



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ 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

«21» июня 2018 г.

Рабочая программа дисциплины (модуля)

Наименование дисциплины (модуля)

Б1.В.ОД.20 Языки и методы программирования

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

Тип образовательной программы *академический бакалавриат*

Направленность (профиль) подготовки *Математика-Информатика*

Квалификация (степень) выпускника - *бакалавр*

Форма обучения *заочная*

Согласовано с УМС ПИ ИГУ

Протокол №9

от «20» июня 2018 г.

Председатель _____ М.С. Павлова

Рекомендовано кафедрой Информатики и методики обучения информатике

Протокол № 20

от «2» июня 2018 г.

Зав. кафедрой _____ Е.Н. Иванова

Иркутск 2018 г.

Содержание

	стр.
1. Цели и задачи дисциплины (модуля)	3
2. Место дисциплины (модуля) в структуре ОПОП.	3
3. Требования к результатам освоения дисциплины (модуля)	3
4. Объем дисциплины (модуля) и виды учебной работы	4
5. Содержание дисциплины (модуля)	4
6. Перечень практических занятий, лабораторных работ, план самостоятельной работы студентов, методические указания по организации самостоятельной работы студентов	7
7. Примерная тематика курсовых работ (проектов) (при наличии)	9
8. Учебно-методическое и информационное обеспечение дисциплины (модуля): а) основная литература; б) дополнительная литература; в) программное обеспечение; г) базы данных, поисково-справочные и информационные системы	9
9. Материально-техническое обеспечение дисциплины (модуля)	10
10. Образовательные технологии	11
11. Оценочные средства (ОС)	12

1. Цели и задачи дисциплины (модуля):

– систематизация и углубление методологических знаний и умений в предметной области, связанной с разработкой программного обеспечения

Задачи дисциплины:

- систематизировать методы и приемы решения задач по программированию;
- дать представление об объектно-ориентированном программировании;
- развить и углубить общие представления о языках и системах программирования и возможностях их применения в различных областях деятельности человека как одного и эффективных инструментов информационного моделирования.

2. Место дисциплины в структуре ОПОП:

Дисциплина «Языки и методы программирования» относится к обязательным дисциплинам вариативной части профессионального цикла учебного плана. Для изучения данной дисциплины студент должен владеть знаниями и умениями, полученными в рамках таких дисциплин как «Основы алгоритмизации» («Этапы решения задач на ЭВМ»), «Языки программирования высокого уровня» («Инструментальные системы разработки ПО»). Дисциплина «Языки и методы программирования» является предшествующей и преемственной для таких дисциплин как «Практикум по решению задач на ЭВМ», «Объектно-ориентированные технологии», «Педагогические программные средства», «Компьютерное моделирование», «Компонентная модель Java».

3. Требования к результатам освоения дисциплины (модуля):

Процесс изучения дисциплины «Языки и методы программирования» направлен на развитие следующих компетенций:

ОК-3 - способность использовать естественнонаучные и математические знания для ориентирования в современном информационном пространстве;

СПКИ-1 - готовность к применению теоретических знаний и практических умений в профессиональной деятельности.

ПК-11 - готовностью использовать систематизированные теоретические и практические знания для постановки и решения исследовательских задач в области образования.

Знать:

- компонентные модели современных платформ программирования, их структурную организацию (ОК-3, СПКИ-1);
- технологии конструирования объектной модели явления или процесса, описанного в задаче (ОК-3, СПКИ-1, ПК-11);
- языковые средства реализации построенной объектной модели (ОК-3, СПКИ-1, ПК-11).

Уметь:

- описывать информационную модель решаемой задачи (ОК-3, СПКИ-1, ПК-11);
- строить объектную модель явления или процесса, описываемого в задаче (ОК-3, СПКИ-1, ПК-11);
- описывать структурную и функциональную организацию предметов и явлений, заданных в условии задачи в соответствии с выбранной технологией декомпозиции задачи и технологией программирования (ОК-3, СПКИ-1, ПК-11).

