


МИНОБРНАУКИ РОССИИ  
АСТРАХАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

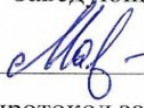
СОГЛАСОВАНО  
Руководитель ОПОП

 А. Н. Марьенков

«11» мая 2021 г.

УТВЕРЖДАЮ

Заведующий кафедрой ИБиЦТ

 А. Н. Марьенков  
протокол заседания кафедры №10

«11» мая 2021 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ  
ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ**

Составители

**Карпенко А.В., ассистент кафедры  
ИБиЦТ**

Направление подготовки /  
специальность

**09.03.02 ИНФОРМАЦИОННЫЕ СИСТЕМЫ И  
ТЕХНОЛОГИИ**

Направленность (профиль) ОПОП

**Безопасность информационных систем**

Квалификация (степень)

**бакалавр**

Форма обучения

**очно-заочная**

Год приема

**2021**

Курс

**3**

Астрахань– 2021 г.

## **1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)**

**1.1. Целями освоения дисциплины «Технология программирования»** является изучение и освоение базовых понятий, методов и приемов программирования на языке программирования C++ в основном в парадигме процедурного программирования.

### **1.2. Задачи освоения дисциплины:**

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 обеспечения

Познакомить обучающихся с существующими подходами к оценке качества создания программного обеспечения.

## **2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОПОП**

**2.1. Учебная дисциплина технологии и методы программирования** относится к обязательной части (базовые дисциплины).

**2.2. Для изучения данной учебной дисциплины (модуля) необходимы следующие знания, умения и навыки, формируемые предшествующими дисциплинами (модулями):**

- Информатика

Знания: базовые понятия информатики и вычислительной техники; понятие информационной системы и информационной технологии; технические и программные средства реализации информационных процессов; основные устройства, входящие в состав ЭВМ, их назначение и характеристики; формы представления и преобразования информации в компьютере.

Умения: применять вычислительную технику для решения практических задач; разработать алгоритм поставленной задачи.

Навыки работы на персональном компьютере.

- Основы программирования

Знания: основные структуры данных, используемые в языках программирования; структуру программ; нахождение значения выражения.

Умения: создавать схему алгоритма для задачи; решать задачи с помощью условных операторов, циклов.

Навыки и (или) опыт деятельности: в области применения функций, работы с файлами.

- Алгоритмы и структуры данных

Знания: способы применения различных структур данных для решения определённых задач; теория графов.

Умения: использовать возможности структур при построении алгоритмов решения задач.

Навыки и (или) опыт деятельности: решении задач оптимизации.

**2.3. Перечень последующих учебных дисциплин (модулей), для которых необходимы знания, умения и навыки, формируемые данной учебной дисциплиной (модулем):**

– Большие данные

– Моделирование систем.

### 3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

Процесс изучения дисциплины (модуля) направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО и ОПОП ВО по данному направлению подготовки (специальности):

а) общепрофессиональных (ОПК):

ОПК–5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем.

ОПК–6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий.

**Таблица 1**

#### *Декомпозиция результатов обучения*

Код компетенции	Планируемые результаты освоения дисциплины (модуля)		
	Знать (1)	Уметь (2)	Владеть (3)
ОПК-5	ИОПК-5.1.1 основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем.	ИОПК-5.2.1 выполнять параметрическую настройку информационных и автоматизированных систем.	ИОПК-5.3.1 инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем.
ОПК-6	ИОПК-6.1.1 методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий.	ИОПК-6.2.1 применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий.	ИОПК-6.3.1 программирования, отладки и тестирования прототипов программно-технических комплексов задач.

### 4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Объем дисциплины «Технология программирования» составляет 4 зачетные единицы, в том числе 18 часов на контактную работу обучающихся с преподавателем (из них 18 часов – лабораторные работы), и 126 часов – на самостоятельную работу обучающихся.

**Таблица 2. Структура и содержание дисциплины (модуля)**

№ п/п	Наименование раздела (темы)	5	Неделя семестра	Контактная работа (в часах)			Самостоят. работа		Формы текущего контроля успеваемости (по неделям семестра)  Форма промежуточной аттестации (по семестрам)
				Л	ПЗ	ЛР	КР	СР	
1.	Тема.1 Структуры	5	1-3			2		14	отчет по

№ п/п	Наименование раздела (темы)		Неделя семестра	Контактная работа (в часах)			Самостоят. работа		Формы текущего контроля успеваемости ( <i>по неделям семестра</i> )  Форма промежуточной аттестации ( <i>по семестрам</i> )
				Л	ПЗ	ЛР	КР	СР	
	данных, используемые в программах.								лабораторной работе 1
2.	Тема 2. Структурное программирование.	5	4-6			2		14	отчет по практическому заданию 1
3.	Тема 3. Тестирование и отладка программного обеспечения	5	7-9			2		14	отчет по лабораторной работе 2
4.	Тема 4. Объектно- ориентированное программирование	5	10- 13			2		14	отчет по практическому заданию 2
5.	Тема 5. Приемы обеспечения технологичности программных продуктов. Методологии разработки.	5	14- 17			2		14	отчет по практическому заданию 3
6.	Тема 6. Критерии эффективной архитектуры программных продуктов	5	1-4			2		14	отчет по лабораторной работе 3
7.	Тема 7. Этапы проектирования ПО	5	5-10			2		14	отчет по практическому заданию 4
8.	Тема 8. Паттерны проектирования	5	11- 13			2		14	отчет по лабораторной работе 4
9.	Тема 9. Разработка пользовательского интерфейса. Оценка качества программного обеспечения	5	14- 16			2		14	отчет по лабораторной работе 5
	Итого 5 семестр								<b>Экзамен</b>

