря данных. Способы описания структур данных	ОПК-2.3.1 ОПК-2.У.1

9	Методы задания спецификаций процессов. Определения, структура спецификации, правила описания. Сравнение методов спецификации процессов.	ОПК-2.В.1
10	Методы задания спецификаций процессов. Структурный естественный язык (псевдокод).	ОПК-2.В.1
11	Методы задания спецификаций процессов. FLOW-формы, структурограммы.	ОПК-2.В.1
12	Методы задания спецификаций процессов. Деревья и таблицы решений	ОПК-2.В.1
13	Проектирование Баз данных. ER-диаграммы. Основные элементы и их свойства.	ОПК-2.3.1 ОПК-2.У.1
14	Нормализация ER-диаграммы ИС. 1, 2, 3 нормальные формы.	ОПК-2.В.1
15	Нормализация ER-диаграммы ИС. Устранение связей типа М:М.	ОПК-2.В.1
16	Разработка структуры Базы данных. Табличные формы БД и их связь с ER- диаграммой.	ОПК-2.3.1 ОПК-2.У.1
17	Тестирование ПО	ОПК-2.3.1 ОПК-2.У.1
18	Основные принципы ООП и их реализация в языке C++	ОПК-2.3.1 ОПК-2.У.1
19	Уровни доступа к элементам класса в языке C++	ОПК-2.3.1 ОПК-2.У.1
20	Структура библиотеки STL языка C++	ОПК-2.3.1 ОПК-2.У.1
21	Библиотека STL: контейнеры	ОПК-2.3.1 ОПК-2.У.1
22	Библиотека STL: итераторы	ОПК-2.3.1 ОПК-2.У.1
23	Библиотека STL: алгоритмы	ОПК-2.3.1 ОПК-2.У.1

Перечень тем для курсового проектирования/выполнения курсовой работы представлены в таблице 17.

Таблица 17 – Перечень тем для курсового проектирования/выполнения курсовой работы

№ п/п	Примерный перечень тем для курсового проектирования/выполнения курсовой работы
	Учебным планом не предусмотрено

Вопросы для проведения промежуточной аттестации в виде тестирования представлены в таблице 18.

Таблица 18 – Примерный перечень вопросов для тестов

№ п/п	Примерный перечень вопросов для тестов	Код индикатора
1	<b>Инструкция: выберите один ответ</b>  Как называется период от момента появления идеи создания некоторого программного обеспечения до момента завершения его поддержки фирмой- разработчиком или фирмой, выполняющей сопровождение?	ОПК-2.3.1

	1. Эпоха 2. Жизненный план 3. Существование 4. Жизненный цикл																	
2	<b>Инструкция: выберите несколько ответов</b>  Какие из перечисленных терминов являются названиями моделей жизненного цикла программного обеспечения?  1. Спиральная 2. Водопроводная 3. Каскадная 4. Потокковая 5. Постоянная 6. Ключевая	ОПК-2.3.1																
3	<b>Инструкция:</b>  Для каждой диаграммы языка UML из левого столбца подберите фразу из правого столбца, которая характеризует эту диаграмму. <table><tr><td>A</td><td>Диаграмма вариантов использования (use cases)</td><td>1</td><td>Показывает последовательность событий, которые генерируются в системе при выполнении определенного сценария варианта использования</td></tr><tr><td>B</td><td>Диаграмма пакетов</td><td>2</td><td>Показывает аппаратные и программные компоненты системы</td></tr><tr><td>C</td><td>Диаграмма последовательностей</td><td>3</td><td>Представляет ожидаемое поведение системы с точки зрения пользователя</td></tr><tr><td>D</td><td>Диаграмма размещения (англ. deployment)</td><td>4</td><td>Позволяет разбить большую диаграмму на части.</td></tr></table>	A	Диаграмма вариантов использования (use cases)	1	Показывает последовательность событий, которые генерируются в системе при выполнении определенного сценария варианта использования	B	Диаграмма пакетов	2	Показывает аппаратные и программные компоненты системы	C	Диаграмма последовательностей	3	Представляет ожидаемое поведение системы с точки зрения пользователя	D	Диаграмма размещения (англ. deployment)	4	Позволяет разбить большую диаграмму на части.	ОПК-2.В.1
A	Диаграмма вариантов использования (use cases)	1	Показывает последовательность событий, которые генерируются в системе при выполнении определенного сценария варианта использования															
B	Диаграмма пакетов	2	Показывает аппаратные и программные компоненты системы															
C	Диаграмма последовательностей	3	Представляет ожидаемое поведение системы с точки зрения пользователя															
D	Диаграмма размещения (англ. deployment)	4	Позволяет разбить большую диаграмму на части.															
4	<b>Инструкция:</b> Запишите лексемы, используемые для объявления словаря в программе на языке C++, в правильном порядке:	ОПК-2.В.1																

