|  |  |  |
| --- | --- | --- |
| C:\Users\e.musatkina\Desktop\HPSCANS\логисты заочка 2019\scan_20200924134732.jpg | МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  РОССИЙСКОЙ ФЕДЕРАЦИИ | |
| Autogenerated |
|  |  |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  «Магнитогорский государственный технический университет им. Г.И. Носова» | |
|  |
|  |  |  |
| УТВЕРЖДАЮ  Директор ИГДиТ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.Е. Гавришев  25.02.2020 г. | | |
|  |  |  |
| **РАБОЧАЯ** **ПРОГРАММА** **ДИСЦИПЛИНЫ** **(МОДУЛЯ)** | | |
|  |  |  |
| ***ОСНОВЫ*** ***ИНФОРМАТИКИ*** ***В*** ***ТРАНСПОРТНЫХ*** ***ПРИЛОЖЕНИЯХ*** | | |
|  |  |  |
| Направление подготовки (специальность)  38.03.02 МЕНЕДЖМЕНТ | | |
| Направленность (профиль/специализация) программы  Логистика | | |
|  |  |  |
| Уровень высшего образования - бакалавриат | | |
| Программа подготовки - прикладной бакалавриат | | |
|  |  |  |
| Форма обучения  заочная | | |
|  |  |  |
| Институт/ факультет | | Институт горного дела и транспорта |
|  |  |  |
| Кафедра | | Логистика и управление транспортными системами |
|  |  |  |
| Курс | | 2 |
|  |  |  |
| Магнитогорск  2019 год | | |

|  |
| --- |
| C:\Users\e.musatkina\Desktop\HPSCANS\логисты заочка 2019\scan_20200924134733.jpgРабочая программа составлена на основе ФГОС ВО по направлению подготовки 38.03.02 МЕНЕДЖМЕНТ (уровень бакалавриата) (приказ Минобрнауки России от 12.01.2016 г. № 7) |
|  |
| Рабочая программа рассмотрена и одобрена на заседании кафедры Логистика и управление транспортными системами  22.01.2020, протокол № 5 |
| Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.Н. Корнилов |
|  |
| Рабочая программа одобрена методической комиссией ИГДиТ  25.02.2020 г. протокол № 7 |
| Председатель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.Е. Гавришев |
|  |
| Рабочая программа составлена: |
| доцент кафедры ЛиУТС, канд. техн. наук \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_П.Н.Мишкуров |
|  |
| Рецензент: |
| Ведущий инженер-технолог ПГТ УЛ ПАО "ММК" , \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Е.В. Полежаев |

|  |  |
| --- | --- |
| **C:\Users\e.musatkina\Desktop\Лист актуализации 2019.jpgЛист** **актуализации** **рабочей** **программы** | |
|  |  |
|  | |
|  |  |
|  | |
|  |  |
| Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2020 - 2021 учебном году на заседании кафедры Логистика и управление транспортными системами | |
|  |  |
|  | Протокол от \_\_ \_\_\_\_\_\_\_\_\_\_ 20\_\_ г. № \_\_  Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.Н. Корнилов |
|  |  |
|  | |
|  |  |
|  | |
|  |  |
| Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2021 - 2022 учебном году на заседании кафедры Логистика и управление транспортными системами | |
|  |  |
|  | Протокол от \_\_ \_\_\_\_\_\_\_\_\_\_ 20\_\_ г. № \_\_  Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.Н. Корнилов |
|  |  |
|  | |
|  |  |
|  | |
|  |  |
| Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2022 - 2023 учебном году на заседании кафедры Логистика и управление транспортными системами | |
|  |  |
|  | Протокол от \_\_ \_\_\_\_\_\_\_\_\_\_ 20\_\_ г. № \_\_  Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.Н. Корнилов |
|  |  |
|  | |
|  |  |
|  | |
|  |  |
| Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2023 - 2024 учебном году на заседании кафедры Логистика и управление транспортными системами | |
|  |  |
|  | Протокол от \_\_ \_\_\_\_\_\_\_\_\_\_ 20\_\_ г. № \_\_  Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.Н. Корнилов |
|  |  |
|  | |
|  |  |
|  | |
|  |  |
| Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2024 - 2025 учебном году на заседании кафедры Логистика и управление транспортными системами | |
|  |  |
|  | Протокол от \_\_ \_\_\_\_\_\_\_\_\_\_ 20\_\_ г. № \_\_  Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.Н. Корнилов |