№ п/п	Наименование раздела (темы)		Неделя семестра	Контактная работа (в часах)			Самостоят. работа		Формы текущего контроля успеваемости ( <i>по неделям семестра</i> )  Форма промежуточной аттестации ( <i>по семестрам</i> )
				Л	ПЗ	ЛР	КР	СР	
	Итого		<b>144</b>			<b>18</b>		<b>126</b>	

Условные обозначения:

Л – занятия лекционного типа; ПЗ – практические занятия, ЛР – лабораторные работы;  
КР – курсовая работа; СР – самостоятельная работа по отдельным темам

**Таблица 3. Матрица соотнесения тем/разделов  
учебной дисциплины/модуля и формируемых в них компетенций**

Темы, разделы дисциплины	Кол-во часов	Компетенции			
		ОПК-5	ОПК-6		общее количество компетенций
Тема.1 Структуры данных, используемые в программах.	16	+	+		2
Тема 2. Структурное программирование	15	+	+		2
Тема 3. Тестирование и отладка программного обеспечения	15	+	+		2
Тема 4. Объектно- ориентированное программирование	15	+	+		2
Тема 5. Приемы обеспечения технологичности программных продуктов. Методологии разработки.	17	+	+		2
Тема 6. Критерии эффективной архитектуры программных	17	+	+		2

продуктов					
Тема 7. Этапы проектирования ПО	17	+	+		2
Тема 8. Паттерны проектирования	16	+	+		2
Тема 9. Разработка пользовательского интерфейса. Оценка качества программного обеспечения	16	+	+		2
<b>Итого</b>	<b>144</b>				

### Содержание дисциплины

Тема.1 Структуры данных, используемые в программах.

Рассматривается вопрос, что такое технология программирования, жизненный цикл, модульное программирование, объектно-ориентированное программирование. Приводится краткое сравнение ООП и структурного программирования. В качестве примеров структур данных рассматриваются списки. линейные и кольцевые списки (однонаправленные и двунаправленные списки). Кратко поясняется термин графа, дерева и множества, а также способы их применения в программировании.

Тема 2. Структурное программирование.

Сравнение структурного и «неструктурное» программирования. Приводятся примеры стиля оформления программы. Средства описания структурных алгоритмов. Различают три вида вычислительного процесса, реализуемого программами:

- а) линейный (для получения результата необходимо выполнить операции в определенной последовательности);
- б) разветвленный (конкретная последовательность операций зависит от значений одной или нескольких переменных);
- в) циклический (для получения результата некоторые действия необходимо выполнить несколько раз).

Эффективность и технологичность. Программирование «с защитой от ошибок». Сквозной структурный контроль.

Тема 3. Тестирование и отладка программного обеспечения.

Классификация ошибок при отладке: синтаксические ошибки (причиной таких ошибок могут быть неправильно написанные ключевые слова, неверно примененные разделители или недопустимые комбинации операторов); ошибки в структуре программы (ошибки такого рода появляется в результате неправильного написания многострочного оператора); ошибки, возникающие во время выполнения программы (это ошибки, возникающие во время работы программы, например, при выполнении деления на ноль или при попытке чтения из несуществующего на диске файла). Методы отладки программного обеспечения – статические и динамические.

Тема 4. Объектно-ориентированное программирование.

Рассматривается понятие класса и объекта класса. Класс - представляет собой шаблон для создания объектов, обеспечивающий начальные значения состояний: инициализация полей-переменных и реализация поведения функций или методов. Объект в программировании — некоторая сущность в цифровом пространстве, обладающая

определённым состоянием и поведением, имеющая определенные свойства (атрибуты) и операции над ними (методы). Основные свойства ООП – наследование, инкапсуляция, полиморфизм. UML - стандартный язык описания разработки программных продуктов с использованием объектного подхода. Определение «вариантов использования». Построение концептуальной модели предметной области - это модель, представленная множеством понятий и связей между ними, определяющих смысловую структуру рассматриваемой предметной области или её конкретного объекта. Разработка структуры программного обеспечения при объектном подходе. Определение отношений между объектами. Уточнение отношений классов. Проектирование классов. Проектирование размещения программных компонентов для распределенных программных систем. Особенность спиральной модели разработки.

Тема 5. Приемы обеспечения технологичности программных продуктов.

