



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФГБОУ ВО «ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ

Кафедра Информатики и методики обучения информатике



Директор ПИ ИГУ А.В. Семиров

«09» апреля 2026 г.

Рабочая программа дисциплины (модуля)

Наименование дисциплины (модуля)

Б1.В.02 Алгоритмизация и программирование

Направление подготовки *44.03.04 Профессиональное обучение (по отраслям)*

Направленность (профиль) подготовки *Автоматика и компьютерная инженерия*

Квалификация (степень) выпускника - *Бакалавр*

Форма обучения *очная*

Согласовано с УМС ПИ ИГУ

Протокол №3 от «26» марта 2026г.

Председатель _____ М.С. Павлова

Рекомендовано кафедрой:

Протокол № 7
от «19» марта 2026 г.

Зав. кафедрой _____ Е.Н. Иванова

Иркутск 2026 г.

I. Цели и задачи дисциплины (модуля):

Цель: освоение специальных научных знаний в предметной области алгоритмизации и программирования для последующего их применения в преподавании по программам учебных предметов, курсов, дисциплин (модулей), соответствующих направленности, а также для их применения в учебно-производственном процессе, отвечающем профессиональной деятельности, осваиваемой обучающимися.

Задачи:

- систематизировать методы и приемы решения задач на алгоритмизацию и программирование;
- дать представление о передовых технологиях разработки прикладного программного обеспечения;
- познакомить с современными языками и системами программирования, изучить возможность их применения в различных областях деятельности человека как одного и эффективных инструментов информационного моделирования.

II. Место дисциплины в структуре ОПОП:

- 2.1. Учебная дисциплина «Алгоритмизация и программирование» относится к части, формируемой участниками образовательных отношений блока 1.
- 2.2. Для изучения данной учебной дисциплины необходимы знания и умения, формируемые в процессе изучения курсов «Информатика», «Ознакомительная практика», «Информатики и ИКТ» среднего общего образования.
- 2.3. Перечень последующих учебных дисциплин и практик, для которых необходимы знания и умения, формируемые данной учебной дисциплиной: Методика обучения видам профессиональной деятельности, Педагогическая практика, Практическое (производственное) обучение.

III. Требования к результатам освоения дисциплины (модуля):

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

Компетенция	Индикаторы компетенций	Результаты обучения
<i>ПК-1. Способен осуществлять преподавание по программам учебных предметов, курсов, дисциплин (модулей), соответствующих направленности (профилю)</i>	<i>ИДК ПК-1.1</i> Разрабатывает программно-методическое обеспечение учебных предметов, курсов, дисциплин (модулей) программ профессионального обучения, СПО и (или) ДПП	Знать: – теоретический материал предметной области алгоритмизации и программирования, необходимый для разработки программно-методического обеспечения учебных предметов, курсов, дисциплин программ профессионального обучения; – структуру теоретического материала, относящегося к предметной области алгоритмизации и программирования. Уметь: – отбирать учебный материал для обеспечения учебных предметов, курсов, дисциплин, связанных с алгоритмизацией и программированием; – выбирать инструментальные системы для преподавания по программам учебных предметов, соответствующих направленности

	<p><i>ИДК ПК-1.2</i> Осуществляет педагогическую деятельность, направленную на овладение учебным материалом, его контроль и оценку по общепрофессиональным дисциплинам и профессиональным модулям преподаваемой предметной области</p>	<p><i>Знать:</i> – подходы к изложению учебного материала по алгоритмизации и программированию; – виды деятельности, выполняемые обучающимися в ходе освоения учебного материала по алгоритмизации и программированию.</p> <p><i>Уметь:</i> – ясно и четко излагать материал, относящийся к предметной области алгоритмизации и программирования; – провести оценку правильности выполнения учебных задач в рамках изучения алгоритмизации и программирования.</p>
<p><i>ПК-2. Способен осуществлять учебно-производственный процесс, соответствующий области профессиональной деятельности, осваиваемой обучающимися</i></p>	<p><i>ИДК ПК-2.1</i> демонстрирует владение содержанием, методами и инструментарием преподаваемой предметной области</p>	<p><i>Знать:</i> – компонентные модели современных платформ программирования, их структурную организацию; – технологии конструирования информационной модели явления или процесса, описанного в задаче; – языковые средства реализации построенной модели.</p> <p><i>Уметь:</i> – описывать информационную модель решаемой задачи; – строить информационную модель явления или процесса, описываемого в задаче; – описывать структурную и функциональную организацию предметов и явлений, заданных в условии задачи в соответствии с выбранной технологией декомпозиции задачи и технологией программирования; – конструировать элементы программного средства в соответствии с построенной моделью явления или процесса, описанного в условии задачи.</p>
	<p><i>ИДК-2 ПК-2.2</i> Устанавливает внутрипредметные и межпредметные связи между различными разделами преподаваемой предметной области</p>	<p><i>Знать:</i> – внутренние связи между тематическими блоками и разделами предметной области алгоритмизации и программирования – возможности программирования как одного из способов информационного моделирования.</p> <p><i>Уметь:</i> – применять методы программирования для решения задач, связанных с компьютерным моделированием явлений и процессов других предметных областей.</p>

IV.

5.4. *Методы класса. Структура метода. Виды методов (возвращающие и не возвращающие результат в явном виде). Параметры метода (формальные и фактические). Особенности передачи параметров в метод (по значению, по ссылке). Перегружаемые методы. Управление доступом к методам. Модифицирующие методы (сеттеры) и методы доступа к данным (геттеры).*

5.5. *Конструкторы объектов. Описание конструктора. Конструктор по умолчанию. Перегрузка конструкторов. Ключевое слово `this` (ссылка на текущий экземпляр класса). Создание экземпляра класса (оператор `new`).*

5.6. *Логическая структурная организация класса: интерфейсная часть класса (интерфейс класса) и реализация.*

Раздел 4. Разработка графического пользовательского интерфейса приложения

4.1. *Характеристика библиотек (пакетов), необходимых для разработки графического пользовательского интерфейса.*

4.2. *Основные классы, необходимые для создания стандартных элементов графического пользовательского интерфейса: фрейм (форма), элементы управления (кнопка, метка, однострочное текстовое поле). Менеджеры компоновки. Принцип наследования в разработке графического пользовательского интерфейса.*

4.3. *Событийно-управляемое программирование. Понятие события. Событийная модель Java (или .Net).*

Раздел 5. Структурированные типы данных

5.1. *Массивы в Java (в C#). Объединения*

5.2. *Строки Java (в C#).*

5.3. *Файлы в Java (в C#).*

Раздел 6. Графические возможности языков программирования

4.1. *Дополнительные возможности при разработке графического пользовательского интерфейса (элементы управления альтернативного выбора, контейнеры, диалоги, меню и т.д.).*

4.2. *Характеристика библиотек (пакетов), необходимых для работы с графикой.*

4.3. *Таймер.*

4.3. Разделы и темы дисциплин (модулей) и виды занятий

№ п/п	Наименование раздела/темы	Виды учебной работы, включая самостоятельную работу обучающихся, практическую подготовку (при наличии) и трудоемкость (в часах)				Оценочные средства	Формируемые компетенции (индикаторы)	Всего (в часах)
		Контактная работа преподавателя с обучающимися			СРС (в том числе, внеаудиторная СР, КСР)			
		Лекции	Практ. занятия	Лаб. занятия				
1.	Понятие об объектно-ориентированном программировании. Принципы ООП	1			2	Выполнение заданий в соответствии с инструкциями и методическими указаниями преподавателя	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	3
2.	Характеристика технологии Java (.Net)	1			2	Выполнение заданий в соответствии с инструкциями и методическими указаниями преподавателя	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	3
3.	Элементарные конструкции языка Java (C#)	2	1	4	6	отчет по лабораторной работе отчет по практической работе	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	13
4	Управляющие конструкции языка программирования	2	2	6	16	отчет по лабораторной работе отчет по практической работе	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	26

5	Классы. Структура классов					отчет по лабораторной работе отчет по практической работе	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	24
		2	4	4	14			
6	Методы. Параметры методов.					отчет по лабораторной работе отчет по практической работе	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	31
		3	4	8	16			
7	Конструкторы объектов. Интерфейсная часть класса и реализация					отчет по лабораторной работе отчет по практической работе	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	11
		2	3	4	2			
8	Графический пользовательский интерфейс на Java (C#)					Выполнение заданий в соответствии с инструкциями и методическими указаниями преподавателя	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	11
		1			10			
9	Создание фрейма (формы)					отчет по лабораторной работе отчет по практической работе	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	21
		1	4	6	10			
10	Основные элементы управления					отчет по лабораторной работе отчет по практической работе	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	23
		1	4	6	12			
11	Добавление элементов управления на фрейм					отчет по лабораторной работе отчет по практической работе	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	15
		1	1	3	10			