|  |  |
| --- | --- |
| **1** **Цели** **освоения** **дисциплины** **(модуля)** | |
| Целями освоения дисциплины (модуля) является развитие у студентов личностных качеств, а также формирование общекультурных и профессиональных компетенций в об-ласти информатики для решения теоретических и практических задач по вопросам повы-шения эффективности функционирования производственных и транспортных систем на основе использования транспортных приложений. | |
|  |  |
| **2** **Место** **дисциплины** **(модуля)** **в** **структуре** **образовательной** **программы** | |
| Дисциплина Основы информатики в транспортных приложениях входит в вариативную часть учебного плана образовательной программы.  Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик: | |
| Информатика | |
| Основы логистики и управление цепями поставок | |
| Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик: | |
| Основы баз данных | |
| Системный анализ в логистике | |
| Информационные системы в логистике | |
| Имитационное моделирование транспортных систем | |
| Агентное моделирование транспортных систем | |
| Разработка веб-сайтов | |
| Современные интернет-технологии | |
|  |  |
| **3** **Компетенции** **обучающегося,** **формируемые** **в** **результате** **освоения**  **дисциплины** **(модуля)** **и** **планируемые** **результаты** **обучения** | |
| В результате освоения дисциплины (модуля) «Основы информатики в транспортных приложениях» обучающийся должен обладать следующими компетенциями: | |
| Структурный  элемент  компетенции | Планируемые результаты обучения |
| ПК-10 владением навыками количественного и качественного анализа информации при принятии управленческих решений, построения экономических, финансовых и организационно-управленческих моделей путем их адаптации к конкретным задачам управления | |
| Знать | * виды и типы языков программирования, их характеристики; * технологические характеристики основных языков программирования; * особенности использования различных языков программирования в современных системах управления на транспорте. |
| Уметь | * структурировать значительные объемы статистической информации об объекте реального мира; * создавать программы на основании обработки информационных процессов; * создавать и использовать транспортные приложения |