Этапы развития технологии программирования. Модели жизненного цикла программного обеспечения

Каскадная модель жизненного цикла программного обеспечения (водопад)

Каскадная модель с промежуточным контролем (водоворот)

V модель (разработка через тестирование)

Модель на основе разработки прототипа

Спиральная модель жизненного цикла программного обеспечения

Понятие эффективности и технологичности программного обеспечения. Модули и их свойства. Средства описания структурных алгоритмов (псевдокоды, блок-схемы)

Тема 5. Разработка пользовательского интерфейса. Оценка качества программного обеспечения

Интерфейс — общая граница между двумя функциональными объектами, требования к которой определяются стандартом; совокупность средств, методов и правил взаимодействия (управления, контроля и т. д.) между элементами системы.

Пользовательский интерфейс (UI) — это способ, которым вы выполняете какую-либо задачу с помощью какого-либо продукта, а именно совершаемые вами действия и то, что вы получаете в ответ.

Типы пользовательских интерфейсов и этапы их разработки:

Создание мокапа

User Flow Diagram (карта экранов)

Утверждение структуры

Выбор стиля UI

Интерактивный прототип

Утверждение результата

Тема 6. Критерии эффективной архитектуры программных продуктов

Эффективность системы является одним из основных критериев и её характеризуют:

Надёжность

Безопасность

Производительность

Масштабируемость

Высокая сопряжённость и слабая связность Когда говорят, что в системе должна быть высокая сопряжённость, это означает эта сопряжённость должна быть внутри модуля. Он должен быть плотным, консистентным и не содержать ничего лишнего. Каждый модуль выполняет свою функцию – одну. Если он выполняет много функций, то его нужно разделить

на несколько подмодулей. Любой элемент, который вносится в модуль, должен иметь непосредственную связь с другими частями этого модуля.

При этом должна присутствовать слабая связность. Это означает что между модулями должна быть слабая связь. Чем меньше они друг с другом связаны тем лучше. Они должны быть независимы, в идеале. Важно отменить, что такое никогда не произойдет на практике.

#### Тема 7. Этапы проектирования ПО

В списке этапов не указана разработка ПО, потому что это эти шаги имеют отношение непосредственно к проектированию. Разработка находится там внутри них на определяемой разработчиком шаге:

1. Формирование требований
2. Разработка концепции
3. Техническое задание
4. Эскизный проект
5. Технический проект
6. Рабочая документация
7. Поставка или ввод в эксплуатацию
8. Сопровождение

Документ технического задания является достаточно строгим, потому что регламентируется ГОСТ. ТЗ это финальная точка по которой происходит реализация. В момент когда написано ТЗ, ничего больше менять нельзя. ТЗ это финальная, четкая формализация того как вы общались с заказчиком. Кроме того, оно идет в комплекте с договором где указаны сроки разработки и стоимость.

#### Тема 8. Паттерны проектирования.

Это повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста. Обычно шаблон не является законченным образцом, который может быть прямо преобразован в код; это лишь пример решения задачи, который можно использовать в различных ситуациях. Объектно-ориентированные шаблоны показывают отношения и взаимодействия между классами или объектами, без определения того, какие конечные классы или объекты приложения будут использоваться.

Порождающие шаблоны— шаблоны проектирования, которые абстрагируют процесс инстанцирования. Они позволяют сделать систему независимой от способа создания, композиции и представления объектов. Шаблон, порождающий классы, использует наследование, чтобы изменять инстанцируемый класс, а шаблон, порождающий объекты, делегирует инстанцирование другому объекту.

Структурные шаблоны определяют различные сложные структуры, которые изменяют интерфейс уже существующих объектов или его реализацию, позволяя облегчить разработку и оптимизировать программу.

#### Тема 9. Разработка пользовательского интерфейса. Оценка качества программного обеспечения

Интерфейс — общая граница между двумя функциональными объектами, требования к которой определяются стандартом; совокупность средств, методов и правил взаимодействия (управления, контроля и т. д.) между элементами системы.

Пользовательский интерфейс (UI) — это способ, которым вы выполняете какую-либо задачу с помощью какого-либо продукта, а именно совершаемые вами действия и то, что вы получаете в ответ.

Типы пользовательских интерфейсов и этапы их разработки:

- Создание мокапа
- User Flow Diagram (карта экранов)

- Утверждение структуры
- Выбор стиля UI
- Интерактивный прототип
- Утверждение результата

## **5. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**

### **5.1. Указания по организации и проведению лекционных, практических (семинарских) и лабораторных занятий с перечнем учебно-методического обеспечения**

Электронный учебно-методический комплекс размещённый на образовательном портале moodle, включает теоретические материалы, порядок выполнения лабораторных работ, список рекомендованной литературы.

Студенты выполняют лабораторные работы и прикрепляют свой ответ на образовательном портале moodle. После проверки преподавателем, выставляется оценка или оставляется комментарий с замечаниями и рекомендациями.

Учебные ресурсы и тренинги ресурса Intuit.ru, который обеспечивает большой спектр курсов в области компьютерных наук для образовательных учреждений. При подготовке к лекционным занятиям необходимо воспользоваться учебно-методической литературой. Лекции необходимо проводить с использованием презентаций, созданных в Microsoft PowerPoint.