Владеть:

- практическими навыками компьютерного моделирования посредством языков и систем программирования с учетом выбранной технологии разработки программного обеспечения (ОК-3, СПКИ-1, ПК-11);
- практическими навыками конструирования всех элементов программного средства в соответствии с построенной моделью явления или процесс, описанного в условии задачи (ОК-3, СПКИ-1);
- практическими навыками использования технологий разработки прикладного программного обеспечения и применения соответствующего инструментария (ОК-3, СПКИ-1).

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

Вид учебной работы	Всего часов/ зачетных единиц	Курс			
		2	3		
Аудиторные занятия (всего)	24	8	16		
В том числе:					
Лекции	12	4	8		
Практические занятия (ПЗ)					
Лабораторные работы (ЛР)	12	4	8		
Самостоятельная работа (всего)*	251	60	191		
В том числе:					
Выполнение индивидуальных заданий по вариантам.	126	30	90		
Самостоятельный поиск информации, необходимой для выполнения индивидуальных заданий.	125	30	101		
Вид промежуточной аттестации (зачет, экзамен)	13	4	9		
Контактная работа (всего)**					
Общая трудоемкость часы зачетные единицы	288	72	216		
	8	2	6		

5. Содержание дисциплины (модуля)

5.1. Содержание разделов и тем дисциплины (модуля). Все разделы и темы нумеруются.

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. Элементарные конструкции языка программирования. Алфавит языка программирования. Требования к идентификаторам величин. Операции. Структура исходно-

го кода приложения. Комментарии к программному коду. Комментарии к программному коду. Типы данных. Характеристика простых типов данных. Описание переменных и констант. Преобразование типов данных.

2.2. Исполняемые операторы языка программирования. Способы ввода-вывода данных (консоль). Форматированный вывод. Управляющие конструкции: программирование развилки, структуры «выбор», циклов.

Раздел 3. Разработка графического пользовательского интерфейса приложения. Событийно-управляемое программирование

3.1. Характеристика библиотек, необходимых для разработки графического пользовательского интерфейса.

3.2. Основные классы, необходимые для создания стандартных элементов графического пользовательского интерфейса: форма, элементы управления (кнопка, метка, однострочное текстовое поле), контейнеры, элементы управления альтернативного выбора, мен и т.д.

3.3. Событийно-управляемое программирование. Понятие события.

Раздел 4. Структурированные типы данных

4.1. Массивы. Объединения (множества), записи.

4.2. Строки.

4.3. Файлы.

Раздел 5. Графические возможности языков программирования

5.1. Характеристика библиотек (накетов), необходимых для работы с графикой. Создание примитивов, работа с битовыми изображениями и т.д.

5.2. Таймер.

5.2 Разделы дисциплины и междисциплинарные связи с обеспечиваемыми (последующими) дисциплинами

№ п/п	Наименование обеспечиваемых (последующих) дисциплин	№№ разделов и тем данной дисциплины, необходимых для изучения обеспечиваемых (последующих) дисциплин (вписываются разработчиком)								
		1	2	3	4	5	6			
1	Практикум по решению задач на ЭВМ	1	2	3	4	5	6			
2	Объектно-ориентированные технологии	1	2	3	4	5	6			
3	Педагогические программные средства	1	2	3	4	5	6			
4	Компьютерное моделирование	1	2	3	4	5	6			
7	Технология Java	1	2	3	4	5	6			
8	Компонентная модель Java	1	2	3	4	5	6			

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

№ п/п	Наименование раздела	Наименование темы	Виды занятий в часах					
			Лекц.	Практ. зан.	Семина	Лаб. зан.	СРС	Всего
1	Алгоритмическая и объектно-ориентированная декомпозиция задачи	Понятие об объектно-ориентированном программировании.	2					2
		Принципы ООП	2					2
2	Элементарные и управляющие конструкции языков программирования	Элементарные конструкции языка программирования. Требования к идентификаторам величин. Операции. Типы данных. Характеристика простых типов данных. Описание переменных и констант. Преобразование типов данных.	4			2	24	30
		Исполняемые операторы языка программирования. Способы ввода-вывода данных (консоль). Управляющие конструкции: программирование развилки, структуры «выбор», циклов.	4			4	26	34
3	Разработка графического пользовательского интерфейса приложения. Событийно-управляемое программирование	Характеристика библиотек, необходимых для разработки графического пользовательского интерфейса.					10	10
		Основные классы, необходимые для создания стандартных элементов графического пользовательского интерфейса.					40	40
		Событийно-управляемое программирование. Понятие события.					20	20