12	Менеджеры компоновки. Контейнерные элементы управления	2	2	4	6	отчет по лабораторной работе отчет по практической работе	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	14
13	Событийно-управляемое программирование. Событийная модель Java (C#)	2	2	10	16	отчет по лабораторной работе отчет по практической работе	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	30
14	Классы-адаптеры	1	2		4	отчет по практической работе	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	7
15	Анонимные классы	1	2		4	отчет по практической работе	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	7
16	Классы-оболочки	1	1		4	отчет по практической работе	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	6
17	Массивы	4		8	8	отчет по лабораторной работе	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	20

18	Строки					отчет по лабораторной работе	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	21
		4		7	10			
19	Дополнительные элементы управления.					отчет по лабораторной работе	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	18
		2		6	10			
20	Классы, обеспечивающие программирование графики					отчет по лабораторной работе	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i> <i>ИДК -1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	28
		2		14	12			
	Промежуточная аттестация							44
	Консультации							2
	Контроль							18
...	ИТОГО (в часах)							396

4.4. Методические указания по организации самостоятельной работы студентов

1. Выполнение индивидуальных заданий по вариантам. Каждый вариант предполагает решение нескольких задач. Решение каждой задачи оформляется в виде отчета в соответствии со схемой решения задач на ЭВМ. Самостоятельная работа обучающихся поддерживается образовательным порталом ИГУ.

2. Самостоятельный поиск информации, необходимой для выполнения индивидуальных заданий, осуществляется с помощью литературных источников, справочной литературы из фонда библиотеки, а также с помощью сети Интернет.

4.5. Примерная тематика курсовых работ (проектов):

курсовая работа не предусмотрена.

VI. Учебно-методическое и информационное обеспечение дисциплины (модуля):

а) основная литература

1. Баженова, И. Ю. Языки программирования [Электронный ресурс] : учеб. для студ. учрежд. высш. проф. образования, обуч. по направл. "Фундамент. информатика и информ. технологии" и "Информ. безопасность" / И. Ю. Баженова. – ЭВК. – М. : Академия, 2012. – Режим доступа: ЭЧЗ "Библиотех". – 20 доступов.

2. Гниденко, И. Г. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2022. — 235 с. — Текст : электронный // Образовательная платформа Юрайт - Неогранич. доступ.

3. Головин, И. Г. Языки и методы программирования [Электронный ресурс] : учеб. для студ. вузов, обуч. по направл. 010400 "Прикл. математика и информ." и 010300 "Фундамент. информ. и информ. технологии" / И. Г. Головин. – ЭВК. – М. : Академия, 2012. – Режим доступа: ЭЧЗ "Библиотех". – 20 доступов.

4. Зыков, С. В. Программирование. Объектно-ориентированный подход : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2022. — 155 с. — Текст : электронный // Образовательная платформа Юрайт - Неогранич. доступ.

5. Зыков, С. В. Программирование : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2022. — 320 с. — Текст : электронный // Образовательная платформа Юрайт - Неогранич. доступ.

6. Информатика и программирование. Основы информатики [Электронный ресурс] : учеб. для студ. вузов, обуч. по направл. подгот. "Программная инженерия". – ЭВК. – М. : Академия, 2012. – Режим доступа: ЭЧЗ "Библиотех". – 20 доступов.

7. Казанский, А. А. Программирование на Visual C# : учебное пособие для вузов / А. А. Казанский. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2022. — 192 с. — Текст : электронный // Образовательная платформа Юрайт

8. Конова, Е. А. Алгоритмы и программы. Язык C++ [Электронный ресурс] / Е. А. Конова. - Москва : Лань", 2016. - Режим доступа: ЭБС "Издательство Лань". - Неогранич. доступ.

9. Кувшинов, Д. Р. Основы программирования : учебное пособие для вузов / Д. Р. Кувшинов. — Москва : Издательство Юрайт, 2022. — 104 с. — (Высшее образование). — Текст : электронный // Образовательная платформа Юрайт Неогранич. доступ.

10. Кудрина, Е. В. Основы алгоритмизации и программирования на языке C# : учебное пособие для вузов / Е. В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2022. — 322 с. — (Высшее образование). — Текст : электронный // Образовательная платформа Юрайт - Неогранич. доступ.

11. Огнева, М. В. Программирование на языке С++: практический курс : учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина. — Москва : Издательство Юрайт, 2022. — 335 с. — Текст : электронный // Образовательная платформа Юрайт - Неогранич. доступ.

12. Подбельский, В. В. Программирование. Базовый курс С# : учебник для вузов / В. В. Подбельский. — Москва : Издательство Юрайт, 2022. — 369 с. — Текст : электронный // Образовательная платформа Юрайт - Неогранич. доступ.

13. Семакин, И. Г. Основы алгоритмизации программирования [Текст] : учебник / И. Г. Семакин, А. П. Шестаков. — 2-е изд., стер. — М. : Академия, 2011. — 392 с. - 10 экз.

14. Трофимов, В. В. Алгоритмизация и программирование : учебник для вузов / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2022. — 137 с. — Текст : электронный // Образовательная платформа Юрайт - Неогранич. доступ.

15. Тузовский, А. Ф. Объектно-ориентированное программирование: учебное пособие для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2022. — 206 с. — Текст : электронный // Образовательная платформа Юрайт - Неогранич. доступ.

16. Ульянов, В. С. Технологии разработки программного обеспечения [Текст] : учеб. пособие / В. С. Ульянов ; ред. Е. А. Черкашин ; рец. И. С. Абдрахимов ; Иркутский гос. ун-т, Ин-т мат., экон. и информ. — Иркутск : Изд-во ИГУ, 2012. — 108 с. — 56 экз.

17. Черпаков, И. В. Основы программирования : учебник и практикум для вузов / И. В. Черпаков. — Москва : Издательство Юрайт, 2022. — 219 с. — Текст : электронный // Образовательная платформа Юрайт - Неогранич. доступ.

18. Якимов, С. П. Структурное программирование : учебное пособие для вузов / С. П. Якимов. — Москва : Издательство Юрайт, 2022. — 342 с. — Текст : электронный // Образовательная платформа Юрайт - Неогранич. доступ.

б) список авторских методических разработок:

1. Лесников, И.Н. Методические аспекты обучения решению задач на алгоритмизацию и программирование: учеб. пособие / И.Н. Лесников, [и др.]. — Иркутск: Изд-во «Отгиск», 2017. — 80 с.- 25 экз

2. Лесников, И. Н. Применение языка программирования Python в решении некоторых классов типовых задач по информатике в 10-11 классах [Электронный ресурс]: учебное пособие / И.Н. Лесников, А.В. Родионов, Ю.А. Дядькин, — Электрон. текст. дан. (5,48 Мб). — Иркутск: Издательство «Аспринт», 2019. — 170 с. — . - Режим доступа: ЭБС "БиблиоТех". - Неогранич. доступ.

г) базы данных, информационно-справочные и поисковые системы:

1. ЭБС «Юрайт»
2. ЭБС «Библиотех».
3. ЭБС «Издательство «Лань».
4. Национальный цифровой ресурс «РУКОНТ».
5. ЭБС «Айбукс».
6. ИС «Единое окно доступа к образовательным ресурсам».
7. The Java Tutorials – практическое руководство для программистов, использующих язык Java.
8. ИНФОСАЙТ.РУ – библиотека гостей, стандартов и нормативов.

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Помещения и оборудование

Помещения – учебные аудитории для проведения учебных занятий, предусмотренных учебным планом ОПОП ВО бакалавриата, оснащены оборудованием и техническими средствами обучения.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду ФГБОУ ВО «ИГУ».

Оборудование специализированная учебная мебель

Технические средства обучения.