|  |  |
| --- | --- |
| Владеть |  способами системами сбора, обработки и хранения информации;   методами обобщения, разделения и анализа данных о транспортном объекте;   методиками программирования процесса управления перевозочным процессом. |
| ПК-11 владением навыками анализа информации о функционировании системы внутреннего документооборота организации, ведения баз данных по различным показателям и формирования информационного обеспечения участников организационных проектов | |
| Знать | * показатели и характеристики информационных потоков в транспортных системах; * основные инструменты написания программ; * основы языка программирования -java; |
| Уметь | * выделять требуемые характеристики объекта для написания программного кода; * структурировать программный код при формировании транспортного приложения; * адаптировать разработанные программы к условия функционирования транспортного предприятия; |
| Владеть | * методами обработки данных о программируемом объекте; * методами описания информационных процессов на алгоритмическом языке; * программными инструментами цифровизации объектов транспорта . |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **4.** **Структура,** **объём** **и** **содержание** **дисциплины** **(модуля)** | | | | | | | | |
| Общая трудоемкость дисциплины составляет 3 зачетных единиц 108 акад. часов, в том числе:  – контактная работа – 6,4 акад. часов:  – аудиторная – 6 акад. часов;  – внеаудиторная – 0,4 акад. часов  – самостоятельная работа – 97,7 акад. часов;  – подготовка к зачёту – 3,9 акад. часа  Форма аттестации - зачет | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
| Раздел/ тема  дисциплины | | Курс | Аудиторная  контактная работа  (в акад. часах) | | | Самостоятельная работа студента | Вид самостоятельной  работы | Форма текущего контроля успеваемости и  промежуточной аттестации | Код компетенции |
| Лек. | лаб.  зан. | практ. зан. |
| 1. Раздел «Программное обеспечение и вычислительной техники и его классификация» | | |  | | | | | | |
| 1.1 «Основные понятия теории информации» | | 2 | 0,1 |  | 0,3 | 5,3 | Поиск дополнительной информации по заданной теме | Устный опрос | ПК-10, ПК-11 |
| 1.2 Тема «Тенденции развития программного обеспечения для ЭВМ и систем коммуникации» | | 0,2 |  | 0,3 | 5,3 | Поиск дополнительной информации по заданной теме | Устный опрос | ПК-10, ПК-11 |
| 1.3 «Классификация базового и прикладного программного обеспечения» | | 0,2 |  | 0,3 | 5,3 | Поиск дополнительной информации по заданной теме | Устный опрос | ПК-10, ПК-11 |
| 1.4 «Примеры использования программного обеспечения в различных отраслях народного хозяйства» | | 0,1 |  | 0,2 | 5,3 | Самостоятельное изучение учебной и научной литературы | Устный опрос | ПК-10, ПК-11 |
| Итого по разделу | | | 0,6 |  | 1,1 | 21,2 |  |  |  |
| 2.Раздел «Объектно-ориентированный подход к компьютерному программированию» | | |  | | | | | | |
| 2.1 «Основы объектно-ориентированный подхода к компьютерному программированию» | | 2 | 0,1 |  | 0,2 | 5,3 | Самостоятельное изучение учебной и научной литературы | Устный опрос | ПК-10, ПК-11 |
| 2.2 «Типы данных Java и оператор if» | | 0,1 |  | 0,2 | 5,3 | Подготовка к практическим занятиям, выполнение практических работ | Проверка практических заданий, устный опрос. | ПК-10, ПК-11 |
| 2.3 «Массивы» | | 0,1 |  | 0,2 | 5,3 | Подготовка к практическим занятиям, выполнение практических работ | Проверка практических заданий, устный опрос | ПК-10, ПК-11 |
| 2.4 «Циклы» | | 0,1 |  | 0,2 | 5,3 | Подготовка к практическим занятиям, выполнение практических работ | Проверка практических заданий, устный опрос. | ПК-10, ПК-11 |
| 2.5 «Коллекции» | | 0,1 |  | 0,2 | 5,3 | Подготовка к практическим занятиям, выполнение практических работ | Проверка практических заданий, устный опрос. | ПК-10, ПК-11 |
| 2.6 «Операторы сравнения, логические операторы и switch» | | 0,1 |  | 0,2 | 5,3 | Подготовка к практическим занятиям, выполнение практических работ | Подготовка к практическим занятиям, выполнение практических работ | ПК-10, ПК-11 |
| 2.7 «Видимость и связи» | | 0,1 |  | 0,2 | 5,3 | Подготовка к практическим занятиям, выполнение практических работ | Проверка практических заданий, устный опрос. | ПК-10, ПК-11 |
| 2.8 «Взаимодействие объектов в AnyLogic» | | 0,1 |  | 0,2 | 5,3 | Подготовка к практическим занятиям, выполнение практических работ | Проверка практических заданий, устный опрос. | ПК-10, ПК-11 |
| 2.9 «Диаграммы состояний» | | 0,1 |  | 0,2 | 5,3 | Подготовка к практическим занятиям, выполнение практических работ | Проверка практических заданий, устный опрос. | ПК-10, ПК-11 |
| Итого по разделу | | | 0,9 |  | 1,8 | 47,7 |  |  |  |
| 3. Раздел «Основные понятие метода имитационного моделирования» | | |  | | | | | | |
| 3.1 «Основные понятие метода имитационного моделирования» | | 2 | 0,1 |  | 0,2 | 5,3 | Поиск дополнительной информации по заданной теме | Устный опрос | ПК-10, ПК-11 |
| 3.2 «Системно-динамический подход к построению имитационных моделей» | | 0,1 |  | 0,3 | 5,3 | Поиск дополнительной информации по заданной теме | Устный опрос | ПК-10, ПК-11 |
| 3.3 «Дискретно-событийный и агентный подходы к построению имитационных моделей» | | 0,1 |  | 0,2 | 5,4 | Поиск дополнительной информации по заданной теме | Устный опрос | ПК-10, ПК-11 |
| 3.4 «Особенности построения имитационных моделей транспортных систем» | | 0,1 |  | 0,2 | 6,4 | Самостоятельное изучение учебной и научной литературы | Устный опрос | ПК-10, ПК-11 |
| 3.5 «Универсальные системы построения имитационных моделей. Инструмент имитационного моделирования AnyLogic» | | 0,1 |  | 0,2 | 6,4 | Поиск дополнительной информации по заданной теме | Устный опрос | ПК-10, ПК-11 |
| Итого по разделу | | | 0,5 |  | 1,1 | 28,8 |  |  |  |
| Итого за семестр | | | 2 |  | 4 | 97,7 |  | зачёт |  |
| Итого по дисциплине | | | 2 |  | 4 | 97,7 |  | зачет | ПК-10,ПК-11 |

