



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**  
ФГБОУ ВО «ИГУ»

**ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ**

**Кафедра Информатики и методики обучения информатике**



УТВЕРЖДАЮ

Директор ПИ ИГУ А.В. Семиров

«17» июня 2021 г.

**Рабочая программа дисциплины (модуля)**

Наименование дисциплины (модуля)

*Б1.В.04 Алгоритмизация и программирование*

Направление подготовки *44.03.05 Педагогическое образование (с двумя профилями подготовки)*

Направленность (профиль) подготовки *Информатика-Физика*

Квалификация (степень) выпускника - *Бакалавр*

Форма обучения *очная*

**Согласовано с УМС ПИ ИГУ**

Протокол №10 от «15» июня 2021г.

Председатель \_\_\_\_\_ М.С. Павлова

**Рекомендовано кафедрой:**

Протокол № 14  
от «07» июня 2021 г.

Зав. кафедрой \_\_\_\_\_ Е.Н. Иванова

Иркутск 2021 г.

## I. Цели и задачи дисциплины (модуля):

**Цель:** освоение теоретических знаний и овладение практической деятельностью, относящихся к области алгоритмизации и программирования, для их последующего применения в преподавании по программам учебных дисциплин, соответствующих направленности.

### 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.
- 2.2. Для изучения данной учебной дисциплины необходимы знания и умения, формируемые в процессе изучения курса информатики и ИКТ среднего общего образования, а также в ходе изучения дисциплины «Профессиональная ИКТ-компетентность педагога».
- 2.3. Перечень последующих учебных дисциплин и практик, для которых необходимы знания и умения, формируемые данной учебной дисциплиной: объектно-ориентированные технологии, язык программирования Python, объектно-ориентированное программирование на Python, педагогическая практика.

## 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 выполнять педагогическую деятельность по проектированию и реализации образовательного процесса в образовательных организациях основного общего, среднего общего образования</i>	<i>ИДК-1 ПК-1.1</i> Разрабатывает учебно-методическое обеспечение основных общеобразовательных программ дисциплин предметной области знаний для реализации образовательного процесса в образовательных организациях основного общего, среднего общего образования.	<i>Знать:</i> – теоретический материал предметной области алгоритмизации и программирования, необходимый для разработки учебно-методического обеспечения дисциплин программ основного общего, среднего общего образования; – структуру теоретического материала, относящегося к предметной области алгоритмизации и программирования.  <i>Уметь:</i> – отбирать учебный материал для обеспечения учебных дисциплин, связанных с алгоритмизацией и программированием; – выбирать инструментальные системы для преподавания по программам учебных предметов, соответствующих направленности
	<i>ИДК-2 ПК-1.2</i>	<i>Знать:</i>

	<p>Осуществляет урочную и внеурочную деятельность по дисциплинам предметной области знаний</p>	<p>– подходы к изложению учебного материала по алгоритмизации и программированию;</p> <p>– виды деятельности, выполняемые обучающимися в ходе освоения учебного материала по алгоритмизации и программированию.</p> <p><i>Уметь:</i></p> <p>– ясно и четко излагать материал, относящийся к предметной области алгоритмизации и программирования;</p> <p>– провести оценку правильности выполнения учебных задач в рамках изучения алгоритмизации и программирования.</p>
<p><i>ПК-2. Способен к применению теоретических знаний и практических умений в преподаваемой предметной области</i></p>	<p><i>ИДК-1 ПК-2.1</i></p> <p>демонстрирует владение содержанием, методами и инструментарием преподаваемой предметной области</p>	<p><i>Знать:</i></p> <p>– компонентные модели современных платформ программирования, их структурную организацию;</p> <p>– технологии конструирования информационной модели явления или процесса, описанного в задаче;</p> <p>– языковые средства реализации построенной модели.</p> <p><i>Уметь:</i></p> <p>– описывать информационную модель решаемой задачи;</p> <p>– строить информационную модель явления или процесса, описываемого в задаче;</p> <p>– описывать структурную и функциональную организацию предметов и явлений, заданных в условии задачи в соответствии с выбранной технологией декомпозиции задачи и технологией программирования;</p> <p>– 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></p> <p>Устанавливает внутрипредметные и межпредметные связи между различными разделами преподаваемой предметной области</p>	<p><i>Знать:</i></p> <p>– внутренние связи между тематическими блоками и разделами предметной области алгоритмизации и программирования</p> <p>– возможности программирования как одного из способов информационного моделирования.</p> <p><i>Уметь:</i></p> <p>– применять методы программирования для решения задач, связанных с компьютерным моделированием явлений и</p>