Характеристика материально-технического обеспечения аудиторий ПИ ИГУ, где возможно проведение дисциплины

Аудитория	Учебное оборудование, установленное в аудитории
Поточные аудитории (Учебный корпус №11, 664011, Иркутская область, г. Иркутск, ул. Нижняя Набережная, д. 6)	
304	Проектор SANYO PLC-XM100L 5000 ANSI Im 1024*768 с объективом моторизованным LNS-S20 – 1шт.; экран натяжной DRAPER Luma 2 MW Формат экрана 3:4 267*356 см – 1шт.; доска
305	Мультимедиа проектор Casio XJ-V1; Видеоплеер Panasonic CJ5; Микшерный пульт PHONIC MM1002; Субвуфер активный ELTAX A-10; Системный блок в сборе ProfitPro: (В состав входит: - Процессор Intel Original Core i5 8400 - 1 шт. - Устройство охлаждения(кулер) Deepcool GAMMA ARCHER 3-pin 26dB A1 95 W - 1 шт. - Материнская плата Asrock H310CM- HDV - 1шт. - Корпус Accord ACC-CT308 черный - 1 шт. - Память KingstonDDR4 4Gb 2400MHz - 2шт. - Жесткий диск WD 1Tb WD10EZEX 3.5" - 1шт. - Блок питания Aerocool ATX 400W VX PLUS 400W - 1 шт. - Привод DVD-RW LiteON DVD-RW/+RW iHAS122-14/18/04 - 1шт., Монитор, клавиатура, мышь) - 1 шт.
316	Настенное крепление BEN Q 0.6 Wall Mount; Проектор Ben Q MW 860 USTI; Проектор Ben Q MW 860 USTI Экран Classic Norma 305*406 MW
Учебные и специализированные кабинеты (Учебный корпус №11, 664011, Иркутская область, г. Иркутск, ул. Нижняя Набережная, д. 6)	
246	Компьютер Intel i5-2500 MSI H67MS-E23/DDR3 4096Mb/WD 1TB/DVD-RW/ATX/KW/MOU/ Монитор ViewSonic VX2239Wm-3 – 43 шт.; Системный блок "Снежный барс" + Монитор AOC TFT 23" E2350Sda Black – 7 шт.; Системный блок "Снежный барс"+ Монитор Aser LCD 19" AL-1916 Cs – 1 шт.; Системный блок в комплекте:ASUS H81M-E+ Монитор LG TFT 23" E2350S – 2 шт.; Системный блок в комплекте:ASUS H81M-E+ Монитор Samsung S22C200B – 7 шт.; Проектор ViewSonic PJD8633WS.DLP projector.ultra- Short-Throw Lens 1280*800 - 1 шт.; Экран Screen Media Cololview - 1 шт.; Шкаф настенный металлический - 1шт.; Доска аудиторная ДА 32 белая 3032*1012 - 1 шт.

306	Моноблок Aquarius Mnb Pro T514 R53 - 44 шт.; Интерактивный учебный комплекс SMART Technologies Smart Board 685ix/UX60 - 1 шт.; Коммутатор D-Link DGS-1024 D - 1 шт.; Коммутатор D-Link DGS-1024 C/B1A24 G неуправляемый - 1 шт.; Доска аудиторная ДА-12 белая 1512 x 1012 - 1 шт.
309	Системный блок в сборе + Монитор 23,8 Acer V246HYLBD – 25 шт.; Доска аудиторная ДА-12 белая 1512*1012
312	Системный блок ATN Core is (Монитор LCD 21.5 Viewsonic) – 3 шт.; Персональный компьютер "Система" + Монитор Philips 21,5 226V4LSB – 6 шт.; Системный блок в комплекте: ASUS H81M-E+ Монитор LG TFT 23" E2350S – 1 шт.; доска белая с магнитной поверхностью 120*90-(2002г) – 1шт.

6.2. Лицензионное и программное обеспечение

Windows 10 pro; Adobe acrobat reader DC; Audacity; Firebird; IBExpert; Blender; Codeblocks; GPSS World Student Version 5.2; Lazarus; LibreOffice; DIA; Eclipse IDE for C/C++ Developers; Eclipse IDE for Java Developers; Visual Studio Enterprise; python; IDLE; Far; Firefox; Gimp; Google Chrome; InkScape; Kaspersky AV; MS Office 2007; VisioProfessional; NetBeans; SMART NoteBook; Peazip; Scratch; WinDjView; XnView MP; Компас 3D; Access; GanttProject; AnyLogic; VLC; SMART NoteBook.

VII. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В образовательном процессе используются активные и интерактивные формы проведения занятий (проблемная лекция, тест, мозговой штурм), в том числе дистанционные образовательные технологии, развивающие у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств и формирующие компетенции.

Учитывая, что дисциплина предполагает организацию самостоятельной работы обучающихся, то наряду с указанными видами деятельности, также студентами реализуется поисковая деятельность в направлении обозначенной проблемы (проблемно-ориентированная деятельность) либо без указания направления поиска (поисковая деятельность). В этом случае в рамках дисциплины предполагается использование также информационно-образовательных ресурсов сети Интернет (тексты, видео-лекции ученых и т.д.) и баз данных источников информации вуза как одного из технологических направлений в рамках компьютерных технологий обучения.

Наименование тем занятий с использованием образовательных технологий

№	Тема занятия	Вид занятия	Форма / Методы интерактивного обучения	Кол-во часов
1	Понятие об объектно-ориентированном программировании. Принципы ООП	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	1
2	Характеристика технологии Java (.Net)	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	1
3	Элементарные кон-	Лекция	Презентация материала с применением вспо-	2

	струкции языка Java (C#)		могательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	
		Практика	Интерактивное занятие с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) для организации коллективного решения творческих задач, мозгового штурма и разбора конкретных производственных ситуаций	1
4	Управляющие конструкции языка программирования	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	2
		Практика	Интерактивное занятие с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) для организации коллективного решения творческих задач, мозгового штурма и разбора конкретных производственных ситуаций	2
5	Классы. Структура классов	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	2
		Практика	Интерактивное занятие с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) для организации коллективного решения творческих задач, мозгового штурма и разбора конкретных производственных ситуаций	4
6	Методы. Параметры методов.	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	3
		Практика	Интерактивное занятие с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) для организации коллективного решения творческих задач, мозгового штурма и разбора конкретных производственных ситуаций	4
7	Конструкторы объектов. Интерфейсная часть класса и реализация	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	2
		Практика	Интерактивное занятие с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) для организации	3

			коллективного решения творческих задач, мозгового штурма и разбора конкретных производственных ситуаций	
8	Графический пользовательский интерфейс на Java (C#)	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	1
9	Создание фрейма (формы)	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	1
		Практика	Интерактивное занятие с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) для организации коллективного решения творческих задач, мозгового штурма и разбора конкретных производственных ситуаций	4
10	Основные элементы управления	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	1
		Практика	Интерактивное занятие с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) для организации коллективного решения творческих задач, мозгового штурма и разбора конкретных производственных ситуаций.	4
11	Добавление элементов управления на фрейм	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	1
		Практика	Интерактивное занятие с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) для организации коллективного решения творческих задач, мозгового штурма и разбора конкретных производственных ситуаций	1
12	Менеджеры компоновки. Контейнерные элементы управления	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	2
		Практика	Интерактивное занятие с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) для организации коллективного решения творческих задач, мозгового штурма и разбора конкретных	2

			производственных ситуаций	
13	Событийно-управляемое программирование. Событийная модель Java (C#)	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	2
		Практика	Интерактивное занятие с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) для организации коллективного решения творческих задач, мозгового штурма и разбора конкретных производственных ситуаций	2
14	Классы-адаптеры	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	1
		Практика	Интерактивное занятие с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) для организации коллективного решения творческих задач, мозгового штурма и разбора конкретных производственных ситуаций	2
15	Анонимные классы	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	1
		Практика	Интерактивное занятие с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) для организации коллективного решения творческих задач, мозгового штурма и разбора конкретных производственных ситуаций	2
16	Классы-оболочки	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	1
		Практика	Интерактивное занятие с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) для организации коллективного решения творческих задач, мозгового штурма и разбора конкретных производственных ситуаций	1
17	Массивы	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	4

18	Строки	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	4
19	Дополнительные элементы управления.	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	2
20	Классы, обеспечивающие программирование графики	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	2
Итого часов:				68

VIII. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

8.1. Оценочные средства для проведения текущего контроля успеваемости

- выполнение задания преподавателя в соответствии с инструкцией (аудиторные занятия);
- выполнение самостоятельных частей к теоретическому материалу курса;
- подготовка отчета по индивидуальным заданиям;

