

Белорусский государственный университет

УТВЕРЖДАЮ

Проректор по учебной работе и
образовательным инновациям

О.И. Чуприс

2019 г.

Регистрационный № УД- 7275/уч.



ПРИКЛАДНОЕ ПРОГРАММИРОВАНИЕ

**Учебная программа учреждения высшего образования
по учебной дисциплине для специальности:**

1-31 04 04 Аэрокосмические радиоэлектронные и информационные системы и технологии

2019 г.

Учебная программа составлена на основе *ОСВО 1-31 04 04-2018* и учебного плана, регистрационный номер № *G31-223/уч. от 13.07.2018*

СОСТАВИТЕЛЬ:

А. В. Дигрис, старший преподаватель кафедры системного анализа и компьютерного моделирования факультета радиофизики и компьютерных технологий Белорусского государственного университета

РЕЦЕНЗЕНТ:

Д. А. Стрикелев, заместитель директора по производству в сфере инноваций ЗАО «Интранзишэн», кандидат технических наук

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

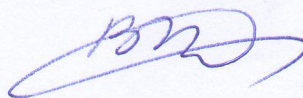
Кафедрой системного анализа и компьютерного моделирования Белорусского государственного университета

(протокол № 1 от 3 сентября 2019 г.);

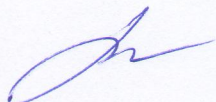
Научно-методическим Советом БГУ

(протокол № 1 от 25 сентября 2019 г.)

Заведующий кафедрой системного анализа
и компьютерного моделирования,
доцент



В.В. Скакун



ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Цели и задачи учебной дисциплины

Цель учебной дисциплины – дать студентам теоретические знания и практические навыки разработки кроссплатформенных Java-приложений с графическим интерфейсом пользователя, а также многопоточных приложений.

Задачи учебной дисциплины:

1. знакомство с платформой Java для разработки кроссплатформенных приложений;
2. освоение объектно-ориентированного подхода к программированию;
3. формирование навыков создания Java-приложений, включающих графический интерфейс пользователя;
4. получение навыков разработки многопоточных приложений на Java.
5. практическое освоение современных интегрированных сред разработки кроссплатформенных приложений на Java.

Место учебной дисциплины в системе подготовки специалиста с высшим образованием (магистра).

Учебная дисциплина относится к модулю «Методы вычислительного эксперимента» компонента учреждения высшего образования.

Учебная программа составлена с учетом межпредметных связей и программ по дисциплинам «Математический анализ», «Программирование».

Требования к компетенциям

Освоение учебной дисциплины «прикладное программирование» должно обеспечить формирование следующих специальных компетенций:

специальные компетенции:

СК-2. Уметь применять методы и алгоритмы моделирования случайных величин, случайных потоков и процессов в среде математического программирования Matlab для решения практических задач моделирования сложных процессов и систем; знать принципы построения кроссплатформенных прикладных программ с использованием Java-приложений.

В результате освоения учебной дисциплины студент должен:

знать:

- основы современных технологий программирования;
- принципы построения кроссплатформенных прикладных программ;
- принципы создания Java-приложений, обладающих пользовательским интерфейсом и управляемых событиями;
- принципы разработки собственных элементов графического пользовательского интерфейса;
- особенности построения многопоточных приложений с использованием языка программирования Java.

уметь:

- применяя объектно-ориентированный подход к программированию разрабатывать прикладные приложения с графическим интерфейсом пользователя на основе стандартных библиотек, предоставляемых платформой Java;
- разрабатывать многопоточные кроссплатформенные Java-приложения;
- творчески и эффективно использовать полученные знания в своей профессиональной деятельности.

владеть:

- средствами, предоставляемыми платформой Java, для построения приложений с графическим пользовательским интерфейсом;
- средствами для создания собственных компонент графического пользовательского интерфейса;
- средствами управления потоками выполнения в Java-приложениях;
- современными интегрированными средами разработки Java-приложений.

Структура учебной дисциплины

Дисциплина изучается в 4 семестре. Всего на изучение учебной дисциплины «прикладное программирование» отведено для очной формы получения высшего образования – 108 часов, в том числе 68 аудиторных часов, из них: лекции – 24 часа, лабораторные занятия – 40 часов, управляемая самостоятельная работа – 4 часа.

Трудоемкость учебной дисциплины составляет 3 зачетные единицы.