	<ol style="list-style-type: none"> <li>1. тип данных ключа</li> <li>2. map</li> <li>3. тип данных значения</li> <li>4. &gt;</li> <li>5. имя словаря</li> <li>6. &lt;</li> <li>7. ,</li> </ol>	
5	<p><b>Инструкция:</b> Прочитайте текст и запишите развернутый обоснованный ответ.</p> <p>Что показывает контекстная диаграмма в рамках иерархии диаграмм потоков данных и какие элементы содержит?</p>	ОПК-2.В.1
6	<p><b>Инструкция: выберите один ответ</b></p> <p>Какой графический элемент используется в flow-форме?</p> <ol style="list-style-type: none"> <li>1. Волнистая линия</li> <li>2. Стрелка</li> <li>3. Прямоугольник</li> <li>4. Трапеция</li> </ol>	ОПК-2.В.1
7	<p><b>Инструкция:</b> Запишите лексемы, используемые для объявления вектора в программе на языке C++, в правильном порядке:</p> <ol style="list-style-type: none"> <li>1. тип данных значения</li> <li>2. vector</li> <li>3. &gt;</li> <li>4. имя вектора</li> <li>5. &lt;</li> </ol>	ОПК-2.В.1
8	<p><b>Инструкция:</b> Запишите в правильном порядке лексемы, используемые для объявления итератора в программе на языке C++:</p> <ol style="list-style-type: none"> <li>1. iterator</li> <li>2. ::</li> <li>3. описание контейнерного класса</li> <li>4. имя итератора</li> </ol>	ОПК-2.В.1
9	<p><b>Инструкция: выберите несколько ответов</b></p> <p>Какие из перечисленных ключевых слов управляют доступом к элементам класса в языке C++?</p> <ol style="list-style-type: none"> <li>1. new</li> <li>2. public</li> <li>3. private</li> </ol>	ОПК-2.В.1



	4. void 5. protected									
10	<b>Инструкция: выберите один ответ</b>  Как называется процесс, который сокращает избыточность хранения и устраняет аномалии в базах данных?  1. Верификация 2. Нормализация 3. Стабилизация 4. Оптимизация	ОПК-2.В.1								
11	<b>Инструкция: выберите один ответ</b>  Как называется группа шаблонов (паттернов) проектирования, предназначенных для создания новых объектов?  1. Завершающие 2. Структурные 3. Оптимизирующие 4. Порождающие	ОПК-2.В.1								
12	<b>Инструкция:</b>  Для каждой связи диаграммы классов языка UML из левого столбца подберите фразу из правого столбца, которая характеризует эту связь. <table><tr><td>A</td><td>Агрегация</td><td>1</td><td>Отношение «часть-целое» между двумя объектами, при котором контейнер управляет существованием своего содержимого. При уничтожении контейнера содержимое также уничтожается.</td></tr><tr><td>B</td><td>Обобщение</td><td>2</td><td>отношение «часть-целое» между двумя равноправными объектами, которые существуют независимо. При уничтожении контейнера его</td></tr></table>	A	Агрегация	1	Отношение «часть-целое» между двумя объектами, при котором контейнер управляет существованием своего содержимого. При уничтожении контейнера содержимое также уничтожается.	B	Обобщение	2	отношение «часть-целое» между двумя равноправными объектами, которые существуют независимо. При уничтожении контейнера его	ОПК-2.В.1
A	Агрегация	1	Отношение «часть-целое» между двумя объектами, при котором контейнер управляет существованием своего содержимого. При уничтожении контейнера содержимое также уничтожается.							
B	Обобщение	2	отношение «часть-целое» между двумя равноправными объектами, которые существуют независимо. При уничтожении контейнера его							