|  |
| --- |
| **5** **Образовательные** **технологии** |
|  |
| Для реализации предусмотренных видов учебной работы в качестве образовательных технологий в преподавании дисциплины «Основы информатики в транспортных приложениях» используются традиционные интерактивная и модульно-компетентностная технологии.  В ходе проведения лекционных и практических занятий предусматривается:  - использование электронного демонстрационного материала по темам, требующим иллюстрации работы специализированного программного обеспечения, сложных структурных схем и большого объема графического материала;  -активные и интерактивные формы обучения: вариативный опрос, дискуссии, устный опрос, разбор конкретных ситуаций и т.д.  Образовательные технологии в сочетании с внеаудиторной работой нацелены на формирование и развитие профессиональных навыков обучающихся.  Самостоятельная работа стимулирует студентов в процессе подготовки домашних заданий, при подготовке к итоговой аттестации, которая осуществляется в форме защиты подготовленных рефератов. |
|  |
| **6** **Учебно-методическое** **обеспечение** **самостоятельной** **работы** **обучающихся** |
| Представлено в приложении 1. |
|  |
| **7** **Оценочные** **средства** **для** **проведения** **промежуточной** **аттестации** |
| Представлены в приложении 2. |
|  |
| **8** **Учебно-методическое** **и** **информационное** **обеспечение** **дисциплины** **(модуля)** |
| **а)** **Основная** **литература:** |
| 1. Боброва, И. И. Информатика : учебное пособие / И. И. Боброва ; МГТУ. - Магнитогорск : МГТУ, 2016. - 1 электрон. опт. диск (CD-ROM). - Загл. с титул. экрана. - URL: <https://magtu.informsystema.ru/uploader/fileUpload?name=2469.pdf&show=dcatalogues/1/1130212/2469.pdf&view=true> (дата обращения: 04.10.2019). - Макрообъект. - Текст : электронный. - Сведения доступны также на CD-ROM.  2. Лактионова, Ю. С. Информатика : учебное пособие / Ю. С. Лактионова, Л. С. Брябрина ; МГТУ. - Магнитогорск : МГТУ, 2015. - 1 электрон. опт. диск (CD-ROM). - Загл. с титул. экрана. - URL: <https://magtu.informsystema.ru/uploader/fileUpload?name=1507.pdf&show=dcatalogues/1/1124041/1507.pdf&view=true> (дата обращения: 04.10.2019). - Макрообъект. - Текст : электронный. - Сведения доступны также на CD-ROM. |
|  |
| **б)** **Дополнительная** **литература:** |
| 1. Романов, Е. П. Электронно-вычислительная техника и программирование : учебно-методическое пособие / Е. П. Романов ; МГТУ. - Магнитогорск : МГТУ, 2017. - 1 электрон. опт. диск (CD-ROM). - Загл. с титул. экрана. - URL: <https://magtu.informsystema.ru/uploader/fileUpload?name=3068.pdf&show=dcatalogues/1/1135237/3068.pdf&view=true> (дата обращения: 04.10.2019). - Макрообъект. - Текст : электронный. - Сведения доступны также на CD-ROM.  2. Самарина, И. Г. Программирование и основы алгоритмизации : учебное пособие. Ч. 1. Курс лекций / И. Г. Самарина. - Магнитогорск : МГТУ, 2013. - 1 электрон. опт. диск (CD-ROM). - Загл. с титул. экрана. - URL: <https://magtu.informsystema.ru/uploader/fileUpload?name=908.pdf&show=dcatalogues/1/1118881/908.pdf&view=true> |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| (дата обращения: 04.10.2019). - Макрообъект. - Текст : электронный. - Сведения доступны также на CD-ROM. | | | | | |
|  |  |  |  |  | |
| **в)** **Методические** **указания:** | | | | | |
| 1. Торшина, О. А. Объектно-ориентированное программирование : учебное пособие / О. А. Торшина ; МГТУ. - Магнитогорск : МГТУ, 2018. - 1 электрон. опт. диск (CD-ROM). - Загл. с титул. экрана. - URL: <https://magtu.informsystema.ru/uploader/fileUpload?name=3613.pdf&show=dcatalogues/1/1524595/3613.pdf&view=true> (дата обращения: 04.10.2019). - Макрообъект. - Текст : электронный. - ISBN 978-5-9967-1132-1. - Сведения доступны также на CD-ROM. | | | | | |
|  |  |  |  |  | |
| **г)** **Программное** **обеспечение** **и** **Интернет-ресурсы:** | | | | | |
|  | | | | | |
|
|  |  |  |  |  | |
| **Программное** **обеспечение** | | | | | |
|  | Наименование ПО | № договора | Срок действия лицензии |  | |
|  | MS Windows 7 Professional(для классов) | Д-1227-18 от 08.10.2018 | 11.10.2021 |  | |
|  | MS Windows 7 Professional (для классов) | Д-757-17 от 27.06.2017 | 27.07.2018 |  | |
|  | MS Office 2007 Professional | № 135 от 17.09.2007 | бессрочно |  | |
|  | 7Zip | свободно распространяемое ПО | бессрочно |  | |
|  | FAR Manager | свободно распространяемое ПО | бессрочно |  | |
|  |  |  |  |  | |
| **Профессиональные** **базы** **данных** **и** **информационные** **справочные** **системы** | | | | | |
|  | Название курса | | Ссылка |  | |
|  | Национальная информационно-аналитическая система – Российский индекс научного цитирования (РИНЦ) | | URL: https://elibrary.ru/project\_risc.asp |  | |
|  |  | |
|  | Поисковая система Академия Google (Google Scholar) | | URL: https://scholar.google.ru/ |  | |
| **9** **Материально-техническое** **обеспечение** **дисциплины** **(модуля)** | | | | | |
|  |  |  |  |  | |
| Материально-техническое обеспечение дисциплины включает: | | | | | |
| Учебные аудитории для проведения занятий лекционного типа: мультимедийные средства хранения, передачи и представления информации.  Учебные аудитории для проведения практических занятий, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации: мультимедийные средства хранения, передачи и представления информации.  Помещения для самостоятельной работы обучающихся: персональные компьютеры с пакетом MS Office, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета.  Помещения для хранения и профилактического обслуживания учебного оборудования: стеллажи для хранения учебно-методической документации, учебного оборудования и учебно-наглядных пособий. | | | | |

**Приложение 1**

**Учебно-методическое обеспечение самостоятельной работы обучающихся**

По дисциплине «Основы информатики в транспортных приложениях» предусмотрена аудиторная и внеаудиторная самостоятельная работа обучающихся.