		процессов других предметных областей.
--	--	---------------------------------------

#### IV. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

##### 4.1. Объем дисциплины (модуля) и виды учебной работы (разделяется по формам обучения)

Вид учебной работы	Всего часов/ зачетных единиц	Семестры			
		2	3		
<b>Аудиторные занятия (всего)</b>	124	60	64		
В том числе:					
Лекции (Лек) / (Электр)	52	20	32		
Практические занятия (Пр) / (Электр)					
Лабораторные работы (Лаб)	72	40	32		
<b>Консультации (Конс)</b>	<b>3</b>	<b>1</b>	<b>2</b>		
<b>Самостоятельная работа (СР)</b>	63	39	24		
Вид промежуточной аттестации (зачет, экзамен), часы (Контроль)	44		44		
Контроль (КО)	18	8	10		
<b>Контактная работа, всего (Конт. раб)</b>	145	62	83		
Общая трудоемкость:                   зачетные единицы  часы	7	3	4		
	252	108	144		

##### 4.2. Содержание учебного материала дисциплины (модуля)

###### Раздел 1. Алгоритмическая и объектно-ориентированная декомпозиции задачи

1.1. Суть алгоритмической и объектно-ориентированной декомпозиций задачи, их отличие. Базовые понятия, связанные с объектно-ориентированной технологией решения задачи: объект, поведение и состояние объекта, свойства, методы, класс.

1.2. Объектно-ориентированное программирование. Принципы объектно-ориентированного программирования (инкапсуляция, наследование, полиморфизм).

###### Раздел 2. Элементарные и управляющие конструкции языков программирования

2.1. Представление о технологиях Java и .Net. Понятие сред исполнения. Среды исполнения в технологии Java (виртуальная машина) и в технологии .Net (общезыковая среда исполнения). Языки программирования, поддерживаемые технологией Java и .Net. Характеристика сред программирования на Java (C# в .Net)

2.2. Элементарные конструкции языка программирования (Java, C#). Алфавит языка программирования. Требования к идентификаторам величин. Лексемы. Операции. Структура исходного кода приложения. Комментарии к программному коду. Комментарии к программному коду. Типы данных (простые и ссылочные). Характеристика простых типов данных. Описание переменных и констант. Понятие пакета в Java (про-

странства имен в C#). Математические функции (методы класса Math). Преобразование типов данных. Операция приведения типов. Применение суффиксов к литералам.

2.3. Исполняемые операторы языка программирования. Способы ввода-вывода данных (консоль): потоки ввода-вывода, классы и методы для работы с ними. Форматированный вывод. Управляющие конструкции: программирование развилки, структуры «выбор», циклов.

### **Раздел 3. Введение в объектно-ориентированное программирование**

4.3. Класс. Описание класса. Члены класса: поля, методы. Описание полей класса. Управление доступом к полям класса.

4.4. 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).

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

4.6. 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)
		Лекц.	Лаб. занятия	СРС			
1	Понятие об объектно-ориентированном программировании. Принципы ООП	1		1	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>	2
2	Характеристика технологии Java (.Net)	1			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>	1
3	Элементарные конструкции языка Java (C#)	2	4	2	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>	8
4	Управляющие конструкции языка программирования	2	6	6	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>	14
5	Классы. Структура классов	2	4	4	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>	10
6	Методы. Параметры методов.	4	8	6	Отчет по лабораторной работе	<i>ИДК-1 ПК-1.1</i> <i>ИДК-2 ПК-1.2</i>	18

						<i>ИДК-1 ПК-2.1</i> <i>ИДК-2 ПК-2.2</i>	
7	Конструкторы объектов. Интерфейсная часть класса и реализация	2	4	2	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>	8
8	Графический пользовательский интерфейс на Java (C#)	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
9	Создание фрейма (формы)	4	6	2	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>	12
10	Основные элементы управления	4	6	4	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>	14
11	Добавление элементов управления на фрейм	2	3	6	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>	11
12	Менеджеры компоновки. Контейнерные элементы управления	4	4	2	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>	10
13	Событийно-управляемое программирование. Событийная модель Java (C#)	4	8	6	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