4	Структурированные типы данных	Массивы. Объединения (множества), записи.				4	40	44
		Строки				2	30	32
		Файлы					20	20
5	Графические возможности языков программирования	Характеристика библиотек (пакетов), необходимых для работы с графикой. Создание примитивов, работа с битовыми изображениями и т.д.					20	20
		Таймер					21	21

6. Перечень практических занятий и лабораторных работ

№ п/п	№ раздела и темы дисциплины (модуля)	Наименование семинаров, практических и лабораторных работ	Трудоемкость (час.)	Оценочные средства	Формируемые компетенции
1	2	3	4	5	6
1	2	Элементарные конструкции языка	2	Отчет по лабораторной работе	ОК-3, СПКИ-1, ПК-11
2	2	Управляющие конструкции языка программирования	4	Отчет по лабораторной работе	ОК-3, СПКИ-1, ПК-11
3	4	Массивы	4	Отчет по лабораторной работе	ОК-3, СПКИ-1, ПК-11
4	4	Строки	2	Отчет по лабораторной работе	ОК-3, СПКИ-1, ПК-11

6.1. План самостоятельной работы студентов

№ нед.	Тема	Вид самостоятельной работы	Задание	Рекомендуемая литература	Количество часов
1	Элементарные конструкции языка программирования. Требования к идентификаторам величин. Операции. Типы данных. Характеристика простых типов данных. Описание переменных и констант. Преобразование типов дан-	Выполнение индивидуальных заданий по вариантам. Самостоятельный поиск информации, необходимой для выполнения индивидуальных заданий.	Задания по вариантам.	№№ 1-4 из списка основной литературы. №№ 1-2 из списка дополнительной литературы. Ресурсы сети Интернет.	24

	ных.				
2	Исполняемые операторы языка программирования. Способы ввода-вывода данных (консоль). Управляющие конструкции: программирование развилки, структуры «выбор», циклов.	Выполнение индивидуальных заданий по вариантам. Самостоятельный поиск информации, необходимой для выполнения индивидуальных заданий.	Задания по вариантам.	№№ 1-4 из списка основной литературы. №№ 1-2 из списка дополнительной литературы. Ресурсы сети Интернет.	26
3	Характеристика библиотек, необходимых для разработки графического пользовательского интерфейса.	Выполнение индивидуальных заданий по вариантам. Самостоятельный поиск информации, необходимой для выполнения индивидуальных заданий.	Задания по вариантам.	№№ 1-4 из списка основной литературы. №№ 1-2 из списка дополнительной литературы. Ресурсы сети Интернет.	10
4	Основные классы, необходимые для создания стандартных элементов графического пользовательского интерфейса.	Выполнение индивидуальных заданий по вариантам. Самостоятельный поиск информации, необходимой для выполнения индивидуальных заданий.	Задания по вариантам.	№№ 1-4 из списка основной литературы. №№ 1-2 из списка дополнительной литературы. Ресурсы сети Интернет.	40
5	Событийно-управляемое программирование. Понятие события.	Выполнение индивидуальных заданий по вариантам. Самостоятельный поиск информации, необходимой для выполнения индивидуальных заданий.	Задания по вариантам.	№№ 1-4 из списка основной литературы. №№ 1-2 из списка дополнительной литературы. Ресурсы сети Интернет.	20
6	Массивы. Объединения (множества), записи.	Выполнение индивидуальных заданий по вариантам. Самостоятельный поиск информации, необходимой для выполнения индивидуальных заданий.	Задания по вариантам.	№№ 1-4 из списка основной литературы. №№ 1-2 из списка дополнительной литературы. Ресурсы сети Интернет.	40
7	Строки	Выполнение индивидуальных заданий по вариантам. Самостоя-	Задания по вариантам.	№№ 1-4 из списка основной литературы. №№ 1-2 из спи-	30