Карта оценки компетенций

Шифр компетенции и ее содержание	Показатели (наблюдаемые признаки)	Критерии	Вид оценочного средства
<i>ПК-1. Способен осуществлять преподавание по программам учебных предметов, курсов, дисциплин (модулей), соответствующих направленности (профилю)</i>	Владеет содержанием педагогической деятельности, необходимым для разработки программно-методического обеспечения учебных предметов, курсов, дисциплин (модулей) программ профессионального обучения, СПО и (или) ДПП	Демонстрация владения теоретическим материалом предметной области алгоритмизации и программирования, необходимым для разработки программно-методического обеспечения учебных предметов, курсов, дисциплин программ профессионального обучения	Самостоятельные части к теоретическому материалу курса
		Способность описать структуру теоретического материала, относящегося к предметной области алгоритмизации и программирования	Самостоятельные части к теоретическому материалу курса, индивидуальные задания
		Осуществление выбора инструментальных систем для преподавания по про-	Индивидуальные задания

		граммам учебных предметов, соответствующих направленности	
	Осуществляет педагогическую деятельность, направленную на овладение учебным материалом, его контроль и оценку по общепрофессиональным и профессиональным дисциплинам и профессиональным модулям преподаваемой предметной области	Демонстрация владения видами деятельности, выполняемыми обучающимися в ходе освоения учебного материала по алгоритмизации и программированию.	Индивидуальные задания
		Ясное и четкое изложение материала, относящегося к предметной области алгоритмизации и программирования.	
		Проводит оценку правильности выполнения учебных задач в рамках изучения алгоритмизации и программирования.	
<i>ПК-2. Способен осуществлять учебно-производственный процесс, соответствующий области профессиональной деятельности, осваиваемой обучающимися</i>	Демонстрирует владение содержанием, методами и инструментарием преподаваемой предметной области	Способность выполнять анализ и декомпозицию решаемой задачи (независимо от предметной области) с целью поиска ее решения	Индивидуальное задание (задача)
		Способность преобразовывать содержательную постановку задачи в формальную посредством формальных языков (язык математики)	
		Способность формулировать условия и ограничения на функционирование будущей компьютерной модели (программы) явления или процесса, описанной в задаче	
		Способность осуществлять выбор метода решения задачи	
		Способность описывать алгоритм решения либо объектную модель поставленной задачи	
		Способность создавать проект с помощью программных средств, относящихся к классу инструментального ПО	
		Способность выполнять отладку приложения с помощью соответствующего инструментария разработки ПО.	

Шкала оценки уровня сформированности компетенции

Каждый критерий наблюдаемого признака (показателя) компетенции оценивается по шкале от 0 до 2 баллов:

0 баллов – не выполнен либо выполнен неверно;

1 балл – выполнен частично (имеются неточности);

2 балла – полностью выполнен.

Отчет по индивидуальным заданиям лабораторной работы считается зачтенным, если зачтены все индивидуальные задачи.

Индивидуальная задача считается зачтенной, если сумма баллов, набранных в процессе оценки критериев наблюдаемых признаков для каждой компетенции в отдельности (уровень сформированности компетенции), составил не менее 60% от максимально возможной суммы.

Расчет доли набранных баллов для компетенции от максимально возможной суммы баллов по данной компетенции (уровень сформированности компетенции) осуществляется по формуле:





$$b = \frac{s}{2k} \cdot 100\% ,$$

где b – уровень сформированности компетенции (%), s – суммарный балл по критериям всех наблюдаемых признаков компетенции, k – общее количество критериев по всем наблюдаемым признакам компетенции.

ТЕСТОВЫЕ ЗАДАНИЯ ДЛЯ ФОРМИРОВАНИЯ БАНКА ДЛЯ ПРОВЕДЕНИЯ ТЕСТИРОВАНИЯ

Номер задания	Содержание вопроса	Ответы Правильный ответ выделить жирным шрифтом или цветом	Компетенция (шифр компетенции из УП)
1. Задание закрытого типа на установление соответствия (проверяется автоматически) Инструкция: <i>Прочитайте текст и установите соответствие. К каждой позиции левого столбца выберите позицию в правом столбце</i>			
1.	Задание. Установите соответствие между наименованием операции и ее знаком на языке C++ Наименование операции 1. Отрицание 2. Инкремент 3. Остаток от деления 4. Конъюнкция 5. Декремент 6. Дизъюнкция 7. Сравнение на отношение неравенства 8. Сравнение на отношение равенства Знак на языке C++ a) && b) and c) % d) -- e) mod f) = g) not h) ++ i) or j) !=	1. m 2. h 3. c 4. a 5. d 6. l 7. j 8. k	<i>ПК-1, ПК-2</i>

	k) == l) m) !		
2.	<p>Задание. Установите соответствие между наименованием блока в соответствии с ГОСТ «Схемы алгоритмов, программ, данных и систем» и наименованием команды, обозначаемой этим блоком.</p> <p>Наименование блока</p> <ol style="list-style-type: none"> 1. Данные 2. Процесс 3. Решение <p>Наименование команды</p> <ol style="list-style-type: none"> a) Команда ввода или вывода данных b) Определение истинностного значения логического выражения (вычисление условий) c) Команда присваивания d) Команда многократного выполнения последовательности действий. 	<ol style="list-style-type: none"> 1. a 2. c 3. b 	<i>ПК-1, ПК-2</i>
3.	<p>Задание. Установите соответствие между наименованием системы счисления и префиксом, используемым в C++ для записи целочисленной константы в данной системе счисления.</p> <p>Наименование системы счисления</p> <ol style="list-style-type: none"> 1. Двоичная 2. Восьмеричная 3. Шестнадцатеричная 	<ol style="list-style-type: none"> 1. b 2. c 3. a 	<i>ПК-1, ПК-2</i>

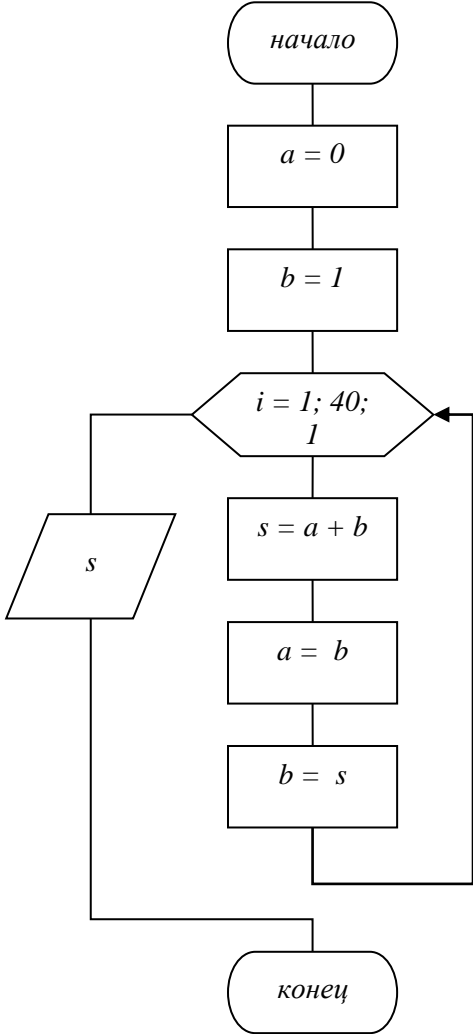
	<p>Префикс</p> <p>a) 0X b) 0B c) 0 d) X0 e) B0</p>		
4.	<p>Задание. Соотнесите изображение блока с его наименованием в ГОСТ «Схемы алгоритмов, программ, данных и систем»</p> <p>Блок</p> <p>1. </p> <p>2. </p> <p>3. </p> <p>4. </p> <p>Наименование блока</p> <p>a) процесс b) терминатор c) подготовка d) решение e) данные</p>	<p>1. b 2. e 3. a 4. d</p>	<p><i>ПК-1, ПК-2</i></p>
5.	<p>Задание. Установите соответствие между зарезервирован-</p>	<p>1. e</p>	<p><i>ПК-1, ПК-2</i></p>

	<p>ным словом языка C++ и его назначением в языке.</p> <p>Зарезервированное слово</p> <ol style="list-style-type: none"> 1. cout 2. cin 3. int 4. if 5. main <p>Назначение зарезервированного слова</p> <ol style="list-style-type: none"> a) оператор условного перехода (условный оператор) b) имя главной функция c) имя стандартного потока ввода d) оператор цикла e) имя стандартного потока вывода f) идентификатор целого типа данных 	<ol style="list-style-type: none"> 2. c 3. f 4. a 5. b 	
6.	<p>Задание. Установите соответствие между оператором C++ и его наименованием</p> <ol style="list-style-type: none"> 1. if 2. for 3. switch 4. while <p>Варианты значений</p> <ol style="list-style-type: none"> a) условный тернарный оператор b) цикл с параметром c) оператор выбора d) условный оператор 	<ol style="list-style-type: none"> 1. d 2. b 3. c 4. e 	ПК-1, ПК-2