14	Классы-адаптеры	2		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>	4
15	Анонимные классы	2		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>	4
16	Классы-оболочки	2		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>	4
17	Массивы	4	6	4	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>	14
18	Строки	4	5	4	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>	13
19	Дополнительные элементы управления	2	4	4	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>	10
20	Классы, обеспечивающие программирование графики	3	4	2	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>	9



	Промежуточная аттестация						44
	Консультации						3
	Контроль						18
	<b>ИТОГО (в часах)</b>						<b>252</b>

#### **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.

### **V. 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. – ISBN 978-5-7695-6856-5.

2. 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.

3. Казанский, А. А. Программирование на Visual C# : учебное пособие для вузов / А. А. Казанский. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2020. — 192 с. — (Высшее образование). — ISBN 978-5-534-12338-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/451467>.

4. Конова, Е. А. Алгоритмы и программы. Язык C++ [Электронный ресурс] / Е. А. Конова. - Москва : Лань", 2016. - Режим доступа: [http://e.lanbook.com/books/element.php?pl1\\_id=72986](http://e.lanbook.com/books/element.php?pl1_id=72986). - Режим доступа: ЭБС "Издательство Лань". - Неогранич. доступ. - ISBN 978-5-8114-2020-9.

5. Кудрина, Е. В. Основы алгоритмизации и программирования на языке C# : учебное пособие для вузов / Е. В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2021. — 322 с. — (Высшее образование). — ISBN 978-5-534-09796-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/475189>.

6. Подбельский, В. В. Программирование. Базовый курс C# : учебник для вузов / В. В. Подбельский. — Москва : Издательство Юрайт, 2020. — 369 с. — (Высшее образование). — ISBN 978-5-534-10616-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/450868>.

7. Семакин, И. Г. Основы алгоритмизации программирования [Текст] : учебник / И. Г. Семакин, А. П. Шестаков. – 2-е изд., стер. – М. : Академия, 2011. – 392 с. – ISBN 978-5-7695-8187-8 всего 10 экз.

8. Тузовский, А. Ф. Объектно-ориентированное программирование : учебное пособие для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2021. — 206 с. — (Высшее образование). — ISBN 978-5-534-00849-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/470223>.

#### **б) дополнительная литература**

1. 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. - ISBN 978-5-7695-8144-1.

2. 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. – ISBN 978-5-9624-0705-0 всего 56 экз.

**в) 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 «Оттиск», 2017. – 80 с.

**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.

**VI. 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
<b>Поточные аудитории (Учебный корпус №11, 664011, Иркутская область, г. Иркутск, ул. Нижняя Набережная, д. 6)</b>	
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
<b>Учебные и специализированные кабинеты (Учебный корпус №11, 664011, Иркутская область, г. Иркутск, ул. Нижняя Набережная, д. 6)</b>	
246	Компьютер BEENEXT-45G-12 (Системный блок, Монитор Beng TET22''G2200W)-60 шт; Проектор ViewSonic PJD8633WS.DLP projector.ultra- Short-Throw Lens 1280*800; Экран Screen Media Cololview; Шкаф настенный металлический; Доска аудиторная ДА 32 белая 3032*1012
306	Системный блок ATN Core is (Монитор LCD 21.5 Viewsonic)- 23 шт; Персональный компьютер "Система", Монитор Philips 21,5 226V4LSB – 21 шт; Интерактивный учебный комплекс SMART Technologies Smart Board 685ix/UH60; Коммутатор D-Link DGS-1024 D; Коммутатор D-Link DGS-1024 C/B1A24 G неуправляемый; Доска аудиторная ДА-12 белая 1512 x 1012
309	Системный блок в сборе – 25 шт.; Монитор 23,8 Acer V246HYLBD – 25 шт; Доска аудиторная ДА-12 белая 1512*1012
312	Системный блок в сборе, монитор 23,8 Acer V246HYLBD-22шт.; Доска аудиторная ДА-12 белая 1512*1012; Интерактивная доска Smart Board 680; Мультимедиа-проектор EPSON EMP-830