Форма текущей аттестации – зачет.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. Основные структурные элементы языка Java

Тема 1.1 Платформа Java. Принципы построения Java-приложений

Краткая история создания и основные особенности платформы Java. Типы платформ Java и их назначение. Интегрированные среды разработки приложений на Java.

Тема 1.2. Создание объектно-ориентированных программ на Java

Концепция объектно-ориентированного программирования и его основные принципы. Понятие класса и объектов класса. Структура программ на Java, класс как основной структурный элемент Java-программ. Создание классов и организация доступа к их компонентам. Интерфейсы и абстрактные классы. Организация наследования в Java. Встроенные классы, их категории и назначение.

Тема 1.3. Основные структурные элементы языка Java

Синтаксис языка программирования Java. Идентификаторы и требования к их оформлению. Ключевые слова и их разновидности. Задание литералов, их основные типы. Разделители в программах на Java. Основные операторы и их использование. Типы данных, понятия простых и ссылочных типов. Преобразование простых числовых типов. Организация обработки исключений.

Тема 1.4. Передача и хранение данных в Java-программах

Основные типы потоков ввода-вывода и их использование. Потоки байт и символов. Базовые классы потоков ввода-вывода и их возможности. Массивы и работа с их элементами. Особенности организации многомерных массивов в Java. Коллекции как расширение возможностей массивов. Иерархия коллекций и их группы. Интерфейс Collection и реализующие его типы коллекций. Создание и использование списков. Основные классы для создания очередей и их возможности. Создание множеств. Особенности работы с множествами на базе перечислений. Доступ к элементам коллекций, использование объектов-итераторов. Организация карт отображений, интерфейс Map. Основные классы карт отображений в Java, их особенности и назначение.

Раздел 2. Построение графического интерфейса пользователя

Тема 2.1. Основы создания графического интерфейса пользователя

Библиотеки Java для построения графического пользовательского интерфейса. Создание окон верхнего уровня, их разновидности и отличительные черты. Организация управления содержимым окна, класс JRootPane. Создание фреймов, управление их положением, размерами и оформлением. Организация древообразной структуры компонентов окна, класс JPanel. Размещение элементов внутри фрейма. Менеджеры компоновки и их основные виды. Создание диалоговых окон.

Использование классов библиотеки Java для создания диалоговых окон сообщений, настроек и работы с файлами. Разработка собственных классов диалоговых окон, организация обмена данными с диалоговыми окнами.

Тема 2.2. Компоненты графического интерфейса

Возможности библиотеки Swing для создания компонентов графического интерфейса пользователя. Создание надписей, управление внешним видом и взаимным положением текстовой и графической составляющих надписи. Создание текстовых полей, управление расположением и внешним видом текста, его доступностью для редактирования. Организация проверки правильности введенной информации в текстовом поле. Компоненты для ввода многострочного текста и их использование. Компоненты для создания кнопок, флажков и комбинированных списков. Организация основного меню приложения, создание и управление положением и доступностью пунктов меню. Создание ускорителей и мнемоник для работы с пунктами меню посредством клавиатуры. Организация реакции на действия пользователей с интерфейсом приложения. Понятия события, слушателя и источника событий. Модель делегирования обработки событий. Цикл обработки событий. Способы создания слушателей событий. Основные интерфейсы и классы библиотеки Swing для обработки событий.

Тема 2.3. Создание таблиц

Организация таблиц. Простейшие способы создания таблиц и их недостатки. Понятие модели данных таблицы, интерфейс TableModel. Создание собственных моделей данных таблицы на основе класса AbstractTableModel. Визуализаторы ячеек таблицы, интерфейс TableCellRenderer. Порядок выбора визуализатора ячеек таблицы. Основные подходы для создания собственных визуализаторов ячеек таблицы. Управление редактированием содержимого ячеек таблицы. Порядок выбора редактора ячеек таблицы. Разработка классов для создания редакторов ячеек таблицы. Интерфейс TableCellEditor и класс DefaultCellEditor. Разработка альтернативных редакторов ячеек таблицы на основе класса AbstractCellEditor. Сортировка данных в таблицах. Класс TableRowSorter и его использование для создания объектов-сортировщиков. Построение фильтров строк таблицы.

Тема 2.4. Принципы построения графических изображений

Организация вывода графической информации. Порядок прорисовки компонент графического пользовательского интерфейса. Инициирование перерисовки в ходе выполнения программного кода. Создание изображений собственных компонент графического пользовательского интерфейса. Библиотека Java 2D и ее возможности. Классы Graphics и Graphics2D. Задание стиля линий и границ геометрических фигур. Управление цветовыми схемами, интерфейс Paint. Однородные цветовые схемы, класс Color. Создание и порядок использования линейных и радиальных цветовых схем. Организация цветовых схем на основе текстур.