Аудиторная самостоятельная работа студентов на практических занятиях осуществляется под контролем преподавателя в виде решения задач и выполнения упражнений, которые определяет преподаватель для студента.

Внеаудиторная самостоятельная работа обучающихся осуществляется в виде изучения литературы по соответствующему разделу с проработкой материала; выполнение практических заданий.

Перечень практических заданий:

**Практическое задание** **№1** на тему «Типы данных»

Общие правила приведения типов переменных Java в выражениях:

• типы переменныхbyte, short и char повышаются до типа int;

• разрядность целевой переменной (выражения) соответствует максимальной разрядно-сти операнда (переменной выражения).

Пример корректного использования типа целевой переменной:

byte b = 45;

char c = 'c';

short s = 1005;

inti = 700000;

float f = 4.55f;

double d = 1.456;

double result = (f \* b) + (i / c) - (d \* s);

System.out.println("resultравен " + result);

**Практическое задание** **№2** на тему «Логические операторы»

Пример вложенного оператора if:

if(i == 10)

{

if(j < 20) a =b;

if(k > 100) c =d;

else a = c; // else относится к if(k > 100)

}

else a = d; // else относится к if(i == 10)

Конструкция if-else

intlargerNum;

intlowNum = 9;

inthighNum = 27;

if(lowNum<highNum) // если первое число меньше второго

{

largerNum = highNum;

} else{ // иначе

largerNum = lowNum;

}

Тернарный оператор

int lowNum = 9;

int highNum = 27;

int largerNum = lowNum<highNum ? highNum :lowNum;

int absval, val;

val = 5;

absval = val< 0 ? -val:val;

// выводим число

System.out.println("" + absval);

val = -5;

absval = val< 0 ? -val:val;

System.out.println("" + absval);

Пример использования оператора switch– программа для получения кода типа вагона.

Stringвагон = "полувагон";

intкод = 0;

switch (вагон) {

case"крытый":код = 2;

break;

case"платформа":код = 4;

break;

case"полувагон":код = 6;

break;

case"цистерна":код = 7;

break;

case"изотермический":код = 8;

break;

case"прочие":код = 9;

break;

default: System.out.println("Неверныйтипвагона");

break;

}

if (код != 0) System.out.println("Типвагона: " + вагон + ",код: " + код);

**Практическое задание** **№3** на тему «Массивы»

Пример одновременного создания и инициализации массива в AnyLogic (запись помещается в поле «Начальное значение» переменной)

newint[] { 13, x-3, -15, 0, max(a,100) };

или в программе наязыкеJava

int[] intarray = new int[] { 13, x-3, -15, 0, max( a, 100 ) };

Пример создания массива строковых переменных

String[] список = newString[] { “Света”, “Таня”, “Маша” };

Двумерный массив значений типа double

double[][] двумерныйМассив = newdouble[10][20];

intчислоСтрок = двумерныйМассив.length;

intчислоСтолбцов = двумерныйМассив[0].length;

System.out.println(“Двумерный массив содержит ”+ числоСтрок+ ” строк и ” + числоСтолбцов + ” столбцов”);

При инициализации многомерных массивов используют дополнительные фигурные скобки, например, в программе на языке Java

int[][] intarrayD = {

{ 1, 2, 3 },

{ 4, 5, 6 }

}

или при указании начального значения переменной-массива в AnyLogic

**newint**[][] {{ 1, 2, 3 },{ 4, 5, 6 }}

Данный массив будет содержать две строки, в каждой из которых находится три элемента. Представление созданного в предыдущем примере массива форме таблицы

|  |  |  |  |
| --- | --- | --- | --- |
| Индексы строк | Индексы столбцов | | |
| **0** | **1** | **2** |
| **0** | 1 | 2 | 3 |
| **1** | 4 | 5 | 6 |

Элемент массива intarrayD[1][2], например, будут равен 6.