		тельный поиск информации, необходимой для выполнения индивидуальных заданий.		ска дополнительной литературы. Ресурсы сети Интернет.	
8	Файлы	Выполнение индивидуальных заданий по вариантам. Самостоятельный поиск информации, необходимой для выполнения индивидуальных заданий.	Задания по вариантам.	№№ 1-4 из списка основной литературы. №№ 1-2 из списка дополнительной литературы. Ресурсы сети Интернет.	20
9	Характеристика библиотек (пакетов), необходимых для работы с графикой. Создание примитивов, работа с битовыми изображениями и т.д.	Выполнение индивидуальных заданий по вариантам. Самостоятельный поиск информации, необходимой для выполнения индивидуальных заданий.	Задания по вариантам.	№№ 1-4 из списка основной литературы. №№ 1-2 из списка дополнительной литературы. Ресурсы сети Интернет.	20
10	Таймер	Выполнение индивидуальных заданий по вариантам. Самостоятельный поиск информации, необходимой для выполнения индивидуальных заданий.	Задания по вариантам.	№№ 1-4 из списка основной литературы. №№ 1-2 из списка дополнительной литературы. Ресурсы сети Интернет.	21

6.2. 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.

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.

7. Примерная тематика курсовых работ (проектов) (при наличии): курсовая работа не предусмотрена.

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

4. 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.

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) программное обеспечение:

ОС MS Windows, антивирус, офисный пакет LibreOffice, OpenOffice, MS Office, архиватор PeaZip, медиа-плеер VLC, браузер Mozilla Firefox, WinDjView, XnView MP, Acrobat Reader, системы программирования Eclipse, NetBean.

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

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

9. Материально-техническое обеспечение дисциплины (модуля):

Специальные помещения:

учебная аудитория для проведения занятий лекционного и семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, укомплектованная специализированной учебной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории (неограниченный доступ к сети Интернет); помещение для самостоятельной работы, оснащенное компьютерной техникой с возможностью подключения к сети Интернет и доступом к электронной информационно-образовательной среде организации.

Техническое обеспечение:

Компьютер, проектор, экран, доска аудиторная, интерактивная доска.

10. Образовательные технологии:

В рамках дисциплины применяется традиционные технологии обучения (объяснительно-иллюстративные технологии) в сочетании с технологиями, основанными на повышении эффективности управления и организации учебного процесса, а именно компьютерные (новые информационные) технологии обучения.

По видам занятий, предусмотренных учебным планом, и видам учебной деятельности, реализуемой в рамках настоящей дисциплины, компьютерные технологии обучения характеризуются следующими аспектами их применения:

Вид занятия	Вид деятельности студента	Компьютерные технологии обучения (технологическое направление)
Лекция	Восприятие и обработка информации (перцептивная деятельность).	1. Применение интерактивных технических средств обучения при объяснении материала. 2. Информационная поддержка учебного процесса посредством хранилищ данных, портала.
Лабораторная работа	Изменение воспринятой и запомненной информации, ее применение с учетом новых условий, либо получение новой информации (продуктивная деятельность)	1. Применение интерактивных технических средств при демонстрации применения различных методов, способов и приемов решения практических (профессиональных) задач. 2. Применение специального программного обеспечения, необходимого для решения профессиональных задач и проведения компьютерных (численных) экспериментов. 3. Информационная поддержка учебного процесса посредством хранилищ данных, портала.

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.