Тема 2.5. Рисование геометрических фигур и текстовой информации

Основные классы библиотеки Java 2D для построения геометрических фигур. Классы для рисования сегментов линий. Создание сложных линий с использованием класса GeneralPath. Рисование дуг эллипсов и окружностей. Рисование прямоугольных областей. Создание сложных областей с использованием класса Area. Рисование текста, управление шрифтами. Аффинные преобразования и их использование для управления изображением.

Раздел 3. Разработка многопоточных приложений

Тема 3.1. Создание многопоточных приложений

Понятие потока выполнения. Главный и дочерние потоки выполнения. Создание потоков выполнения в программах на Java. Класс Thread и интерфейс Runnable. Приоритеты потоков. Синхронизация потоков в Java. Концепция мониторов. Виды синхронизации потоков. Приостановка и возобновление работы потоков

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Дневная форма получения образования

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСП	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
1	Основные структурные элементы языка Java							
1.1.	Платформа Java. Принципы построения Java-приложений	2						Опрос.
1.2.	Создание объектно-ориентированных программ на Java	2			4			Опрос, отчет по лабораторной работе
1.3.	Основные структурные элементы языка Java	2			4			Опрос, отчет по лабораторной работе
1.4.	Передача и хранение данных в Java-программах	4						Опрос.
2	Построение графического интерфейса пользователя							
2.1.	Основы создания графического интерфейса пользователя	2			4			Опрос, отчет по лабораторной работе
2.2.	Компоненты графического интерфейса	4			4			Опрос, отчет по лабораторной работе
2.3.	Создание таблиц	2			8			Опрос, отчет по лабораторной работе, дискуссия
2.4.	Принципы построения графических изображений	2			4			Опрос, отчет по лабораторной работе
2.5	Рисование геометрических фигур и текстовой информации	2			4		4	Опрос, отчет по лабораторной работе, дискуссия
3	Разработка многопоточных приложений							
3.1	Создание многопоточных приложений	2			8			Опрос. Отчет по лабораторной работе

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Перечень основной литературы

1. Дигрис, А.В., Прикладное программирование: пособие / А.В. Дигрис. – Минск : БГУ, 2019. – 196 с.
2. Корнелл, Г. Java. Библиотека профессионала: в 2т. / Г. Корнелл, К. Хорстманн. М.: Вильямс. – Т. 1: Основы, 2016. – 864 с.
3. Корнелл, Г. Java. Библиотека профессионала: в 2т. / Г. Корнелл, К. Хорстманн. М.: Вильямс. – Т. 2: Расширенные средства программирования, 2017. – 976 с.
4. Портянкин, И.А., Swing. Эффективные пользовательские интерфейсы / И.А. Портянкин. – Лори, 2011. – 608 с.

Перечень дополнительной литературы

1. Шилдт, Г. Swing руководство для начинающих / Г. Шилдт. – Вильямс, 2007. – 704 с.
2. Машнин, Т. Современные Java-технологии на практике / Т. Машнин. – БХВ-Петербург, 2010. – 560 с.
3. Bloch, J. Effective Java / J. Bloch. – Addison-Wesley, 2018. – 412 с.
4. Naftalin, M. Java Generics and Collections / M. Naftalin, P. Wadler. – O'Reilly Media, 2006. – 273 с.
5. Oaks, S. Java Threads. / S. Oaks, H. Wong. – O'Reilly Media, 2004. – 362 с.

Перечень рекомендуемых средств диагностики и методика формирования итоговой оценки

Для диагностики компетенций учащихся рекомендуется использовать устную форму контроля знаний.

Оценка за ответы по материалам лекций (опрос) может включать в себя полноту ответа, наличие аргументов, примеров из практики и т.д.

При оценивании лабораторных работ принимается во внимание правильность полученных результатов, способность учащегося обосновать и детально пояснить полученные результаты, а также продемонстрировать понимание программного кода выполненной работы и использованных технологий программирования.

Формой текущей аттестации по дисциплине «прикладное программирование» учебным планом предусмотрен зачет.

При формировании итоговой оценки используется оценка текущей успеваемости студента, дающая возможность проследить и оценить динамику процесса достижения целей обучения. Оценка текущей успеваемости предусматривает использование весовых коэффициентов для форм текущего контроля знаний студентов по дисциплине.