При подготовке к лабораторным занятиям необходимо воспользоваться учебно-методической литературой, а также пользоваться ресурсами сети Интернет.

#### **Методические рекомендации по выполнению лабораторных и контрольных работ, проведению экзамена**

##### **Отчет по лабораторной работе**

Отчет по лабораторной работе представляется в электронном виде. Защита отчета проходит в форме доклада студента по выполненной работе и ответов на вопросы преподавателя. В случае, если оформление отчета и поведение студента во время защиты соответствуют указанным требованиям, студент получает максимальное количество баллов.

Основаниями для снижения количества баллов в диапазоне от max до min являются:

- отсутствие списка использованной литературы,
- небрежное выполнение,
- отсутствие выводов.

Отчет не может быть принят и подлежит доработке в случае:

- отсутствия необходимых разделов,
- отсутствия необходимого графического материала,
- неверных результатов расчета.

В отчете по выполненной лабораторной работе должны быть указаны:

- тема лабораторной работы,
- пакет документов в соответствии с темой лабораторной работы,
- использованная литература.

##### **Экзамен**

Экзамен заключается в письменном ответе на 2 теоретических вопроса и устном собеседовании по каждому теоретическому вопросу.

Основаниями для снижения оценки за теоретический вопрос являются:

- небрежное выполнение;
- неполный ответ;
- наличие мелких неточностей или незначительных искажений фактов;
- неточные объяснения при собеседовании;

- отсутствие ответов на заданные при собеседовании вопросы.

Оценивание студентов на экзамене осуществляется в соответствии с требованиями и критериями 100-балльной шкалы. Учитываются как результаты текущего контроля, так и знания, навыки и умения, непосредственно показанные студентами в ходе экзамена.

## 5.2. Указания для обучающихся по освоению дисциплины (модулю)

Самостоятельная работа студентов подразумевает чтение и анализ технической литературы по предмету, документации на программное обеспечение, самостоятельное создание схемы алгоритма для задачи, проведение отладки и тестирования созданных модулей, выполнение индивидуального домашнего задания по одной из выбранных предметных областей.

Таблица 4. Содержание самостоятельной работы обучающихся

Номер раздела (темы)	Темы/вопросы, выносимые на самостоятельное изучение	Кол-во часов	Формы работы
Тема 1.	Графы. Представление графа в компьютерной системе. Массив ребер. Матрица смежности. Деревья. Множества. Ассоциативные массивы и множества.	14	Изучение учебной литературы, подготовка к занятиям
Тема 2.	Классификация программных продуктов по функциональному признаку. Основные эксплуатационные требования к программным продуктам.	14	Изучение учебной литературы, подготовка к занятиям
Тема 3.	Виды контроля качества разрабатываемого программного обеспечения. Ручной контроль программного обеспечения. Структурное тестирование. Функциональное тестирование.	14	Изучение учебной литературы, подготовка к занятиям
Тема 4.	UML - стандартный язык описания разработки программных продуктов с использованием объектного подхода.	14	Изучение учебной литературы, подготовка к занятиям
Тема 5.	Порождающий паттерн Prototype, Singleton. Поведенческие и структурные паттерны.	14	Изучение учебной литературы, подготовка к занятиям
Тема 6.	Понятие технологичности программного обеспечения. Модульное программирование. Модули и их свойства.	14	Изучение учебной литературы, подготовка к занятиям
Тема 7.	Пользовательские интерфейсы прямого манипулирования и их проектирование. Интеллектуальные элементы пользовательских интерфейсов.	14	Изучение учебной литературы, подготовка к занятиям
Тема 8.	Графы. Представление графа в	14	Изучение учебной

	компьютерной системе. Массив ребер. Матрица смежности. Деревья. Множества. Ассоциативные массивы и множества.		литературы, подготовка к занятиям
Тема 9.	Документирование программного обеспечения	14	Внеаудиторная, изучение учебных пособий

### **5.3. Виды и формы письменных работ, предусмотренных при освоении дисциплины, выполняемые обучающимися самостоятельно - проект.**

#### **Задание для выполнения проекта**

Студенты в составе группы реализуют проект по разработке программного обеспечения. В задачи проекта входят: поиск заказчика и формулировка проекта, генерация требований к проекту, проектирование, разработка, тестирование и документирование программного обеспечения; презентация результатов проекта

Студенты разбиваются на группы по 4-6 человек и реализуют проект по разработке программного обеспечения. На первом этапе группа должны найти заказчика проекта, определить подлежащую решению с помощью разрабатываемого ПО проблему и сформулировать тематику проекта. Заказчиком может являться произвольная бизнес-структура либо одно из подразделений университета. Проект должен быть направлен на решение реальной задачи.

Работа над проектом предполагает использование гибких методологий разработки ПО, но при этом не исключает полностью этапа предварительного эскизного проектирования.

При работе над проектом студенты осуществляют сбор и документирование требований к проекту, строят базовые модели системы с использованием UML.