**Практическое задание** **№4** на тему «Циклы»

Пример цикла **for** на языке Java:

String[] список = new String[] { “Света”, “Таня”, “Маша”, “Дима” };

intдлинаСписка = список.length;

intсчетчик;

for (счетчик = 0; счетчик<длинаСписка; счетчик++) {

String пассажир = список[счетчик];

int номер = счетчик+1;

System.out.println(“Пассажир №” + номер + “” + пассажир);

}

В результате выполнения цикла будут напечатано:

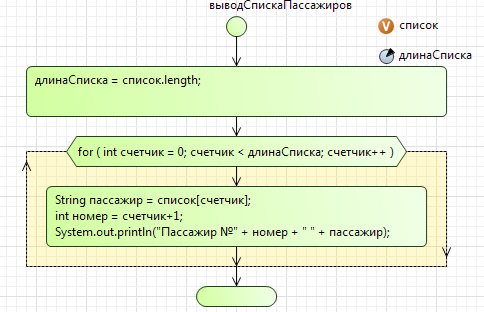
Пассажир №1 Света

Пассажир №2 Таня

Пассажир №3 Маша

Пассажир №4 Дима

На следующем рисунке представлен пример создания цикла **for** при помощи «диаграммы действий» AnyLogic



Цикл **while**

Проверка условия окончания цикла в цикле while происходит ДО начала выполнения цикла. Например, такой цикл выполнится четыре раза, а на экран будет выведено «1 2 3 4 »

inti = 1;

while (i< 5) {

System.out.println(i + " ");

i++;

}

При определённых условиях цикл while может ни разу не выполниться, например, поскольку значение i больше нуля, то тело данного цикла не выполнится и начего напечатано не будет

inti = 1;

while (i< 0) {

System.out.println(i + " ");

i++;

}

Цикл whileможет выполняться бесконечное число раз, например

int i = 1;

while (true) {

System.out.println(i + " ");

i++;

}

Цикл **do… while**

Условие окончания цикла в цикле**do … while** происходит ПОСЛЕ выполнения тела цикла, поэтому в цикле **do … while**операторы тела цикла выполняются как минимум один раз.

Например, такой цикл выполнится один раз, а на экран будет выведено «2 »

**int** i = 1;

**do** {

i++;

System.out.println(i + " ");

} **while** (i< 0);

Следующий цикл выполнится четыре раза, а на экран будет выведено «2 3 4 5 »

**int** i = 1;

**do** {

i++;

System.out.println(i + " ");

} **while** (i < 5);

**Практическое задание** **№5** на тему «Коллекции»

Создать коллекцию ArrayList, инициализировать ее тридцатью элементами. Идентифицировать десятый и двадцатый элемент коллекции. Добавить в коллекцию 5 элементов. Удалить первый элемент коллекции и заменить одиннадцатый.

Методы коллекции ArrayList:

* add(элемент)– добавление элемента в конец коллекции. Элемент может переменной любого типа Java, экземпляром класса Javaили агентом AnyLogic.
* get(индекс элемента) –получение элемента коллекции по его индексу. Нумерация индексов в коллекции начинается так же как и у массивов – с нуля.
* size() –получение числа элементов коллекции.
* indexOf(элемент) – получение индекса элемента в коллекции.
* contains(элемент)– определение наличия элемента коллекции. Если элемент присутствует, то результатом выполнения метода будут true, в противном случае – false.
* remove(индекс)илиremove(элемент) – удаление элемента из коллекции, соответственно, по его индексу или по содержимому элемента.
* set(индекс, элемент) – замена элемента в коллекции с индексом «индекс» на новый элемент, указанный в качестве параметра метода set.
* clear() – очистка коллекции.

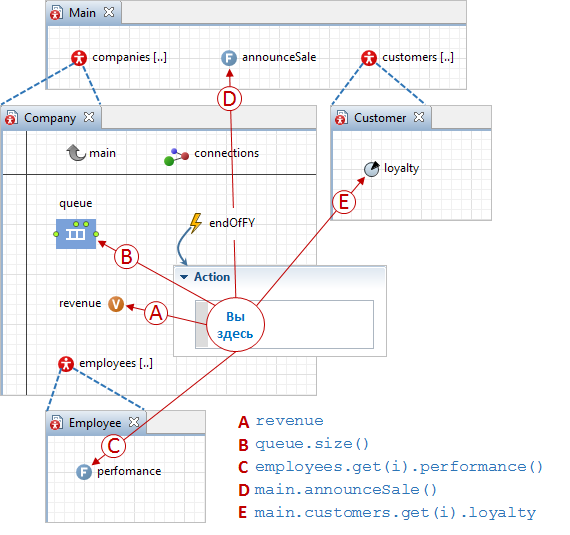
Создать коллекцию LinkedList, инициализировать ее тридцатью элементами. Идентифицировать десятый и двадцатый элемент коллекции. Добавить в коллекцию 5 элементов. Удалить первый элемент коллекции.

Дополнительные методы коллекции LinkedList:

* addLast(элемент) и addFirst(элемент) – добавление элемента, соответственно, в конец и начало списка.
* add(индекс, элемент) – добавление элемента в список ПЕРЕД элементом, с указанным индексом.
* removeLast()и removeFirst()–удаление элемента, находящегося, соответственно, в конце и начале списка.

**Практическое задание** **№6** на тему «Взаимодействие объектов»

Получить доступ к другим элементам модели.



Объявить модификаторы.