Формирование оценки за текущую успеваемость:

- опрос по материалам лекций – 20 %;

– отчет по лабораторной работе – 80 %.

Решение о допуске к сдаче зачета принимается на основе полученной оценки текущей успеваемости. К зачету допускаются студенты, получившие оценку текущей успеваемости не ниже 4 баллов.

Примерный перечень заданий для управляемой самостоятельной работы студентов

Тема 1. Использование средств библиотеки Java 2D при создании графических компонентов с расширенными возможностями. (4 часа)

Создание графических изображений с использованием комбинаций аффинных преобразований. Разработка компонент пользовательского интерфейса с возможностью масштабирования изображения. Построение настраиваемых компонент пользовательского интерфейса.

Примерная тематика лабораторных занятий

Лабораторная работа № 1. Создание консольного приложения, использующего разработанную иерархию классов. (8 часов).

Лабораторная работа № 2. Создание графического пользовательского интерфейса на основе стандартных компонент (8 часов).

Лабораторная работа № 3. Совершенствование интерфейса пользователя с применением модели и средств отображения данных в таблицах (8 часов).

Лабораторная работа № 4. Создание графических изображений собственных компонент (8 часов).

Лабораторная работа № 5. Разработка многопоточных приложений (8 часов).

Описание инновационных подходов и методов к преподаванию учебной дисциплины (эвристический, проективный, практико-ориентированный)

При организации образовательного процесса используется *метод учебной дискуссии*, который предполагает участие студентов в целенаправленном обмене мнениями, идеями для предъявления и/или согласования существующих позиций по определенной проблеме.

Использование метода обеспечивает появление нового уровня понимания изучаемой темы, применение знаний (теорий, концепций) при решении проблем, определение способов их решения.

При организации образовательного процесса *используются методы и приемы развития критического мышления*, которые представляют собой систему, формирующую навыки работы с информацией в процессе чтения, понимания информации как отправного, а не конечного пункта критического мышления.

Кроме того, при организации образовательного процесса *используется метод проектного обучения*, который предполагает организацию учебной

деятельности студентов, направленной на развитие навыков планирования, самоорганизации, сотрудничества и предполагающий создание собственного программного продукта.

Использование метода проектного обучения обеспечивает практическое применение полученных знаний и позволяет приобрести опыт коллективной работы, необходимый для молодых специалистов.

Методические рекомендации по организации самостоятельной работы обучающихся

При изучении учебной дисциплины рекомендуется использовать следующие формы самостоятельной работы:

- поиск (подбор) и обзор литературы и электронных источников по индивидуально заданной проблеме курса;
- работы, предусматривающие решение задач и выполнение упражнений, выдаваемых на лабораторных занятиях;
- подготовка к лабораторным занятиям с использованием размещенных в сетевом доступе учебных и учебно-методических материалов (программа курса, электронный вариант учебного пособия, методические указания, задания и информационные ресурсы для выполнения лабораторных работ, список рекомендуемой литературы и др.).

Примерный перечень вопросов к зачету

1. Платформа Java, её типы и особенности.
2. Основные принципы построения объектно-ориентированных программ на Java.
3. Структура программ на Java. Создание классов и их основные элементы. Понятие объектов класса, их создание и работа с их содержимым.
4. Организация иерархии классов в Java. Понятие интерфейса. Работа с содержимым суперклассов.
5. Организация взаимного использования классов. Множественное наследование и композиция. Создание встроенных классов и их разновидности.
6. Синтаксис программ на Java. Кодировка для написания программ. Организация комментариев и их виды.
7. Идентификаторы, ключевые слова в программах на Java. Литералы и их разновидности.
8. Арифметические, битовые и логические операторы в Java.
9. Операторы языка Java для организации ветвлений и циклов.
10. Простые типы данных в Java. Преобразование типов. Классы-оболочки для простых типов данных.
11. Представление строк в Java. Классы String и StringBuffer.
12. Класс Object и его основные возможности.
13. Обработка и генерация исключений в Java. Основные типы исключений и их классы. Создание собственных исключений.