## 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), 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).	2
2	Характеристика технологии Java (или. 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 конструкции языка Java (или С#)	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	2
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
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
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).	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
8	Графический пользовательский ин-	Лекция	Презентация материала с применением вспомогательных средств (интерактивного обо-	1

	терфейс на Java (или C#)		рудования и средств ИКТ) с возможностью его последующего обсуждения (демонстрационно-дискуссионная форма взаимодействия участников).	
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).	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).	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).	2
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).	4
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).	4
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).	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).	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).	2
17	Массивы	Лекция	Презентация материала с применением вспомогательных средств (интерактивного оборудования и средств ИКТ) с возможностью его последующего обсуждения (демонстра-	4

			ционно-дискуссионная форма взаимодействия участников).	
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).	4
Итого часов:				52

## 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 выполнять педагогическую деятельность по проектированию и реализации образовательного процесса в образовательных организациях основного общего, среднего общего образования</i>	Владеет содержанием педагогической деятельности, необходимым для разработки учебно-методического обеспечения, дисциплин программ основного общего, среднего общего образования	Демонстрация владения теоретическим материалом предметной области алгоритмизации и программирования, необходимым для разработки учебно-методического обеспечения дисциплин программ основного общего, среднего общего образования	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	Индивидуальные задания

		для преподавания по программам учебных предметов, соответствующих направленности	
	Осуществляет урочную и внеурочную деятельность по дисциплинам предметной области знаний	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 к применению теоретических знаний и практических умений в преподаваемой предметной области</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.

*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:**  $balance_0 = initialBalance$ ,

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

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

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 от»

	рицательной! », 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.

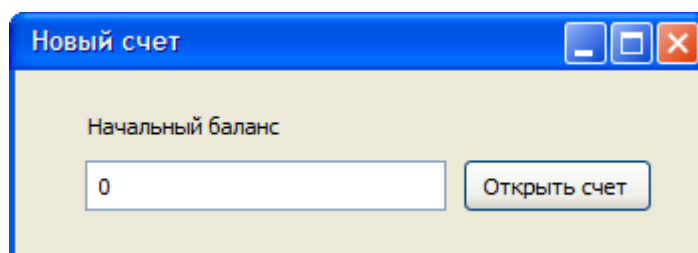


Рисунок 1 – Форма «Новый счет»

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)

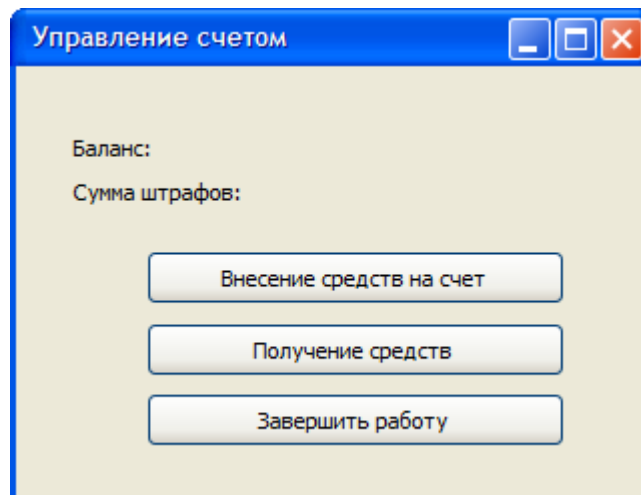


Рисунок 2 – Форма «Управление счетом»

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).

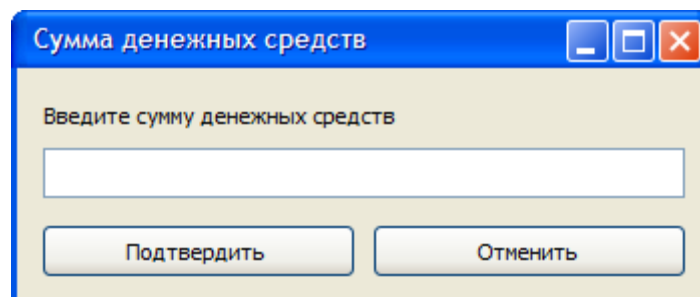


Рисунок 3 – Форма «Сумма денежных средств»

Описание входной информации представлено в таблице 3.