При разработке проекта студенты должны использовать методы обеспечения качества программного продукта: модульное и интеграционное тестирование, системы управления версиями, системы отслеживания ошибок.

По завершении проекта на него подготавливается программная документация и студенты выполняют презентацию результатов проекта.

#### **Критерии оценки проекта:**

– оценка «отлично» выставляется обучающемуся, если студент представил проект в соответствии с методическими указаниями, информация в проекте сформулирована обоснованно, логично и последовательно, применен творческий подход;

– оценка «хорошо» выставляется обучающемуся, если студент представил проект в соответствии с методическими указаниями, информация в проекте сформулирована обоснованно, формулировки конкретные, допущены некоторые неточности, имеется одна негрубая ошибка;

– оценка «удовлетворительно» выставляется обучающемуся, если студент представил проект в соответствии с методическими указаниями, информация в проекте сформулирована с

нарушением логики, не полная, формулировка общая или неполная, имеются одна или две негрубые ошибки;

– оценка «неудовлетворительно» выставляется обучающемуся, если студент не представил проект или выполнил ее неверно, без использования методических указаний, обоснования неверные, сделаны грубые ошибки.

## 6. ОБРАЗОВАТЕЛЬНЫЕ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

### 6.1. Образовательные технологии

В соответствии с требованиями ФГОС ВО по направлению подготовки бакалавров в рамках изучения дисциплины предусмотрено использование в учебном процессе следующих активных и интерактивных форм проведения занятий:

№ п/п	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
1	2	3	4
1	Лекции	с использованием мультимедийных роликов	
2	Контрольная работа	Средство проверки умений применять полученные знания для решения задач определенного типа по теме или разделу	Комплект контрольных заданий по вариантам
3	Лабораторная работа	Средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п. Может выполняться в индивидуальном порядке или группой обучающихся.	Задания для лабораторных работ

Цели курса достигаются путём сочетания комплекса методов обучения, включающих лекции и лабораторные работы, выполняемые на ЭВМ.

Лабораторные работы на ЭВМ ориентированы на формирование деятельностных компетентностей. Они заключаются в выполнении цикла лабораторных работ. В процессе выполнения лабораторных работ достигаются следующие цели:

- закрепляются теоретические познания, полученные на лекциях, актуализируется их практическая значимость, закрепляется мотивация к освоению курса;
- студент приобретает практические навыки программирования на языках высокого уровня;
- приобретаются начальные навыки использования сред разработки программных проектов;
- формируется навык выявления ошибочных и нестандартных ситуаций и реагирования на них.

## 6.2. Информационные технологии

Название информационной технологии	Темы, разделы дисциплины	Краткое описание применяемой технологии
Использование возможностей Интернета в учебном процессе	1 – 9	Проведение входного, текущего и рейтингового контроля знаний учащихся (в системах дистанционного обучения)
Использование возможностей электронной почты преподавателя	1 – 9	Подготовка к защите отчетов по лабораторным работам
Использование средств представления учебной информации	1 – 9	Использование мультимедийной презентации

## 6.3. Перечень программного обеспечения и информационных справочных систем

### – Лицензионное программное обеспечение

Название программного обеспечения	Назначение
Платформа дистанционного обучения LMS Moodle	Виртуальная обучающая среда
Mozilla FireFox	Браузер
Microsoft Office 2013, Microsoft Office Project 2013, Microsoft Office Visio 2013	Пакет офисных программ
7-zip	Архиватор
Microsoft Windows 7 Professional	Операционная система
Kaspersky Endpoint Security	Средство антивирусной защиты
PascalABC.NET	Среда разработки
PyCharm EDU	Среда разработки

### б) Информационные справочные системы:

#### – Современные профессиональные базы данных, информационных справочных систем

Электронно-библиотечная система elibrary. <http://elibrary.ru>.

## 7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

### 7.1. Паспорт фонда оценочных средств

При проведении текущего контроля и промежуточной аттестации по дисциплине (модулю) «Технологии программирования» проверяется сформированность у обучающихся компетенций, указанных в разделе 3 настоящей программы. Этапность формирования данных компетенций в процессе освоения образовательной программы определяется последовательным освоением дисциплин (модулей) и прохождением практик, а в процессе

освоения дисциплины (модуля) – последовательным достижением результатов освоения содержательно связанных между собой разделов, тем.