11. Оценочные средства (ОС):

11.1. Оценочные средства для определения степени сформированности компетенций студентов

№ п/п	Вид контроля	Контролируемые темы (разделы)	Компетенции, компоненты которых контролируются	Оценочные средства
1	Текущий	Все разделы	ОК-3, СПКИ-1, ПК-11	Отчет по индивидуальным заданиям.
2	Промежуточный (в форме зачета с оценкой)	Раздел 1. Алгоритмическая и объектно-ориентированная декомпозиции задачи Раздел 2. Элементарные и управляющие конструкции языков программирования Раздел 3. Введение в объектно-ориентированное программирование Раздел 4. Разработка графического пользовательского интерфейса приложения	ОК-3, СПКИ-1, ПК-11	Пакет отчетов по индивидуальным заданиям лабораторных работ
3	Промежуточный (в форме экзамена)	Все разделы	ОК-3; СПКИ-1, ПК-11	Экзаменационный билет, включающий один теоретический вопрос по содержанию дисциплины и одну задачу.

Этап формирования компетенции

Код компетенции	Этап
ОК-3	2
СПКИ-1	2
ПК-11	2

Соотнесение с совокупным ожидаемым результатом образования в компетентностном формате

Коды	Компетенции выпускника вуза как совокупный ожидаемый результат по завершении обучения по ОПОП ВО	Совокупность оценочных заданий (Даются содержательные формулировки каждого из оценочных заданий)
		Задания
1	2	3
ОК-3	способность использовать естественнонаучные и математические знания для ориентирования в современном информационном пространстве	<ul style="list-style-type: none"> – выполнение задания преподавателя в соответствии с инструкцией (аудиторные занятия); – выполнение индивидуальных заданий (задач) – подготовка отчета по лабораторной работе
СПКИ-1	готовность к применению теоретических знаний и практических умений в профессиональной деятельности.	<ul style="list-style-type: none"> – выполнение индивидуальных заданий (задач) – подготовка отчета по лабораторной работе
ПК-11	готовностью использовать систематизированные теоретические и практические	<ul style="list-style-type: none"> – выполнение индивидуальных заданий (задач) – подготовка отчета по лабораторной работе

	знания для постановки и решения исследовательских задач в области образования	
--	---	--

Карта оценки компетенций

Шифр компетенции и ее содержание	Показатели (наблюдаемые признаки)	Критерии	Вид оценочного средства
<i>ОК-3 – способность использовать естественнонаучные и математические знания для ориентирования в современном информационном пространстве</i>	Способность выполнять анализ и декомпозицию решаемой задачи (независимо от предметной области) с целью поиска ее решения	Выделена предметная область задачи	Индивидуальное задание (задача)
		Выделено требование задачи	
		Выделены качественные характеристики объектов задачи	
		Выделены количественные характеристики объектов задачи	
		Указаны существенные характеристики объектов задачи для достижения требования задачи	
		Сделан вывод о корректности содержательной постановки задачи	
	Способность преобразовывать содержательную постановку задачи в формальную посредством формальных языков (язык математики)	Выделены исходные данные задачи	
		Определена ОДЗ величин в составе исходных данных	
		Выделены выходные данные задачи	
		Определена ОДЗ величин в составе выходных данных	
		Установлена связь между выходными и исходными данными	
		Обоснована адекватность построенной математической модели	
	Способность формулировать условия и ограничения на функционирование будущей компьютерной модели (программы) явления или процесса, описанной в задаче	Определена спецификация величин, участвующих в построении алгоритма решения задачи	
		Сформулированы условия и ограничения на функционирование будущей компьютерной модели (программы, приложения)	
	Способность осуществлять выбор метода решения задачи	Обоснован и описан метод решения задачи	
	Способность описывать алгоритм решения либо объектную модель по-	Приведено описание алгоритма решения задачи (либо объектной модели) в	