Таблица 3

Описание входных данных для формы «Сумма денежных средств»

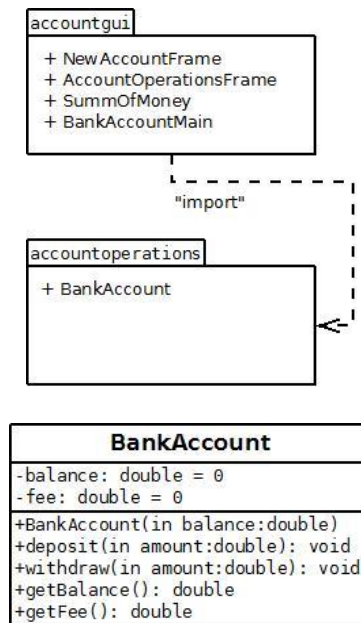
Наименование величины (данных)	Тип данных	Формат представления	Ограничения	Примечание
Сумма денежных средств	Числовой	0,00	Положительное значение	Задается в рублях

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
	$initialBalance$	$amount_i$	$operation_i$	$balance_i$	$fee_i$	
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;
    }

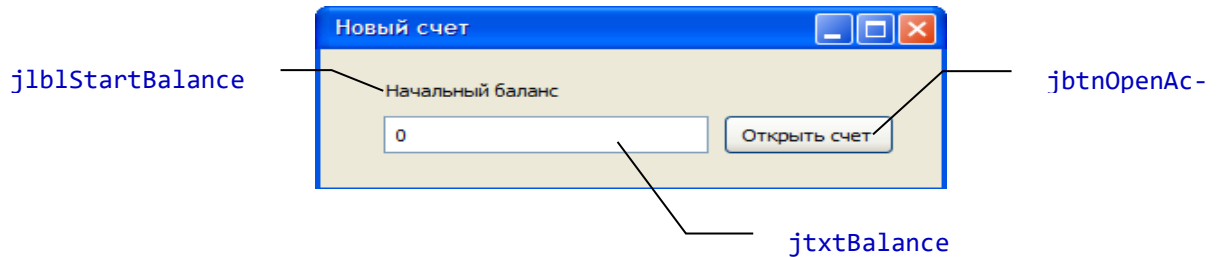
    public void withdraw(double amount){
        if (this.balance - amount >= 0)
            balance -= amount;
        else{
            this.balance -= (5.0 + amount);
            this.fee +=5;
        }
    }

    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	<code>jlblStartBalance</code>			Метка с надписью «Начальный баланс»
JTextField	<code>jtxtBalance</code>			Однострочное текстовое поля для ввода суммы баланса
JButton	<code>jbtnOpenAccount</code>	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) {
        double a = Double.parseDouble(jtxtBalance.getText());
        if (a > 0){
            account = new BankAccount(a);
            JOptionPane.showMessageDialog(NewAccountFrame.this,
"Счет открыт!");

            NewAccountFrame.this.setVisible(false);
            (new AccountOpera-
tionsFrame(account)).setVisible(true);

        }
        else{
            JOptionPane.showMessageDialog(NewAccountFrame.this,
"Денежная сумма не может быть отрицательной!");
            jtxtBalance.requestFocus();
            jtxtBalance.selectAll();
        }
    }

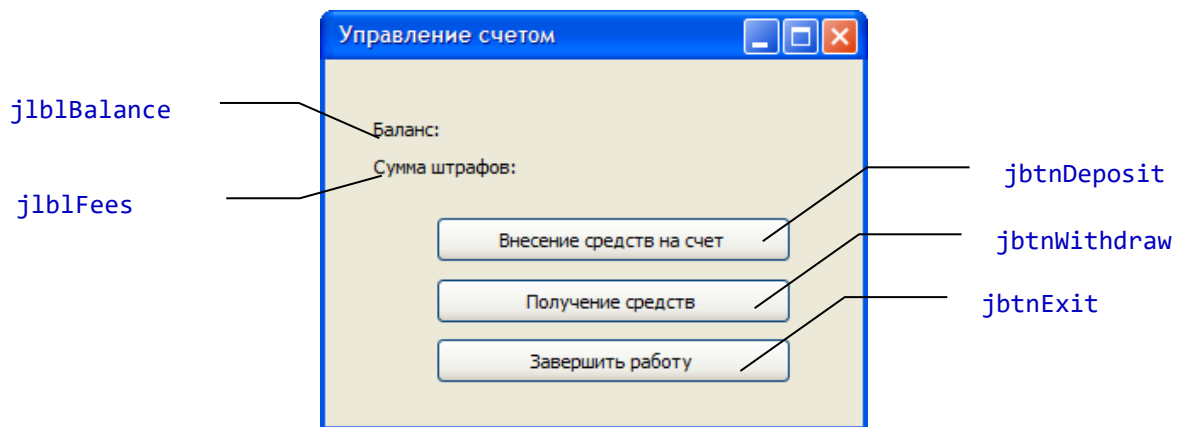
});