				содержимое остается.		
	C	Композиция	3	Связь между интерфейсом и классом.		
	D	Реализация	4	Отношение, показывающее, что один класс получает все элементы другого класса.		
13	<b>Инструкция: выберите несколько ответов</b>  Выберите названия типов связей (отношений) между классами в диаграмме классов языка UML  1. привязанность 2. зависимость 3. нейтральность 4. ассоциация 5. защищенность					ОПК-2.В.1
14	<b>Инструкция: Прочитайте текст и запишите развернутый обоснованный ответ.</b>  Объясните, почему исчерпывающее (полное) тестирование обычно невозможно.					ОПК-2.У.1
15	<b>Инструкция: выберите несколько ответов</b>  Выберите названия методов тестирования программного обеспечения  1. красный ящик 2. дырявый ящик 3. белый ящик 4. картонный ящик 5. черный ящик					ОПК-2.В.1
16	<b>Инструкция:</b>  Для каждого вида тестирования из левого столбца подберите соответствующую фразу из правого столбца.					ОПК-2.В.1
	A	Позитивное	1	Используются программные средства для выполнения		

			тестов и проверки их результатов		
	В	Автоматизированное	2	Использование почти готовой версии программного продукта добровольцами	
	С	Бета-тестирование	3	Тестирование изолированных частей программы	
	D	Модульное	4	В тесте используются только корректные данные.	
17	<b>Инструкция: выберите один ответ</b>  Какой графический символ используется для обозначения варианта использования на диаграммах языка UML?  1. Овал 2. Прямоугольник 3. Ромб 4. Круг				ОПК-2.В.1
18	<b>Инструкция: выберите несколько ответов</b>  Выберите названия диаграмм языка UML.  1. диаграмма коробок 2. диаграмма деятельности 3. диаграмма пакетов 4. диаграмма компонентов 5. контекстная диаграмма				ОПК-2.В.1
19	<b>Инструкция: выберите один ответ</b>  Как называется класс, который содержит или наследует хотя бы одну чистую виртуальную функцию (язык C++)?  1. Контейнер 2. Конструктор 3. Шаблон 4. Абстрактный				ОПК-2.В.1
20	<b>Инструкция: выберите несколько ответов</b>  Какие шаблоны можно создать в программе на языке C++?				ОПК-2.В.1

	1. отношения 2. деятельности 3. программы 4. функции 5. класса	
--	--	--

Перечень тем контрольных работ по дисциплине обучающихся заочной формы обучения, представлены в таблице 19.

Таблица 19 – Перечень контрольных работ

№ п/п	Перечень контрольных работ
1	Знакомство с контейнерными классами библиотеки STL языка C++
2	Знакомство с шаблонами проектирования

10.4. Методические материалы, определяющие процедуры оценивания индикаторов, характеризующих этапы формирования компетенций, содержатся в локальных нормативных актах ГУАП, регламентирующих порядок и процедуру проведения текущего контроля успеваемости и промежуточной аттестации обучающихся ГУАП.

#### 11. Методические указания для обучающихся по освоению дисциплины

Целью дисциплины «Технология программирования» является получение студентами необходимых знаний, умений и навыков в области проектирования информационных систем в различных прикладных областях с использованием современных методологий и технологических средств проектирования.

Одной из целей является приобрести умение провести декомпозицию крупной прикладной задачи на более мелкие, с целью организации работы коллектива разработчиков, а также знать современные методики и уметь оценить качество разработанного программного обеспечения, и, используя различные модели ЖЦ ПО, знать возможности организации эффективной работы такого коллектива разработчиков. Важным также является получение студентами практических навыков работы с CASE-средствами различного уровня.

11.1. Методические указания для обучающихся по освоению лекционного материала.

Основное назначение лекционного материала – логически стройное, системное, и ясное изложение учебного материала. В начальных лекциях необходимо раскрыть основные цели данной дисциплины, как части профессиональной подготовки по конкретной специальности, показать реальные задачи и методы их решения на базе полученных знаний, рассмотреть историю возникновения и перспективы развития данной дисциплины. В лекциях всех разделов также целесообразно рассматривать примеры применения излагаемого теоретического материала для решения конкретных прикладных задач, перспективы использования полученных знаний и навыков в последующей работе выпускников. Планируемы результаты при освоении обучающимся лекционного материала:

- получение современных, целостных, взаимосвязанных знаний в области проектирования ИС;
- развитие профессионально–деловых качеств и самостоятельного творческого мышления;

- получение знаний о современном уровне развития науки «Технология программирования», а также о прогнозе и потребностях ее развития на ближайшие годы;
- получение практического понимания всех используемых терминов и понятий.