**Таблица 5**  
**Соответствие разделов, тем дисциплины (модуля),**  
**результатов обучения по дисциплине (модулю) и оценочных средств**

№ п/п	Контролируемые разделы дисциплины (модуля)	Код контролируемой компетенции (компетенций)	Наименование оценочного средства
1	Тема.1 Структуры данных, используемые в программах.	ОПК-5 ОПК-6	индивидуальное собеседование по лабораторной работе 1
2	Тема 2. Структурное программирование.	ОПК-5 ОПК-6	индивидуальное собеседование по практическому заданию 1
3	Тема 3. Тестирование и отладка программного обеспечения	ОПК-5 ОПК-6	индивидуальное собеседование по лабораторной работе 2
4	Тема 4. Объектно- ориентированное программирование	ОПК-5 ОПК-6	индивидуальное собеседование по практическому заданию 2
5	Тема 5. Приемы обеспечения технологичности программных продуктов. Методологии разработки.	ОПК-5 ОПК-6	индивидуальное собеседование по практическому заданию 3
6	Тема 6. Критерии эффективной архитектуры программных продуктов	ОПК-5 ОПК-6	индивидуальное собеседование по лабораторной работе 3
7	Тема 7. Этапы проектирования ПО	ОПК-5 ОПК-6	индивидуальное собеседование по практическому заданию 4
8	Тема 8. Паттерны проектирования	ОПК-5 ОПК-6	индивидуальное собеседование по лабораторной работе 4
9	Тема 9. Разработка пользовательского интерфейса. Оценка качества программного обеспечения	ОПК-5 ОПК-6	индивидуальное собеседование по лабораторной работе 5. Вопросы к экзамену

*Рекомендуемые типы контроля для оценивания результатов обучения.*

Для оценивания результатов обучения в виде **знаний** используются следующие типы контроля:

- тестирование;

- индивидуальное собеседование,
- письменные ответы на вопросы.

Тестовые задания должны охватывать содержание всего пройденного материала. Индивидуальное собеседование, письменная работа проводятся по разработанным вопросам по отдельному учебному элементу программы (дисциплине).

Для оценивания результатов обучения в виде **умений и владений** используются следующие типы контроля:

- практические контрольные задания (далее – ПКЗ), включающих одну или несколько задач (вопросов) в виде краткой формулировки действий (комплекса действий), которые следует выполнить, или описание результата, который нужно получить.

По сложности ПКЗ разделяются на простые и комплексные задания.

Простые ПКЗ предполагают решение в одно или два действия. К ним можно отнести: простые ситуационные задачи с коротким ответом или простым действием; несложные задания по выполнению конкретных действий. Простые задания применяются для оценки умений. Комплексные задания требуют многоходовых решений как в типичной, так и в нестандартной ситуациях. Это задания в открытой форме, требующие поэтапного решения и развернутого ответа, в т.ч. задания на индивидуальное или коллективное выполнение проектов, на выполнение практических действий или лабораторных работ. Комплексные практические задания применяются для оценки владений.

Типы практических контрольных заданий:

- задания на установление правильной последовательности, взаимосвязанности действий, выяснения влияния различных факторов на результаты выполнения задания;
- установление последовательности (описать алгоритм выполнения действия),
- нахождение ошибок в последовательности (определить правильный вариант последовательности действий);
- указать возможное влияние факторов на последствия реализации умения и т.д.
- задания на принятие решения в нестандартной ситуации (ситуации выбора, многоальтернативности решений, проблемной ситуации);
- задания на оценку последствий принятых решений;
- задания на оценку эффективности выполнения действия.

## 7.2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Каждая лабораторная работа оценивается в диапазоне от 0 до 100 баллов.

Сумма баллов по выполнению лабораторной работы	Критерии оценивания
90-100	Все задания лабораторной работы выполнены в полном объеме. Программа работает верно на всех вариантах тестовых данных. Алгоритмы в коде программы реализованы корректно.
85-89	В программе реализованы все функции, заявленные в задании лабораторной работы.
75-84	

70-74	Программа не работает корректно на всех
65-69	В разработанной программе отсутствует реализация всех функций, заявленных в задании лабораторной работы. Программа не работает корректно на всех вариантах входных данных
60-64	
59 и ниже	Разработанная согласно заданию лабораторной работы, программа не предоставлена либо не запускается в интегрированной среде разработки Eclipse.

Практическое задание оценивается в диапазоне от 0 до 100 баллов.

Сумма баллов по выполнению лабораторной работы	Критерии оценивания
90-100	Все пункты технического задания описаны и раскрыты.
85-89	В тексте технического задания не описаны или не раскрыты полностью некоторые пункты.
75-84	
70-74	
65-69	В тексте технического задания отсутствует описание всех пунктов. Либо они недостаточно раскрыты в рамках выбранной предметной области
60-64	
59 и ниже	Текст технического задания по выбранном тематике не предоставлен.

По итогам выполнения лабораторных работ и практического задания, набранные балы суммируются и полученное число делится на общее количество заданий. Полученное число переводится в четырёх бальную систему оценивания соответствии с БАРС.

**Таблица 6.**

Критерии оценивания результатов обучения

5 «отлично»	Полностью выполнил все лабораторные работы и практическое задание в полном объёме.
4 «хорошо»	Предоставил выполненные лабораторные работы и практическое задание с небольшими недочётами. Либо не предоставил лишь некоторые выполненные лабораторные работы.
3 «удовлетворительно»	Предоставил некоторые выполненные лабораторные работы. Либо задания были выполнены не в полном объёме, имелись недочёты в программном коде.
2 «неудовлетворительно»	Не предоставил выполненные задания либо выполнял их полностью неверно.