	ставленной задачи	соответствии с этапами работы с конкретным программным средством, выбранным для построения компьютерной модели (программы, приложения)	
		Произведено преобразование алгоритма решения задачи в систему команд исполнителя (на язык программирования и т.п.)	
<i>СПКИ-1 - готовность к применению теоретических знаний и практических умений в профессиональной деятельности.</i>	Готовность осуществлять контроль и оценку правильности решаемой задачи	Выполнен подбор тестов для проверки функционирования компьютерной модели (программы, приложения) явления или процесса, описанного в задаче	Индивидуальное задание (задача)
		Произведена отладка программы на системе тестов либо проведен компьютерный эксперимент	
	Готовность корректировать собственную деятельность в контексте решаемой задачи и устранять допущенные ошибки	Выявлены типы ошибок, допущенных в ходе построения алгоритма решения задачи либо построения компьютерной модели (программы, приложения и т.п.)	
		Устранены выявленные ошибки	
	Умение интерпретировать полученные в ходе решения результаты и оценивать их адекватность	Установлена связь между полученными в ходе компьютерного моделирования данными и характеристиками объектов задачи	
		Сформулированы выводы по результатам тестирования компьютерной модели (программы, приложения)	
<i>ПК-11 - готовностью использовать систематизированные теоретические и практические знания для постановки и решения исследовательских задач в области образования</i>	Умение осуществлять декомпозицию задачи по программированию с целью построения компьютерной модели явления или процесса, описанного в условии, для дальнейшего его исследования согласно требованию задачи	Умение выполнять решение задачи в соответствии с этапами решения задач на ЭВМ	Индивидуальное задание (задача)
		Умение проводить компьютерный эксперимент и формулировать выводы об исследуемом явлении или процессы на основе получаемых результатов.	

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.

11.2. Оценочные средства для проведения текущего контроля

Текущий контроль осуществляется в виде оценки (зачтено/не зачтено) индивидуальных задач по определенной теме, предлагаемых студенту для самостоятельного решения. Индивидуальные задания являются частью лабораторных работ. Решение каждой задачи оформляется в форме отчета согласно этапам решения задач на ЭВМ.

Форма отчета по 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 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. $(initialBalance \notin R) \vee \vee (\exists i = \overline{1, n}, (amount_i \notin R) \wedge (operation_i \in \{1, 2\}))$	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.

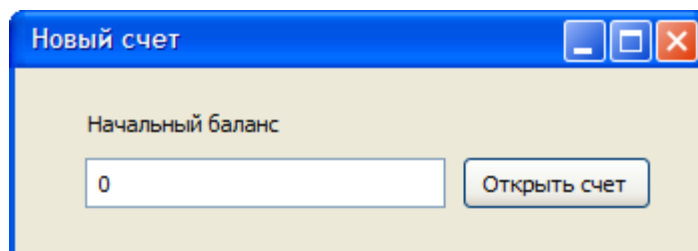


Рисунок 1 – Форма «Новый счет»

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)

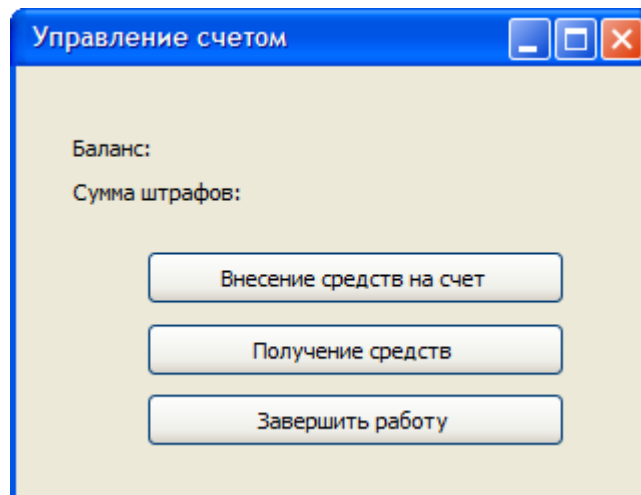


Рисунок 2 – Форма «Управление счетом»

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).

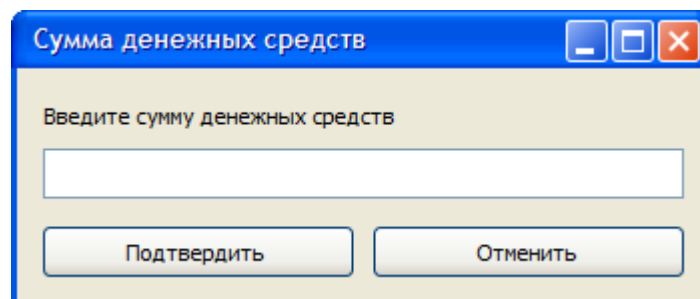


Рисунок 3 – Форма «Сумма денежных средств»

Описание входной информации представлено в таблице 3.