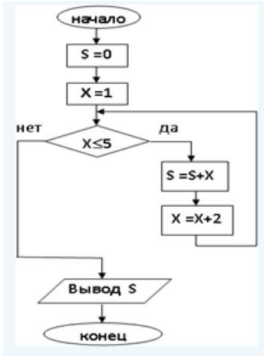
	е) цикл с условием		
2. Задание закрытого типа на установление последовательности (проверяется автоматически) Инструкция: <i>Прочитайте текст и установите последовательность</i>			
7.	Задание. Расположите команды таким образом (установите верную последовательность), чтобы переменные a и b поменялись значениями 1. $x=a$ 2. $a=b$ 3. $b=x$	1 2 3	<i>ПК-1, ПК-2</i>
8.	Задание. Расположите команды таким образом (установите верную последовательность), чтобы переменные a и b поменялись значениями 1. $a=a+b$ 2. $b=a-b$ 3. $a=a-b$	1 2 3	<i>ПК-1, ПК-2</i>
9.	Задание. Установите верную последовательность этапов решения задачи на алгоритмизацию и программирование: 1. Постановка задачи 2. Техническое задание 3. Проектирование текстов 4. Алгоритмизация 5. Построение математической модели	1 5 2 3 4 6 7	<i>ПК-1, ПК-2</i>

	6. Программирование 7. Отладка и тестирование программы		
10.	Задание. Расположите целочисленные константы, записанные на языке C++ в различных системах счисления, по возрастанию: 1. 065 2. 57 3. 0b100100 4. 0x32	4 1 2 3	<i>ПК-1, ПК-2</i>
11.	Задание. Дано натуральное трехзначное число n. Установите верную последовательность заданных операторов, которая меняет местами первую и последнюю цифру этого числа. 1. cout << n; 2. cin >> n; 3. unsigned n, c0, c1, c2; 4. n = 100 * c0 + 10 * c1 + c2; 5. c0 = n % 10; c1 = n / 10 % 10; c2 = n / 100;	3 2 5 4 1	<i>ПК-1, ПК-2</i>
12.	Задание. Установите последовательность логических операций, используемых в языке C++, в порядке убывания приоритета их выполнения 1. && 2. ! 3.	2 1 3	<i>ПК-1, ПК-2</i>
3. Задание открытого типа с развернутым ответом/ задача (проверяет специалист УМУ) Инструкция: <i>Прочитайте текст и запишите развернутый обоснованный ответ</i>			

<p>13.</p>	<p>Задание. Напишите программу на языке C++, вычисляющую площадь правильного многоугольника по заданым количеству сторон этого многоугольника и длине его стороны.</p> <p>В качестве ответа скопируйте созданный программный код в окно ответа. Для поиска площади правильного -n-угольника используйте формулу:</p> $S = \frac{n \cdot a^2}{4 \operatorname{tg} \frac{\pi}{n}}$	<p><u>Примерный вариант программы</u></p> <pre> #include <iostream> #include <cmath> using namespace std; int main() { unsigned n; double a, s; const string ERR1 = "Ошибка: неверно задано количество сторон!\n"; const string ERR2 = "Ошибка: длина стороны неположительна!\n"; setlocale(LC_ALL, "rus"); cout << "Введите количество сторон правильного многоугольника: "; cin >> n; if (n > 2) { cout << "Введите длину стороны: "; cin >> a; if (a > 0) { s = n * a * a / (4 * tan(M_PI / n)); cout << "Площадь многоугольника: " << s << endl; } else cout << ERR2; } else cout << ERR1; return 0; } </pre>	<p>ПК-1, ПК-2</p>
------------	--	--	-------------------

<p>14.</p>	<p>Задание. Используя графический редактор или текстовый процессор составьте блок-схему по следующей инструкции и прикрепите данный файл:</p> <p>1) Определите две переменные, присвоив первой из них значение 0, а второй – значение 1.</p> <p>2) 40 раз выполните последовательность действий:</p> <ul style="list-style-type: none"> – найти сумму переменных, определенных в п.1; – значение второй переменной присвоить первой; – полученную сумму присвоить второй переменной. <p>3) Вывести значение переменной, используемой в п. 2 для хранения суммы.</p>	<p><u>Примерный вариант блок-схемы</u></p>  <pre> graph TD Start([начало]) --> A[a = 0] A --> B[b = 1] B --> Loop{ i = 1; 40; 1 } Loop --> S[s = a + b] S --> A2[a = b] A2 --> B2[b = s] B2 --> Loop Loop --> End([конец]) S --- Out[/s/] </pre>	<p>ПК-1, ПК-2</p>
------------	--	--	-------------------

4. Задания открытого типа с кратким ответом/ вставить термин, словосочетание....., дополнить предложенное (проверяется автоматически, разработчик должен написать правильный вариант ответа) Инструкция: <i>Прочитайте текст и запишите пропущенное слово/словосочетание (термин, словосочетание....., дополнить предложенное)</i>			
15.	<p>Задание. Добавьте оператор вместо знаков ???, чтобы программа корректно рассчитала произведение десяти вводимых целых чисел</p> <pre> int product = 1; int a; for (int n = 1; n <= 10; n++){ cin > a; product *= ???; } </pre>	a	ПК-1, ПК-2
16.	<p>Задание. Задайте начальное значение переменной sum (отмечено знаком ???), чтобы после выполнения приведенного фрагмента программного кода на языке C++ значение этой переменной стало равным 19.</p> <pre> int[matrix[3][3] = { {0, 1, 2}, {3, 4, 5}, {6, 7, 8} }; int sum = ???; for (int i = 0; i < 3 ; i++) for (int j = 2; j < 3 ; j++) sum += matrix[i][j]; </pre>	4	ПК-1, ПК-2
17.	<p>Задание. Дополните определение</p> <p>Исходные ... — это совокупность величин, значениями которых</p>	данные	ПК-1, ПК-2

	можно варьировать в ходе выполнения алгоритма (программы)		
5. Задание с выбором одного правильного ответа из предложенных (проверяется автоматически) Инструкция: Прочитайте текст и выберите один правильный ответ			
18.	<p>Задание. Переменная S после выполнения предложенного алгоритма получит значение...</p>  <p>a) 4 b) 9 c) 15 d) 0</p>	b	<i>ПК-1, ПК-2</i>
19.	<p>Дан фрагмент кода на C++:</p> <pre style="background-color: #e0f0ff; padding: 10px;">const int LENGTH = 6; int vect[LENGTH] = {20, -10, 15, 0, -8, 45}; int sumIndices = 0; for(int i = 0; i < LENGTH; i++) { // пропущенный фрагмент кода sumIndices += i; } cout << sumIndices;</pre>	d) if (vect[i] > 0)	<i>ПК-1, ПК-2</i>

	<p>Выберите необходимый оператор на место пропущенного фрагмента кода, чтобы можно было получить сумму индексов положительных элементов массива vect, как результат выполнения данной последовательности операторов</p> <p>a) if (i >= 0) b) if (vect[i] >= 0) c) if (i > 0) d) if (vect[i] >0)</p>		
<p>6. Задание с выбором нескольких правильных ответов из предложенных (проверяется автоматически) Инструкция: <i>Прочитайте текст и выберите все правильные ответы</i></p>			
20.	<p>Задание. В качестве имени переменной в C++ можно использовать следующие конструкции:</p> <p>a) <code>_myvalue_</code> b) <code>myValue16</code> c) <code>my-value</code> d) <code>16my_value</code> e) <code>myValue</code></p>	<p>a</p> <p>b</p> <p>e</p>	<i>ПК-1, ПК-2</i>
21.	<p>Задание. Пусть переменным x и y заданы некоторые значения. Выберите одно или несколько выражений, которое позволит на языке C++ получить в переменной x произведение текущих значений переменных x и y</p> <p>a) <code>x*=y</code> b) <code>x+=y*x</code> c) <code>x*=x*y</code> d) <code>x=x*y</code></p>	<p>a</p> <p>d</p>	<i>ПК-1, ПК-2</i>

Форма отчета по индивидуальному заданию (задаче)

Задача № 1. Текст задачи

1. Постановка задачи
2. Математическая модель

Исходные данные: ...

Выходные данные: ...

Связь: ...

3. Техническое задание

3.1. Таблица внешней спецификации

Вид данных	Наименование величины (переменной)	Тип	ОДЗ	Назначение	Единицы измерения

3.2. Формулировка условий и ограничений

Условия на исходные данные	Действие алгоритма (программы)
1.	
2.	
...	