}

}

```

**Листинг класса AccountOperationsFrame**



Класс	Имя объекта	Интерфейс слушателя	Метод-обработчик события	Примечание
JLabel	<code>jlblBalance</code>			Метка для вывода суммы на балансе счета.
	<code>jlblFees</code>			Метка для вывода суммы штрафов.
JButton	<code>jbtnDeposit</code>	ActionListener	ActionPerformed(ActionEvent e)	Обработка щелчка мышью на кнопку с надписью «Внесение средств на счет», пополнение счета.
	<code>jbtnWithdraw</code>	ActionListener	ActionPerformed(ActionEvent e)	Обработка щелчка мышью на кнопку с надписью «Получение средств», списание суммы со счета.
	<code>jbtnExit</code>	ActionListener	ActionPerformed(ActionEvent e)	Обработка щелчка мышью на кнопку с надписью «Завершить работу», завершение работы приложения
NewAccountFrame	<code>this</code>	WindowListener	windowActivated(WindowEvent e)	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: ");
    jbbtnDeposit = new JButton("Внесение средств на счет");
    jbbtnWithdraw = new JButton("Получение средств");
    jbbtnWithdraw.setPreferredSize(jbbtnDeposit.getPreferredSize());
    jbbtnExit = new JButton("Завершить работу");
    jbbtnExit.setPreferredSize(jbbtnDeposit.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.BASELINE_LEADING;
    this.add(jlblFees , gbc);

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

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

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

    this.addWindowListener(new WindowAdapter(){

        public void windowActivated(WindowEvent arg0) {
            jlblBalance.setText("Баланс: "+account.getBalance() + "
рублей");

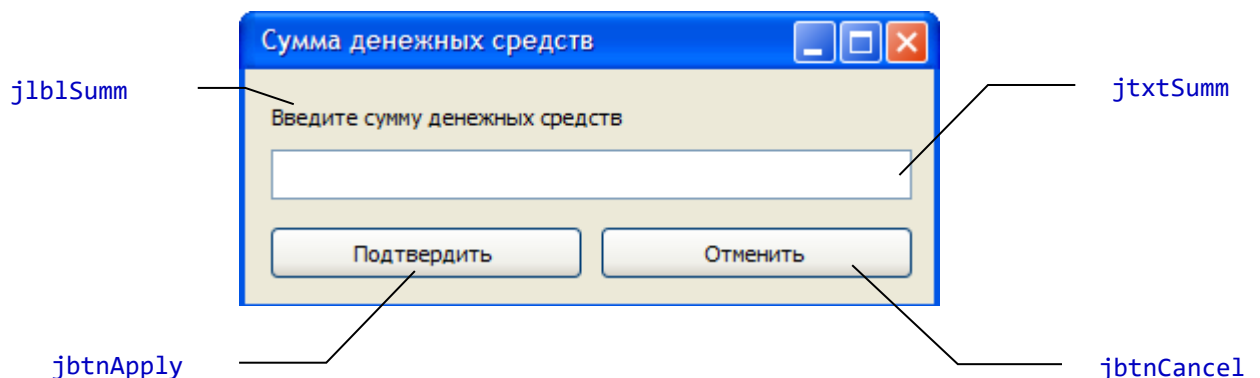
```

```

        рублей");
        jlblFees.setText("Сумма штрафов: "+ account.getFees() + "
    }
});
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 поле для

				ввода суммы
JButton	<code>jbtnApply</code>	ActionListener	ActionPerformed(ActionEvent e)	Обработка щелчка мышью на кнопку с надписью «Подтвердить», выполнение операции
	<code>jbtnCancel</code>	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);
        jbtnApply = new JButton("Подтвердить");
        jbtnCancel = new JButton("Отменить");
        jbtnCancel.setPreferredSize(jbtnApply.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.BASILINE;
this.add(jbtnApply, gbc);

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

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

jbtnApply.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();
        }
        else {
            JOptionPane.showMessageDialog(SummOfMoney.this, "Де-
нежная сумма не может быть отрицательной!");
            jtxtSumm.requestFocus();
            jtxtSumm.selectAll();
        }
    }

});

jbtnCancel.addActionListener(new ActionListener(){

    public void actionPerformed(ActionEvent arg0) {
        SummOfMoney.this.dispose();
    }

});

}

}