**public int** i;

**private double** j, k;

**privateint** createMethod(int a) {...};

**publicclass**Cat{}

**Приложение 2**

**Оценочные средства для проведения промежуточной аттестации**

**а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:**

| Структурный элемент  компетенции | Планируемые результаты обучения | Оценочные средства |
| --- | --- | --- |
| **ПК-10 владение навыками количественного и качественного анализа информации при принятии управленческих решений, построения экономических, финансовых и организационно-управленческих моделей путем их адаптации к конкретным задачам управления** | | |
| Знать | * виды и типы информации, их характеристики; * технологические характеристики основных транспортных процессов; * функции информационных потоков в современных системах управления на транспорте. | **Перечень теоретических вопросов:**  1.Понятие об имитационном моделировании. Назначение имитационных моделей.  2.Современные подходы к имитационному моделированию. Сущность системно-динамического, дискретно-событийного (процессного) и агентного подходов к имитационному моделированию.  3.Область применения системно-динамических имитационных моделей.  4.Область применения дискретно-событийных (процессных) имитационных моделей.  5.Область применения агентных имитационных моделей.  6.Программные системы построения имитационных моделей. Их достоинства, недостатки и области применения. Достоинства мультиподходной системы AnyLogic.  7.Особенности системы построения имитационных моделей AnyLogic. Рабочая область системы. Состав рабочей области программы. Палитры блоков.  8.Сущность объектно-ориентированного подхода к разработке компьютерных про-грамм.  9.Понятие «абстракция» в объектно-ориентированном программировании. Приме-ры.  10.Понятие «инкапсуляция» в объектно-ориентированном программировании. Примеры.  11.Понятие «наследование» в объектно-ориентированном программировании. Примеры.  12.Понятие «полиморфизм» в объектно-ориентированном программировании. Примеры. |
| Уметь | * обрабатывать значительные объемы статистической информации; * создавать информационные системы на основании обработки информационных процессов; * создавать и использовать системы управления транспортными процессами, работающими в режиме реального времени. | **Примерные практические задания:**  **Практическое задание** **№1** на тему «Типы данных»  Общие правила приведения типов переменных Java в выражениях:  • типы переменныхbyte, short и char повышаются до типа int;  • разрядность целевой переменной (выражения) соответствует максимальной разрядно-сти операнда (переменной выражения).  Пример корректного использования типа целевой переменной:  byte b = 45;  char c = 'c';  short s = 1005;  inti = 700000;  float f = 4.55f;  double d = 1.456;  double result = (f \* b) + (i / c) - (d \* s);  System.out.println("resultравен " + result);  **Практическое задание** **№2** на тему «Логические операторы»  Пример вложенного оператора if:  if(i == 10)  {  if(j < 20) a =b;  if(k > 100) c =d;  else a = c; // else относится к if(k > 100)  }  else a = d; // else относится к if(i == 10)  Конструкция if-else  intlargerNum;  intlowNum = 9;  inthighNum = 27;  if(lowNum<highNum) // если первое число меньше второго  {  largerNum = highNum;  } else{ // иначе  largerNum = lowNum;  }  Тернарный оператор  int lowNum = 9;  int highNum = 27;  int largerNum = lowNum<highNum ? highNum :lowNum;  int absval, val;  val = 5;  absval = val< 0 ? -val:val;  // выводим число  System.out.println("" + absval);  val = -5;  absval = val< 0 ? -val:val;  System.out.println("" + absval);  Пример использования оператора switch– программа для получения кода типа вагона.  Stringвагон = "полувагон";  intкод = 0;  switch (вагон) {  case"крытый":код = 2;  break;  case"платформа":код = 4;  break;  case"полувагон":код = 6;  break;  case"цистерна":код = 7;  break;  case"изотермический":код = 8;  break;  case"прочие":код = 9;  break;  default: System.out.println("Неверныйтипвагона");  break;  }  if (код != 0) System.out.println("Типвагона: " + вагон + ",код: " + код); |
| Владеть | * способами системами сбора, обработки и хранения информации; * методами обобщения, разделения и анализа информации; * методиками управления информационными потоками в процессе управления на транспорте. | **Перечень теоретических вопросов:**   1. Системно-динамический подход к построению имитационных моделей 2. Дискретно-событийный (процессный) подход к построению имитационных моделей 3. Агентный подход к построению имитационных моделей. |
| **ПК-11 владение навыками анализа информации о функционировании системы внутреннего документооборота организации, ведения баз данных по различным показателям и формирования информационного обеспечения участников организационных проектов** | | |