**Таблица 7**

### Показатели оценивания результатов обучения в виде умений и владений

Шкала оценивания	Критерии оценивания
5 «отлично»	Демонстрирует способность применять знание теоретического материала при выполнении лабораторных работ, правильно выполняет задания лабораторных работ в полном объеме
4 «хорошо»	Демонстрирует способность применять знание теоретического материала при выполнении лабораторных работ, последовательно и правильно выполняет задания, допускает единичные синтаксические или логически ошибки, исправляемые после замечания преподавателя.
3 «удовлетворительно»	Демонстрирует отдельные, несистематизированные навыки, не способен применить знание особенностей структур данных или алгоритмов при выполнении лабораторных работ, не выполняет работу в полном объеме.
2 «неудовлетворительно»	Не способен правильно выполнить задания лабораторных работ.

### 7.3. Контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности

#### Тема 1. Структуры данных, используемые в программах.

##### Лабораторная работа 1. Структуры данных.

##### Варианты заданий.

##### Задача 1.

- 1) Напишите программу, позволяющую работать со стеком. Она должна позволять:
  - поместить элемент наверх стека (оператор POP)
  - удалить верхушку стека (оператор PUSH)
  - обменять значениями два верхних элемента стека (оператор SWAP)
  - определение текущего числа элементов в стеке
  - очистка стека
  - неразрушающее чтение элемента из вершины стека
- 2) Напишите программу, позволяющую работать с очередью. Она должна позволять:
  - Операция `Insert(queue, x)` помещает элемент `x` в конец очереди `queue`.
  - Операция `x = Remove(queue)` удаляет элемент из начала очереди `queue` и присваивает его значение переменной `x`.
  - Операция `Empty(queue)` возвращает значение `TRUE` или `FALSE` в зависимости от того, является ли очередь пустой или нет.
  - определение текущего числа элементов в очереди
  - очистка очереди
- 3) Напишите программу, позволяющую работать с деком. Она должна позволять:
  - очистка дека;
  - включение элемента справа;
  - включение элемента слева;
  - исключение элемента справа;
  - исключение элемента слева;
  - определение размера дека.
- 4) Напишите программу, позволяющую работать со строками. Она должна позволять:
  - определение длины строки;
  - присваивание строк;
  - конкатенация (сцепление) строк;
  - выделение подстроки;

- поиск вхождения.

Задача 2.

Напишите программу, позволяющую работать с линейным списком. Она должна позволять:

- Перебор элементов списка
- Операция формирования списка.
- Вставка элемента в список.
- Удаление элемента.
- Перестановка элементов списка.
- Печать односвязного списка.

## **Тема 2. Структурное программирование. Практическое задание 1 «Исключения в Java»**

Изучить синтаксис операторов.

Скомпилировать примеры программ с различными типами исключений.

Проанализировать ситуации, когда допустимы те или иные исключения.

## **Тема 3. Тестирование и отладка программного обеспечения Лабораторная работа № 2 «Тестирование и отладка»**

- 1) Выполните пошагово задания начиная от создания или импорта проекта.
- 2) \_\_\_ Организуйте логгирование своего проекта, ознакомьтесь с различными уровнями логгирования.
- 3) Выполните отладку проекта с помощью установки точек останова.

## **Тема 4. ООП**

### **Практическое задание 2 «Классы и объекты»**

Ознакомится с теоретической составляющей изучаемых понятий.

Выполнить пошагово задания, описанные в работе

## **Тема 5. Приемы обеспечения технологичности программных продуктов. Методологии разработки.**

### **Практическое задание 3 «Конструкторы и методы»**

Ознакомится с теоретической составляющей изучаемых понятий.

Выполнить пошагово задания, описанные в работе

## **Тема 6. Критерии эффективной архитектуры программных продуктов Лабораторная работа № 3**

Проанализировать код программ, приведённых в заданиях, проследите, какой они должны давать результат.

Запустите программы и сравните результат их работы с вашими ожиданиями.

Измените код, так чтобы он давал достоверный результат.

## **Тема 7. Этапы проектирования ПО Практическое задание 4 «Описание ТЗ»**

Изучить особенности этапов при создании программного продукта

Рассмотреть принципы, по которым формируется техническое задание.

Написать свой пример технического задания по одной из представленных тем или предложить свою.

## **Тема 8. Паттерны проектирования Лабораторная работа №4**

Изучить теоретическую составляющую темы. Порождающие и поведенческие паттерны  
Пошагово создать проекты, продемонстрированные на занятии.  
Создать собственный проект, реализующий порождающий паттерн проектирования  
(Factory method или Abstract factory).

### **Тема 9. Разработка пользовательского интерфейса. Оценка качества программного обеспечения Лабораторная работа № 5**

Требуется:

Выбрать уже существующие приложение, веб-сайт или придумать свой собственный проект, например приложение для аренды или покупки жилья.

Построить пользовательские пути данного проекта

Необходимо представить как блок-схему в нотации UML так и схему с представлением окон(экранов) с указанием и описание всех переходов состояний

#### **Вопросы к экзамену**

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

7.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

БРС включает оценку по следующим показателям:

- По дисциплине, итоговой формой отчетности для которой является зачет, отводится 100 баллов, которые накапливаются студентом в течение всего семестра изучения дисциплины и распределяются по возможности равномерно по всему семестру.