Таблица 3

Описание входных данных для формы «Сумма денежных средств»

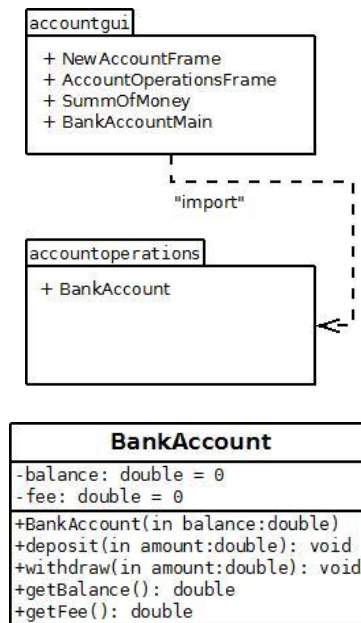
Наименование величины (данных)	Тип данных	Формат представления	Ограничения	Примечание
Сумма денежных средств	Числовой	0,00	Положительное значение	Задается в рублях

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;
    }

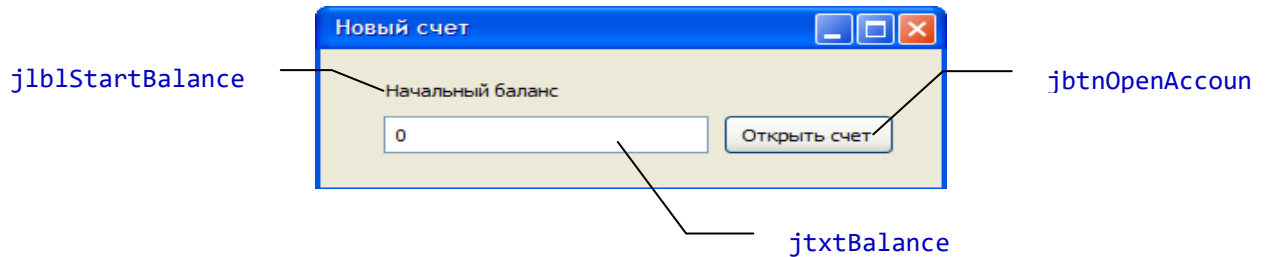
    public void withdraw(double amount){
        if (this.balance - amount >= 0)
            balance -= amount;
        else{
            this.balance -= (5.0 + amount);
            this.fee +=5;
        }
    }

    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	<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.BASILINE;
this.add(jtxtBalance , gbc);

//для кнопки
gbc.gridx = 1;
gbc.gridy = 1;
gbc.anchor = GridBagConstraints.BASILINE;
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
AccountOperationsFrame(account)).setVisible(true);

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

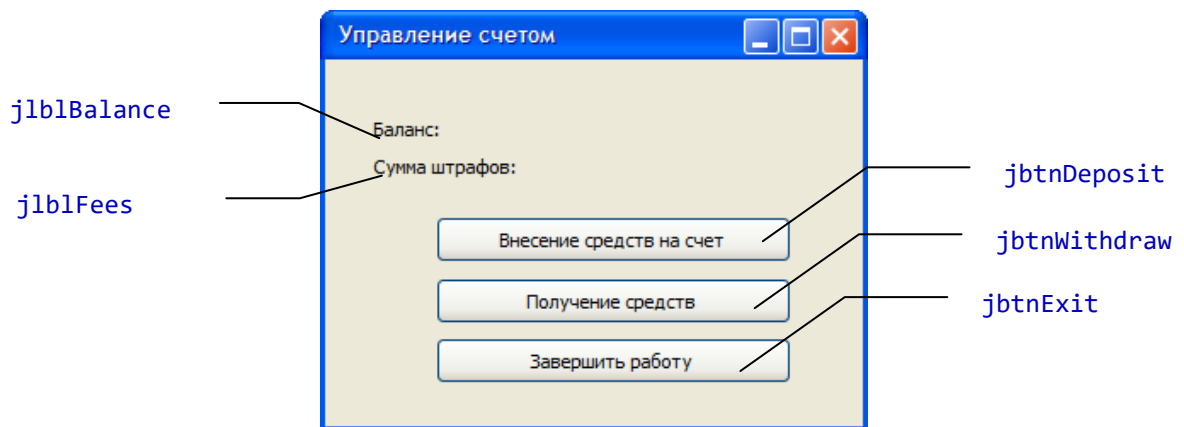
});