4. Проектирование тестов

№ теста	Исходные данные	Выходные данные	Примечание

5. Алгоритмизация / Объектная модель задачи
алгоритм решения задачи либо построение объектной модели задачи

6. Программирование
программный код

Демонстрационный вариант индивидуального задания

Задача.

Создать приложение, моделирующее управление банковским счетом. При создании счета должна быть реализована возможность зачисления определенной суммы денежных средств. Приложение поддерживает возможность внесения дополнительных средств или их списание, а также получение информации о текущем балансе. Если после снятия средств баланс стал отрицательным, то выполняется начисление штрафа в виде списания дополнительных 5 рублей. Реализовать возможность получения информации о сумме штрафов.

1. **Постановка задачи:** допустим, что при открытии счета задается лишь начальное значение баланса (номер счета не задается), данные о клиенте также не регистрируются; допускается снятие денежных средств со счета в количестве, превосходящем баланс.

2. Математическая модель.

Исходные данные: $initialBalance \in R^+ \cup \{0\}$, $n \in N$, $amount_i \in R^+$, $operation_i \in \{1, 2\}$, $i = 1, 2, \dots, n$

Выходные данные: $balance_i \in R$, $fee_i \in R^+ \cup \{0\}$.

Связь: $balance_0 = initialBalance$,

$$balance_i = \begin{cases} balance_{i-1} + amount_i, & \text{если } operation_i = 1; \\ balance_{i-1} - amount_i, & \text{если } (operation_i = 2) \wedge (balance_{i-1} - amount_i) \geq 0; \\ balance_{i-1} - amount_i - 5, & \text{если } (operation_i = 2) \wedge (balance_{i-1} - amount_i) < 0 \end{cases}$$

$$fee_i = \begin{cases} fee_{i-1} + 5_i, & \text{если } (operation_i = 2) \wedge (balance_{i-1} - amount_i < 0); \\ fee_{i-1}, & \text{если } (operation_i = 2) \wedge (balance_{i-1} - amount_i \geq 0), \end{cases}$$

где $i = 1, 2, \dots, n$,

n – общее количество выполненных операций с банковским счетом,

$initialBalance$ – начальное значение баланса, задаваемое при открытии счета,

$balance_i$ – текущее значение баланса на момент выполнения i -ой операции,

$amount_i$ – денежная сумма, зачисляемая или списываемая со счета,

$operation_i$ – вид текущей операции (1 – зачисление денежных средств на счет, 2 – списание денежных средств со счета),

fee_i – сумма штрафов после выполнения i -ой операции.

3. Техническое задание

3.1. Формулировка условий и ограничений

Таблица 1

Условия работы приложения

Условия на исходные данные	Действие приложения (алгоритма, программы)
1. $initialBalance \in R^+ \cup \{0\}$	Открывается счет с заданным начальным балансом, выводится сообщение «Счет открыт!»
2. $initialBalance \in R \setminus R^+$	Счет не открывается, выводится сообщение «Денежная сумма не может быть отрицательной! », ожидается ввод корректного значения начального баланса
3. $\forall i = \overline{1, n}, (amount_i \in R^+) \wedge (operation_i \in \{1, 2\})$	Выполняется требуемая операция и вычисление соответствующих текущей i -ой операции значений $balance_i$ и fee_i
4. $\exists i = \overline{1, n}, (amount_i \in R \setminus R^+) \wedge (operation_i \in \{1, 2\})$	Операция не выполняется, выводится сообщение об ошибке «Сумма не может быть отрицательной!»
5. $(initialBalance \notin R) \vee (\exists i = \overline{1, n}, (amount_i \notin R) \wedge (operation_i \in \{1, 2\}))$	ДСТ (либо результат неопределенный)

Примечание:

– количество проводимых операций заранее не определяется;

– выбор операции ($operation_i$) ограничен двумя вариантами, ввод других вариантов невозможен.

3.2. Описание интерфейса приложения

Ввод-вывод данных осуществляется посредством совокупности форм, макеты которых представлены на рисунках 1-3.

1. Форма «Новый счет» (рис. 1) открывается при запуске приложения.

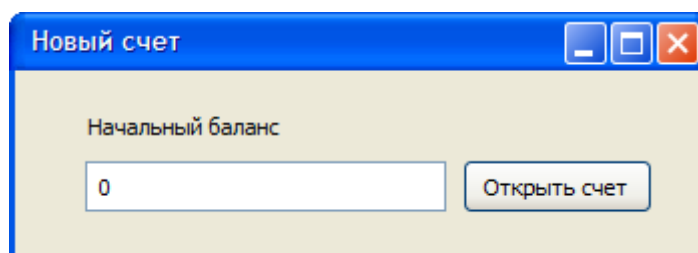


Рисунок 1 – Форма «Новый счет»

Описание входной информации представлено в таблице 2.

Таблица 2

Описание входных данных для формы «Новый счет»

Наименование величины (данных)	Тип данных	Формат представления	Ограничения	Примечание
Начальный баланс	Числовой	0,00	Положительное значение	Задается в рублях

Описание выходной информации: выходной информацией является создание банковского счета с заданной денежной суммой на балансе.

2. Форма «Управление счетом» дает возможность выполнять операции по управлению счетом, а также представляет сведения о текущем состоянии баланса и сумме штрафов. Данная форма открывается при нажатии на кнопку «Открыть счет» формы «Новый счет» (рис. 1)

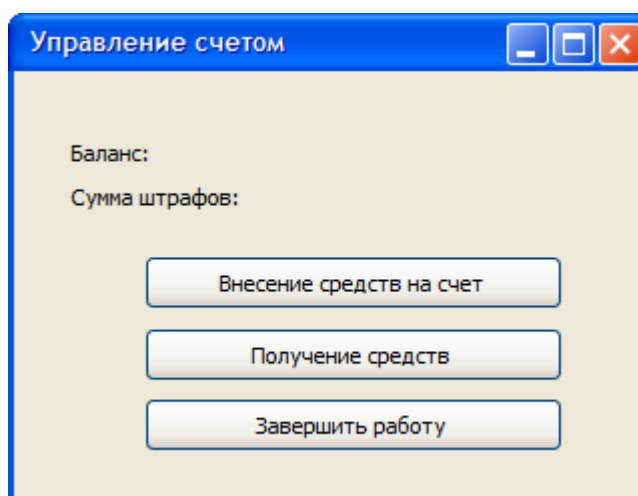


Рисунок 2 – Форма «Управление счетом»

3. Форма «Сумма денежных средств» (рис. 3) предназначена для ввода денежной суммы для внесения или списание со счета. Данная форма открывается при нажатии на кнопку «Получение средств» или «Внесение средств на счет» формы «Управление счетом» (рис. 2).

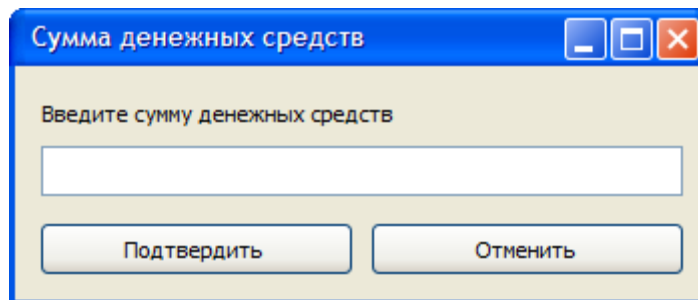


Рисунок 3 – Форма «Сумма денежных средств»

Описание входной информации представлено в таблице 3.

Таблица 3

Описание входных данных для формы «Сумма денежных средств»

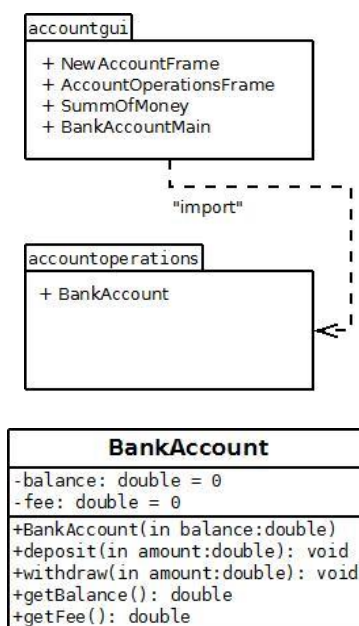
Наименование величины (данных)	Тип данных	Формат представления	Ограничения	Примечание
Сумма денежных средств	Числовой	0,00	Положительное значение	Задается в рублях

Описание выходной информации: выходной информацией является внесение изменений в баланс банковского счета и сумму штрафов, а также обновление сведений о балансе и штрафах представленных на форме «Управление счетом».