14. Организация потоков данных в Java. Потоки байт и символов. Основные классы для создания потоков.
15. Создание массивов в Java и работа с их элементами. Использование класса `java.util.Arrays` для работы с массивами.
16. Коллекции в Java и их разновидности. Базовые классы и интерфейсы для создания классов коллекций.
17. Иерархия классов для создания списков. Виды списков и их отличительные особенности.
18. Классы для создания стеков и очередей.
19. Классы для создания множеств.
20. Создание и особенности использования карт отображений. Интерфейс `Map` и класс `AbstractMap`.
21. Библиотеки для создания графического пользовательского интерфейса в программах на Java. Создание и разновидности окон верхнего уровня.
22. Организация внутреннего содержания окон в `Swing`. Корневая панель, ее структура и назначение.
23. Класс `JPanel` как базовый контейнер для размещения интерфейсных компонент. Задание цвета фона и рамки для панелей.
24. Управление расположением компонент пользовательского интерфейса. Понятие менеджера компоновки.
25. Граничная компоновка. Класс `java.awt.BorderLayout`.
26. Плавающая компоновка. Класс `java.awt.FlowLayout`.
27. Табличная компоновка. Класс `java.awt.GridLayout`.
28. Создание стандартных диалоговых окон в программах на Java. Возможности класса `JOptionPane`.
29. Создание диалоговых окон для работы с файлами (класс `JFileChooser`) и для выбора цвета (класс `JColorChooser`).
30. Создание меток и текстовых полей средствами библиотеки `Swing`. Организация контроля содержимого текстового поля.
31. Создание кнопок, радиокнопок, флажков и комбинированных списков средствами библиотеки `Swing`.
32. Организация основного меню приложения. Клавиатурные операции для работы с пунктами меню.
33. Модель делегирования обработки событий и связанные с ней основные понятия. Цикл обработки событий.
34. Классы событий `ActionEvent` и `ItemEvent`, их назначение и обработка событий данных типов.
35. Представление и обработка событий клавиатуры.
36. Представление и обработка событий мыши.
37. Классы событий `MouseEvent` и `WindowEvent`, их назначение и обработка событий данных типов.
38. Создание таблиц с использованием библиотеки `Swing`. Модель таблицы и ее назначение.
39. Визуализация данных таблицы. Создание визуализаторов на основе интерфейса `TableCellRenderer`. Порядок использования визуализаторов.

40. Редактирование данных таблицы. Создание редакторов на основе интерфейса TableCellEditor. Порядок использования редакторов.
41. Управление порядком отображения данных в таблице, сортировка и фильтрация данных.
42. Организация перерисовки содержимого контейнера верхнего уровня в Swing. Роль и принцип работы менеджера перерисовки.
43. Создание изображений компонент пользовательского интерфейса. Классы Graphics и Graphics2D.
44. Рисование границ геометрических фигур. Основные возможности класса BasicStroke.
45. Задание цвета при рисовании геометрических фигур. Классы Color и SystemColor.
46. Организация градиентных цветовых схем при рисовании геометрических фигур. Классы GradientPaint, LinearGradientPaint и RadialGradientPaint.
47. Организация цветовых схем на основе текстур. Класс TexturePaint.
48. Рисование линий средствами библиотеки Java 2D. Классы Line2D, QuadCurve2D, CubicCurve2D, GeneralPath.
49. Рисование дуг, эллипсов и окружностей средствами библиотеки Java 2D. Классы Arc2D и Ellipse2D.
50. Рисование прямоугольников средствами библиотеки Java 2D. Классы Rectangle2D и RoundRectangle2D.
51. Создание сложных геометрических фигур на основе класса Area.
52. Рисование текстовой информации. Определение расположения текста. Задание шрифта и его параметров.
53. Понятие аффинного преобразования. Создание объектов преобразований с использованием класса AffineTransform и порядок их использования при рисовании.
54. Стандартные аффинного преобразования, их классы и способы их задания.
55. Понятие потока выполнения. Главный и дочерние потоки. Использование класса Thread для получения информации о характеристиках потока выполнения.
56. Группы потоков выполнения и их назначение. Класс ThreadGroup.
57. Способы написания собственного программного кода, выполняющегося в отдельном потоке.
58. Жизненный цикл потока выполнения. Прерывание выполнения потока. Организация взаимодействия потоков выполнения.
59. Проблема возникновения состязания потоков в многопоточном приложении. Синхронизация потоков на основе использования мониторов.

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Согласование с другими дисциплинами не требуется			

**ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ ПО
ИЗУЧАЕМОЙ УЧЕБНОЙ ДИСЦИПЛИНЕ**

на ____ / ____ учебный год

№ п/п	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры
_____ (протокол № ____ от _____ 202_ г.)

Заведующий кафедрой

УТВЕРЖДАЮ
Декан факультета