```

#### Листинг класса BankAccounMain

```

package accountgui;

public class BankAccounMain {

```

```

    public static void main(String[] args) {
        NewAccountFrame startframe = new NewAccountFrame();
        startframe.setVisible(true);
    }
}

```

## 8.2. Оценочные средства для промежуточной аттестации (в форме экзамена или зачета).

### Вопросы к экзамену

1. Объектно-ориентированная декомпозиция задачи. Объектно-ориентированное программирование (идея, базовые понятия, принципы).
2. Характеристика технологии Java (либо .Net). Компиляция исходного кода, написанного на языке Java (либо C#), выполнение приложения.
3. Элементарные конструкции языка программирования Java (либо C#) (алфавит, переменные, константы (литералы), лексемы).
4. Структура программы на языке Java (либо C#).
5. Типы данных в Java (либо C#) (понятия типа данных, специфика хранения величин простых и ссылочных типов данных в оперативной памяти, простые типы данных в Java (либо C#) и их характеристика).
6. Описание переменных и констант. Инициализация переменных в теле метода. Арифметические операции над величинами целого и вещественного типов. Понятие пакета в Java (пространства имен в C#). Математические функции.
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 С#). 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 (С#): 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"> <li>– дан верный и полный ответ на теоретический вопрос из списка вопросов к экзамену;</li> <li>– зачтено выполнение всех индивидуальных заданий;</li> <li>– обучающийся проявил владение каждой из компетенций, развиваемых в рамках дисциплины, на уровне не менее 60% в процессе решения полученной на экзамене типовой задачи.</li> </ul>
Хорошо	<ul style="list-style-type: none"> <li>– дан верный, но неполный ответ на теоретический вопрос из списка вопросов к экзамену;</li> <li>– зачтено выполнение всех индивидуальных заданий;</li> <li>– обучающийся проявил владение каждой из компетенций, развиваемых в рамках дисциплины, на уровне не менее 60% в процессе решения полученной на экзамене типовой задачи.</li> </ul> <p><b>либо</b></p> <ul style="list-style-type: none"> <li>– дан верный и полный ответ на теоретический вопрос из списка вопросов к экзамену</li> <li>– зачтено выполнение от 80 до 95 % всех индивидуальных заданий;</li> <li>– обучающийся проявил владение каждой из компетенций, развиваемых в рамках дисциплины, на уровне не менее 60% в процессе решения полученной на экзамене типовой задачи.</li> </ul>
Удовлетворительно	<ul style="list-style-type: none"> <li>– дан ответ на теоретический вопрос из списка вопросов к экзамену, во время ответа имелись недочеты, ошибки в определениях, классификациях, выводах (в целом представление сформировано)</li> <li>– зачтено выполнение не менее 60% всех индивидуальных заданий;</li> <li>– обучающийся проявил владение каждой из компетенций, развиваемых в рамках дисциплины, на уровне не менее 60% в про-</li> </ul>

	цессе решения полученной на экзамене типовой задачи.
Неудовлетворительно	<ul style="list-style-type: none"> <li>– дан неверный ответ на теоретический вопрос из списка вопросов к экзамену, во время ответа имелись существенные недостатки в определениях, классификациях, выводах;</li> <li>– 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.</li> </ul>

Документ составлен в соответствии с требованиями ФГОС по направлению 44.03.05 «Педагогическое образование (с двумя профилями подготовки)», утвержденного приказом Минобрнауки РФ №125 от 22 февраля 2018 г.

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