4. Проектирование тестов

№	Исходные данные			Выходные данные		Примечание
	<i>initialBalance</i>	<i>amount_i</i>	<i>operation_i</i>	<i>balance_i</i>	<i>fee_i</i>	
1	0	(20; 35; 60)	(1; 2; 1)	(20; -15; 40)	(0; 5; 5)	Выполнено три операции (n = 3)
2	...					
3	...					

5. Алгоритмизация/Объектная модель задачи



6. Программирование

Листинг класса BankAccount

```

package accountoperations;

public class BankAccount {
    // поля класса
    private double balance = 0.0;
    private double fee = 0.0;

    //конструктор
    public BankAccount(double balance){
        this.balance = balance;
    }

    // методы

    public void deposit(double amount){
        if (amount > 0)
            this.balance += amount;
    }

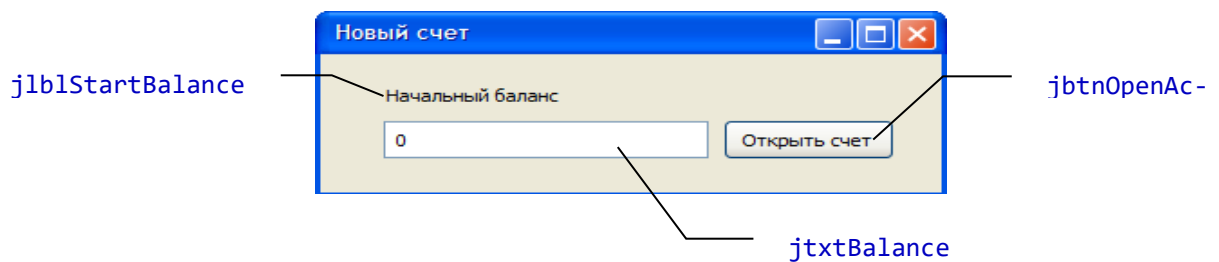
    public void withdraw(double amount){
        if (this.balance - amount >= 0)
            balance -= amount;
        else{
            this.balance -= (5.0 + amount);
            this.fee +=5;
        }
    }

    public double getBalance(){
        return this.balance;
    }

    public double getFees(){
        return this.fee;
    }
}

```

Листинг класса NewAccountFrame



Класс	Имя объекта	Интерфейс слушателя	Метод-обработчик события	Примечание
JLabel	jlblStartBalance			Метка с надписью «Начальный баланс»
JTextField	jtxtBalance			Однострочное текстовое поля для ввода суммы баланса
JButton	jbtnOpenAccount	ActionListener	ActionPerformed(ActionEvent e)	Обработка щелчка мышью на кнопку с надписью «Открыть счет»

```

package accountgui;

```

```

import javax.swing.*;
import accountoperations.*;
import java.awt.*;
import java.awt.event.*;

public class NewAccountFrame extends JFrame{
    // элементы управления
    private JLabel jlblStartBalance;
    private JTextField jtxtBalance;
    private JButton jbtnOpenAccount;
    private BankAccount account;

    //конструктор
    public NewAccountFrame(){
        initComponents();
    }

    public NewAccountFrame(BankAccount account){
        this.account = account;
        initComponents();
    }

    // метод инициализации

    private void initComponents() {
        this.setDefaultCloseOperation(EXIT_ON_CLOSE);

        //создаем элементы в хипе
        jlblStartBalance = new JLabel("Начальный баланс");
        jtxtBalance = new JTextField(10);
        jbtnOpenAccount = new JButton("Открыть счет");

        // менеджер компоновки
        this.setLayout(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        // для метки
        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.insets = new Insets(2,2,2,2);
        this.add(jlblStartBalance , gbc);

        //для текстового поля
        gbc.gridx = 0;
        gbc.gridy = 1;
        gbc.anchor = GridBagConstraints.BASELINE;
        this.add(jtxtBalance , gbc);

        //для кнопки
        gbc.gridx = 1;
        gbc.gridy = 1;
        gbc.anchor = GridBagConstraints.BASELINE;
        this.add(jbtnOpenAccount , gbc);

        this.setSize(400,100);
        setTitle("Новый счет");

        jbtnOpenAccount.addActionListener(new ActionListener(){

            public void actionPerformed(ActionEvent arg0) {

```


				мышью на кнопку с надписью «Завершить работу», завершение работы приложения
NewAccountFrame	<code>this</code>	WindowListener	<code>windowActivated(WindowEvent e)</code>	Обработка события, связанного с активацией окна, обновление данных в метках

```

package accountgui;

import javax.swing.*;
import accountoperations.*;
import java.awt.*;
import java.awt.event.*;

public class AccountOperationsFrame extends JFrame{
    //элементы управления
    private JLabel jlblBalance, jlblFees;
    private JButton jbtnDeposit, jbtnWithdraw, jbtnExit;
    private BankAccount account;

    //конструктор

    public AccountOperationsFrame(BankAccount account){
        this.account = account;
        initComponents();
    }
    public AccountOperationsFrame(){
        initComponents();
    }

    //метод инициализации

    public void initComponents(){

        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setTitle("Управление счетом");
        setSize(400,200);
        //создаем элементы в хипе
        jlblBalance = new JLabel("Баланс: ");
        jlblFees = new JLabel("Сумма штрафов: ");
        jbtnDeposit = new JButton("Внесение средств на счет");
        jbtnWithdraw = new JButton("Получение средств");
        jbtnWithdraw.setPreferredSize(jbtnDeposit.getPreferredSize());
        jbtnExit = new JButton("Завершить работу");
        jbtnExit.setPreferredSize(jbtnDeposit.getPreferredSize());

        //менеджер компоновки

        this.setLayout(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        // для метки "Баланс"
        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.anchor = GridBagConstraints.BASELINE_LEADING;
        gbc.fill = GridBagConstraints.HORIZONTAL;
        gbc.weightx = 100;
        gbc.insets = new Insets(2,2,2,2);
        this.add(jlblBalance , gbc);

        //для метки "Сумма штрафов"

```

```

gbc.gridx = 0;
gbc.gridy = 1;
gbc.anchor = GridBagConstraints.BASILINE_LEADING;
this.add(jlblFees , gbc);

//для кнопки "Внесение средств на счет"
gbc.gridx = 0;
gbc.gridy = 2;
gbc.anchor = GridBagConstraints.BASILINE;
gbc.fill = GridBagConstraints.NONE;
gbc.weightx = 0;
this.add(jbtnDeposit , gbc);

//для кнопки "Получение средств"
gbc.gridx = 0;
gbc.gridy = 3;
this.add(jbtnWithdraw , gbc);

//для кнопки "Завершить работу"
gbc.gridx = 0;
gbc.gridy = 4;
this.add(jbtnExit, gbc);

this.addWindowListener(new WindowAdapter(){

    public void windowActivated(WindowEvent arg0) {
        lblBalance.setText("Баланс: "+account.getBalance() + "
рублей");
        lblFees.setText("Сумма штрафов: "+ account.getFees() + "
рублей");
    }

});

jbtnDeposit.addActionListener(new ActionListener(){

    public void actionPerformed(ActionEvent arg0) {
        (new SummOfMoney(account, 1)).setVisible(true);
    }

});

jbtnWithdraw.addActionListener(new ActionListener(){

    public void actionPerformed(ActionEvent arg0) {
        (new SummOfMoney(account, 2)).setVisible(true);
    }

});

jbtnExit.addActionListener(new ActionListener(){

    public void actionPerformed(ActionEvent e) {
        //System.exit(0);
        Runtime.getRuntime().exit(0);
    }

});

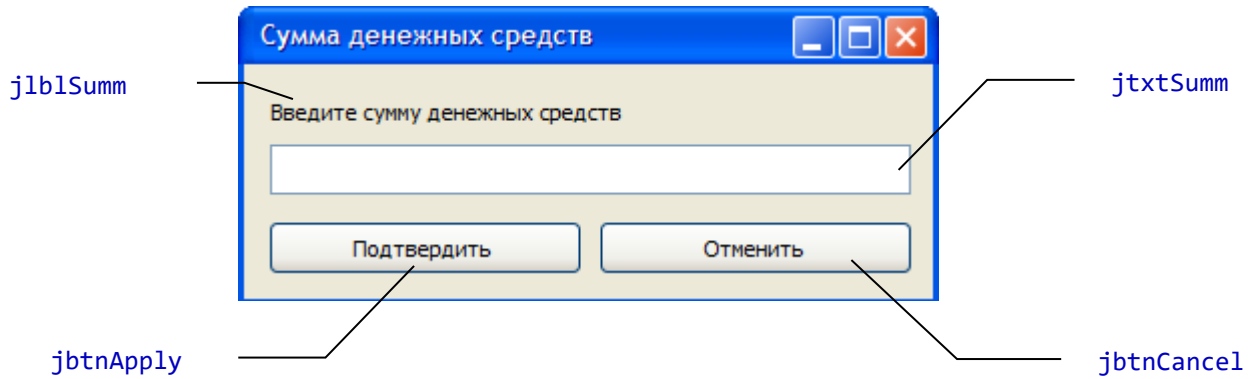
```

```

    }
    });
}
}