Лекционные материалы доступны в электронном виде на сервере кафедры №43 по адресу: \\Dcbm\учебные пособия\Технология программирования

11.2. Методические указания для обучающихся по участию в семинарах *(если предусмотрено учебным планом по данной дисциплине)*

11.3. Методические указания для обучающихся по выполнению лабораторных работ

В ходе выполнения лабораторных работ обучающийся должен развить творческие навыки и умение переходить от практической прикладной проблемы к научной, формализованной постановке задачи проектирования ИС, овладеть современными методиками проектирования, а также получить навыки работы с современными пакетами программ, позволяющими частично автоматизировать процесс проектирования ПО ИС. Выполнение лабораторных работ состоит из аналитической и проектной частей при разработке различного типа диаграмм, а также экспериментально-практической при использовании современных CASE-средств. Выполнение лабораторных работ обучающимся является неотъемлемой частью изучения дисциплины, определяемой учебным планом и относится к средствам, обеспечивающим решение следующих основных задач у обучающегося:

- приобретение навыков анализа процессов и объектов прикладных задач;
- закрепление, развитие и детализация теоретических знаний, изучаемых в рамках данной дисциплины.

#### Задание и требования к проведению лабораторных работ

Лабораторные работы в основном связаны с созданием диаграмм того или иного вида. Требования содержат необходимое количество элементов на диаграмме.

#### Структура и форма отчета о лабораторной работе

Отчет должен содержать следующие разделы: титульный лист, цель работы, постановка задачи, результаты выполнения заданий по лабораторной работе

#### Требования к оформлению отчета о лабораторной работе

Отчет о выполнении лабораторной работы должен быть представлен в электронном виде (файл). В отчет включается диаграмма и комментарии к ней.

Методические указания по выполнению лабораторных работ доступны в электронном виде на сервере кафедры №43 по адресу: \\Dcbm\учебные пособия\Технология программирования

11.4. Методические указания для обучающихся по прохождению самостоятельной работы

В ходе выполнения самостоятельной работы, обучающийся выполняет работу по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

В процессе выполнения самостоятельной работы, у обучающегося формируется целесообразное планирование рабочего времени, которое позволяет им развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивает высокий

уровень успеваемости в период обучения, помогает получить навыки повышения профессионального уровня.

Методическими материалами, направляющими самостоятельную работу обучающихся являются:

- учебно-методический материал по дисциплине;
- методические указания по выполнению контрольных работ (для обучающихся по заочной форме обучения).

Методическими материалами, направляющими самостоятельную работу обучающихся, являются учебно-методический материал по дисциплине.

11.5. Методические указания для обучающихся по прохождению текущего контроля успеваемости.

Текущий контроль успеваемости предусматривает контроль качества знаний обучающихся, осуществляемого в течение семестра с целью оценивания хода освоения дисциплины.

Текущий контроль заключается в проверке выполнения контрольных и лабораторных работ и выставлении баллов. Необходима устная защита лабораторных работ. Требования изложены в соответствующих методических указаниях. Полученные баллы влияют на оценку при проведении промежуточной аттестации.

11.6. Методические указания для обучающихся по прохождению промежуточной аттестации.

Промежуточная аттестация обучающихся предусматривает оценивание промежуточных и окончательных результатов обучения по дисциплине. Она включает в себя:

- экзамен – форма оценки знаний, полученных обучающимся в процессе изучения всей дисциплины или ее части, навыков самостоятельной работы, способности применять их для решения практических задач. Экзамен, как правило, проводится в период экзаменационной сессии и завершается аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

- дифференцированный зачет – это форма оценки знаний, полученных обучающимся при изучении дисциплины, при выполнении курсовых проектов, курсовых работ, научно-исследовательских работ и прохождении практик с аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Экзамен проходит в устной форме. Студенты отвечают по вопросам в билетах. Дифференцированный зачет выставляется либо по результатам работы в семестре, либо по результатам устных ответов на вопросы к зачету.

Лист внесения изменений в рабочую программу дисциплины

Дата внесения изменений и дополнений. Подпись внесшего изменения	Содержание изменений и дополнений	Дата и № протокола заседания кафедры	Подпись зав. кафедрой