- По дисциплине, итоговой формой отчетности для которой является экзамен, балльная оценка распределяется на две составляющие: семестровую (текущий контроль по учебной дисциплине в течение семестра) – 50 баллов и экзаменационную – 50 баллов.

Суммарный рейтинговый балл освоения учебного курса за семестр на экзамене переводится в 4-балльную оценку, которая считается итоговой оценкой по учебному курсу в текущем семестре и заносится в зачетную книжку студента.

Преподаватель, реализующий дисциплину (модуль), в зависимости от уровня подготовленности обучающихся может использовать иные формы, методы контроля и оценочные средства, исходя из конкретной ситуации.

Балльно-рейтинговая система (БАРС)

Сумма баллов по дисциплине в соответствии с БАРС	Оценка по 4-балльной шкале
90-100	5 (отлично), зачтено
85-89	4 (хорошо), зачтено
75-84	
70-74	
65-69	3 (удовлетворительно), зачтено
60-64	
59 и ниже	2 (неудовлетворительно), не зачтено

Обязательным условием перед сдачей экзамена и/или для получения зачета является выполнение студентом необходимых по рабочей программе для дисциплины видов заданий: выполнение и предоставление отчёта о выполнении лабораторных работ и практических заданий.

Преподаватель, реализующий дисциплину, в зависимости от уровня подготовленности, обучающихся может использовать иные формы, методы контроля и оценочные средства, исходя из конкретной ситуации.

## **8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)**

### **а) Основная литература:**

1. Ландовский В.В., Структуры данных [Электронный ресурс]: учеб. пособие / Ландовский В.В. - Новосибирск : Изд-во НГТУ, 2016. - 68 с. - ISBN 978-5-7782-3080-4 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785778230804.html>.
2. Программирование: типовые задачи, алгоритмы, методы [Электронный ресурс] учебное пособие / Златопольский Д. М. - 3-е изд. (эл.). - М. : БИНОМ, 2015. Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785996329328.html>
3. Программирование [Электронный ресурс]: учеб. пособие / Зайцев М.Г. - Новосибирск : Изд-во НГТУ, 2015. Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785778226265.html>

4.

#### **б) Дополнительная литература:**

1. Программирование: типовые задачи, алгоритмы, методы [Электронный ресурс] учебное пособие / Златопольский Д. М. - 3-е изд. (эл.). - М. : БИНОМ, 2015. Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785996329328.html>
2. Программирование [Электронный ресурс]: учеб. пособие / Зайцев М.Г. - Новосибирск : Изд-во НГТУ, 2015. Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785778226265.html>
3. Программирование на Clojure [Электронный ресурс] монография / Эмерик Ч., Карпер Б., Гранд К. - М. : ДМК Пресс, 2015. Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785970602997.html>
4. Программирование в алгоритмах [Электронный ресурс] / С.М. Окулов. - 5-е изд. (эл.). - М. : БИНОМ, 2014. - (Развитие интеллекта школьников). Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785996323111.html>
5. Программирование [Электронный ресурс] учебное пособие / Н.А. Давыдова, Е.В. Боровская. - М. : БИНОМ, 2015. Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785996326471.html>
6. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Д. Приемы объектно-ориентированного проектирования. Паттерны проектирования: Пер. с англ. - М.: ДМК Пресс., 2011- 368 с.: ил. <http://www.studentlibrary.ru>
7. Батоврин В.К. Системная и программная инженерия. Словарь-справочник: учеб. пособие для вузов. – М.: ДМК Пресс, 2010. - 280 с.: ил. <http://www.studentlibrary.ru>
8. Алгоритмы и структуры данных. Новая версия для Оберона/ Алгоритмы и структуры данных. Новая версия для Оберона + CD / Пер. с англ. Ткачев Ф. В. - М.: ДМК Пресс, 2010. - 272 с.: ил <http://www.studentlibrary.ru>
9. Язык UML. Руководство пользователя. 2-е изд.: Пер. с англ. Мухин Н. - М.: ДМК Пресс. 2006 - 496 с.: ил. <http://www.studentlibrary.ru>

#### **в) Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимый для освоения дисциплины (модуля)**

1. Электронно-библиотечная система (ЭБС) ООО «Политехресурс» «Консультант студента». [www.studentlibrary.ru](http://www.studentlibrary.ru).

### **9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)**

Аудиторные занятия проводятся в общих аудиториях, в том числе в аудиториях, оснащенных мультимедийным оборудованием. Для самостоятельной работы в распоряжении студента имеются читальный зал и компьютерные аудитории, обеспечивающие свободный доступ в Интернет.

При необходимости рабочая программа дисциплины (модуля) может быть адаптирована для обеспечения образовательного процесса инвалидов и лиц с ограниченными возможностями здоровья, в том числе для обучения с применением дистанционных

образовательных технологий. Для этого требуется заявление студента (его законного представителя) и заключение психолого-медико-педагогической комиссии.