```

Листинг класса SummOfMoney



Класс	Имя объекта	Интерфейс слушателя	Метод-обработчик события	Примечание
JLabel	jlblSumm			Метка с надписью «Введите сумму денежных средств»
JTextField	jtxtSumm			Однострочное текстовое поля для ввода суммы
JButton	jbtnApply	ActionListener	ActionPerformed(ActionEvent e)	Обработка щелчка мышью на кнопку с надписью «Подтвердить», выполнение операции
	jbtnCancel	ActionListener	ActionPerformed(ActionEvent e)	Обработка щелчка мышью на кнопку с надписью «Отменить», закрывает окно

```

package accountgui;

import javax.swing.*;
import accountoperations.*;
import java.awt.*;
import java.awt.event.*;

public class SummOfMoney extends JFrame {
    private JLabel jlblSumm;
    private JTextField jtxtSumm;
    private JButton jbtnApply;
    private JButton jbtnCancel;
    private BankAccount account;
    private int kindOfOperation;

    public SummOfMoney(){
        initComponents();
    }
    public SummOfMoney(BankAccount account, int kindOfOperation){
        this.account = account;
        this.kindOfOperation = kindOfOperation;
        initComponents();
    }
}

```

```

private void initComponents(){
    this.setDefaultCloseOperation(this.DISPOSE_ON_CLOSE);

    jlblSumm = new JLabel("Введите сумму денежных средств");
    jtxtSumm = new JTextField(10);
    jbbtnApply = new JButton("Подтвердить");
    jbbtnCancel = new JButton("Отменить");
    jbbtnCancel.setPreferredSize(jbbtnApply.getPreferredSize());

    // менеджер компоновки
    this.setLayout(new GridBagLayout());
    GridBagConstraints gbc = new GridBagConstraints();

    // для метки
    gbc.gridx = 0;
    gbc.gridy = 0;
    gbc.gridwidth = 2;
    gbc.insets = new Insets(2,2,2,2);
    this.add(jlblSumm , gbc);

    //для текстового поля
    gbc.gridx = 0;
    gbc.gridy = 1;
    gbc.gridwidth = 2;
    gbc.fill = GridBagConstraints.HORIZONTAL;
    gbc.anchor = GridBagConstraints.BASELINE;
    this.add(jtxtSumm , gbc);

    //для кнопки
    gbc.gridx = 0;
    gbc.gridy = 2;
    gbc.gridwidth = 1;
    gbc.anchor = GridBagConstraints.BASELINE;
    this.add(jbbtnApply, gbc);

    //для кнопки
    gbc.gridx = 1;
    gbc.gridy = 2;

    gbc.anchor = GridBagConstraints.BASELINE;
    this.add(jbbtnCancel, gbc);
    this.setTitle("Сумма денежных средств");
    this.setResizable(false);
    pack();

    jbbtnApply.addActionListener(new ActionListener(){

        public void actionPerformed(ActionEvent e) {
            double a = Double.parseDouble(jtxtSumm.getText());
            if (a > 0){
                if (kindOfOperation == 1)
                    account.deposit(a);
                else
                    account.withdraw(a);
                JOptionPane.showMessageDialog(SummOfMoney.this, "Опера-
ция выполнена!");
            }
            SummOfMoney.this.dispose();
        }
    });
}

```


7. Преобразование типов величин в Java (либо C#). Операция приведения типов данных. Явное указание типа числовой литералы с помощью суффикса.
8. Линейные алгоритмы (понятие, базовые алгоритмические структуры, применяемые для записи линейных алгоритмов). Конструкции языка Java (либо C#), реализующие команды линейных алгоритмов (оператор присваивания, возможности языка Java (либо C#) для реализации ввода-вывода данных в режиме консоли).
9. Форматированный вывод данных на языке Java (либо C#) в режиме консоли.
10. Разветвляющиеся алгоритмы (понятие, базовые алгоритмические структуры, применяемые для записи алгоритма). Оператор условного перехода. Цепочки if-else-if. Условный тернарный оператор.
11. Структура выбора. Оператор выбора на языке Java (либо C#).
12. Циклические алгоритмы (понятие, базовые алгоритмические структуры, применяемые для записи алгоритма). Виды циклов. Операторы языка Java, используемые для записи циклических алгоритмов.
13. Классы в Java (либо C#): понятие класса, описание класса, члены класса, модификаторы доступа к членам класса, статические члены класса, интерфейсная часть класса и реализация.
14. Методы, виды методов. Параметры методов. Особенности передачи параметров в метод на языке Java (либо C#). Управление доступом к методам класса. Статические методы.
15. Конструкторы объектов. Вызов конструктора.
16. Создание графического пользовательского интерфейс средствами Java (либо C#). Контейнеры, стандартные элементы управления (кнопка, однострочное текстовое поле, метка). Структура фрейма (формы). Менеджеры компоновки.
17. Принцип наследования в ООП. Применение наследования при разработке графического пользовательского интерфейса.
18. Событийно-управляемое программирование. Событийная модель Java (либо .Net). Регистрация источника события в слушателе.
19. Классы-адаптеры. Анонимные классы.
20. Классы-обертки (Классы-оболочки).
21. Массивы в Java (C#): базовые понятия, виды массивов, описание и создание массивов.
22. Сортировка массивов. Методы сортировки.
23. Классы Arrays и ArrayList в работе с массивами (назначение, возможности и методы классов).
24. Строки. Классы, обеспечивающие работу со строками. Методы обработки строк.
25. Графические возможности Java (.Net).

Критерии выставления оценки за экзамен

Оценка	Критерии
Отлично	<ul style="list-style-type: none"> – дан верный и полный ответ на теоретический вопрос из списка вопросов к экзамену; – зачтено выполнение всех индивидуальных заданий; – обучающийся проявил владение каждой из компетенций, развиваемых в рамках дисциплины, на уровне не менее 60% в про-

	<p>цессе решения полученной на экзамене типовой задачи.</p>
Хорошо	<ul style="list-style-type: none"> – дан верный, но неполный ответ на теоретический вопрос из списка вопросов к экзамену; – зачтено выполнение всех индивидуальных заданий; – обучающийся проявил владение каждой из компетенций, развиваемых в рамках дисциплины, на уровне не менее 60% в процессе решения полученной на экзамене типовой задачи. <p>либо</p> <ul style="list-style-type: none"> – дан верный и полный ответ на теоретический вопрос из списка вопросов к экзамену – зачтено выполнение от 80 до 95 % всех индивидуальных заданий; – обучающийся проявил владение каждой из компетенций, развиваемых в рамках дисциплины, на уровне не менее 60% в процессе решения полученной на экзамене типовой задачи.
Удовлетворительно	<ul style="list-style-type: none"> – дан ответ на теоретический вопрос из списка вопросов к экзамену, во время ответа имелись недочеты, ошибки в определениях, классификациях, выводах (в целом представление сформировано) – зачтено выполнение не менее 60% всех индивидуальных заданий; – обучающийся проявил владение каждой из компетенций, развиваемых в рамках дисциплины, на уровне не менее 60% в процессе решения полученной на экзамене типовой задачи.
Неудовлетворительно	<ul style="list-style-type: none"> – дан неверный ответ на теоретический вопрос из списка вопросов к экзамену, во время ответа имелись существенные недостатки в определениях, классификациях, выводах; – обучающийся не проявил владение хотя бы одной из компетенций, развиваемых в рамках дисциплины, на уровне не менее 60% в процессе решения полученной на экзамене типовой задачи.

Документ составлен в соответствии с требованиями ФГОС по направлению 44.03.04 «Профессиональное обучение (по отраслям)», утвержденного приказом Минобрнауки РФ №124 от 22 февраля 2018 г.

Настоящая программа не может быть воспроизведена ни в какой форме без предварительного письменного разрешения кафедры-разработчика программы.