}

}

```

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: ");
    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.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(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) {
            jlblBalance.setText("Баланс: "+account.getBalance() + "
рублей");

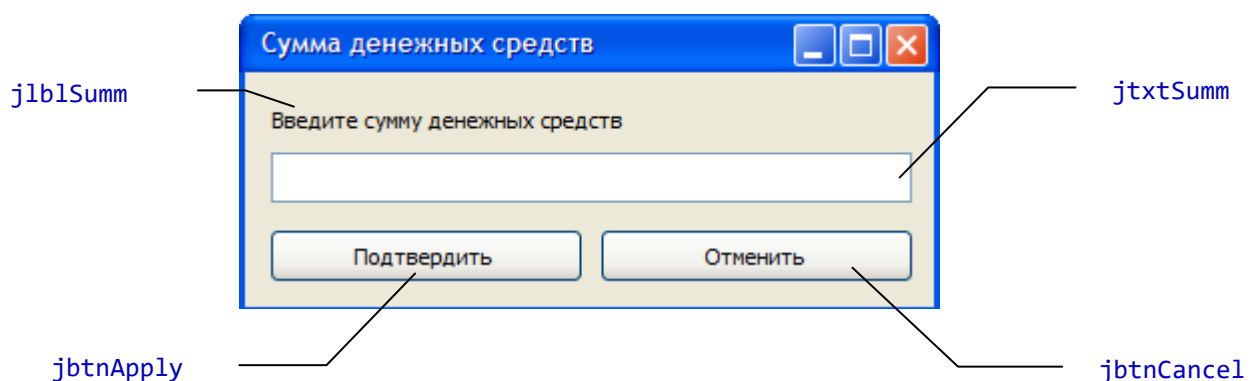
```

```

        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	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);
        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);
    }
}

```

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

Вопросы и задания 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 ссылочные 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 интерфейса 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).

Оценка за экзамен выставляется по пятибалльной шкале как среднее арифметическое оценок по каждому из следующих заданий:

1. Ответить на теоретический вопрос.
2. Решить задачу.

Ответ на теоретический вопрос оценивается по пятибалльной шкале:

- 2 балла – ответ на теоретический вопрос не получен;
- 3 балла – теоретический вопрос раскрыт не полностью либо с существенными недостатками такими, как отсутствие верных формулировок определений понятий, отсутствия обоснования утверждений, наличия противоречий в ходе изложения теоретических положений и т.п.;
- 4 балла – получен полный ответ по теоретическому вопросу, но имеются такие недостатки как исключение существенных признаков в формулировках определений рассматриваемых понятий, утверждений, ошибки в классификации понятий (неверный классификационный признак), ошибки в описании структур, отсутствие примеров и т.п.;
- 5 баллов – получен полный ответ, отражающий последовательное и логичное изложение теоретического материала, сопровождающееся примерами.

Решение задачи оценивается по пятибалльной шкале:

- 2 балла – среднее арифметическое уровней сформированности компетенций, проявленных в ходе решения задачи, меньше 60%;
- 3 балла – среднее арифметическое уровней сформированности компетенций, проявленных в ходе решения задачи, больше либо равно 60% , но меньше 70%;
- 4 балла – среднее арифметическое уровней сформированности компетенций, проявленных в ходе решения задачи, больше либо равно 70% , но меньше 85%;
- 5 баллов – среднее арифметическое уровней сформированности компетенций, проявленных в ходе решения задачи, не менее 85%.

Разработчик:

доцент Лесников И.Н.

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