| Знать | * показатели и характеристики современных транспортных технологий; * оосновные системы управления, используемые в транспортном комплексе; * новейшие информационные технологии; | **Перечень теоретических вопросов:**   1. Особенности языка программирования Java как языка объектно-ориентированного программирования. Направления использования программного кода на языке Java в среде AnyLogic. 2. Типы данных в языке программирования Java. 3. Оператор if в языке программирования Java. Примеры использования оператора if. 4. Массивы в языке программирования Java. Примеры использования массивов. 5. Циклы в языке программирования Java. Виды циклов. Примеры использования циклов. 6. Коллекции в языке Java. Виды коллекций. Методы коллекций. Примеры использо-вания коллекций. 7. Операторы сравнения и логистические операторы Java. Примеры использования операторов сравнения и логических операторов. 8. Оператор switch в языке Java. Примеры использования оператора switch. 9. Правила видимости в языке Java и в системе AnyLogic. Примеры доступа к пере-менным и объектам я системе AnyLogic. 10. Организация связей между агентами и взаимодействие агентов в системе AnyLogic. Примеры организации связей. 11. Понятие «диаграммы состояний». Элементы диаграммы состояний. Порядок и примеры использования диаграмм состояний в системе AnyLogic. 12. Диаграммы действий в системе AnyLogic. Назначение, порядок и примеры исполь-зования диаграмм действий. 13. Системно-динамическое моделирование в AnyLogic. Особенности системно-динамических моделей. Примеры системно-динамических моделей. 14. Основы планирования и проведения экспериментов с имитационными моделями в системе AnyLogic. |
| Уметь | * выделять требуемые информационные потоки из общего информационного поля; * создавать сложные информационные системы управления на транспорте; * адаптировать разработанные информационные технологии к условия функционирования транспортного предприятия; | **Примерные практические задания:**  **Практическое задание** **№3** на тему «Массивы»  Пример одновременного создания и инициализации массива в AnyLogic (запись помещается в поле «Начальное значение» переменной)  newint[] { 13, x-3, -15, 0, max(a,100) };  или в программе наязыкеJava  int[] intarray = new int[] { 13, x-3, -15, 0, max( a, 100 ) };  Пример создания массива строковых переменных  String[] список = newString[] { “Света”, “Таня”, “Маша” };  Двумерный массив значений типа double  double[][] двумерныйМассив = newdouble[10][20];  intчислоСтрок = двумерныйМассив.length;  intчислоСтолбцов = двумерныйМассив[0].length;  System.out.println(“Двумерный массив содержит ”+ числоСтрок+ ” строк и ” + числоСтолбцов + ” столбцов”);  При инициализации многомерных массивов используют дополнительные фигурные скобки, например, в программе на языке Java  int[][] intarrayD = {  { 1, 2, 3 },  { 4, 5, 6 }  }  или при указании начального значения переменной-массива в AnyLogic  **newint**[][] {{ 1, 2, 3 },{ 4, 5, 6 }}  Данный массив будет содержать две строки, в каждой из которых находится три элемента. Представление созданного в предыдущем примере массива форме таблицы   |  |  |  |  | | --- | --- | --- | --- | | Индексы строк | Индексы столбцов | | | | **0** | **1** | **2** | | **0** | 1 | 2 | 3 | | **1** | 4 | 5 | 6 |   Элемент массива intarrayD[1][2], например, будут равен 6.  **Практическое задание** **№4** на тему «Циклы»  Пример цикла **for** на языке Java:  String[] список = new String[] { “Света”, “Таня”, “Маша”, “Дима” };  intдлинаСписка = список.length;  intсчетчик;  for (счетчик = 0; счетчик<длинаСписка; счетчик++) {  String пассажир = список[счетчик];  int номер = счетчик+1;  System.out.println(“Пассажир №” + номер + “” + пассажир);  }  В результате выполнения цикла будут напечатано:  Пассажир №1 Света  Пассажир №2 Таня  Пассажир №3 Маша  Пассажир №4 Дима  На следующем рисунке представлен пример создания цикла **for** при помощи «диаграммы действий» AnyLogic    Цикл **while**  Проверка условия окончания цикла в цикле while происходит ДО начала выполнения цикла. Например, такой цикл выполнится четыре раза, а на экран будет выведено «1 2 3 4 »  inti = 1;  while (i< 5) {  System.out.println(i + " ");  i++;  }  При определённых условиях цикл while может ни разу не выполниться, например, поскольку значение i больше нуля, то тело данного цикла не выполнится и начего напечатано не будет  inti = 1;  while (i< 0) {  System.out.println(i + " ");  i++;  }  Цикл whileможет выполняться бесконечное число раз, например  int i = 1;  while (true) {  System.out.println(i + " ");  i++;  }  Цикл **do… while**  Условие окончания цикла в цикле**do … while** происходит ПОСЛЕ выполнения тела цикла, поэтому в цикле **do … while**операторы тела цикла выполняются как минимум один раз.  Например, такой цикл выполнится один раз, а на экран будет выведено «2 »  **int** i = 1;  **do** {  i++;  System.out.println(i + " ");  } **while** (i< 0);  Следующий цикл выполнится четыре раза, а на экран будет выведено «2 3 4 5 »  **int** i = 1;  **do** {  i++;  System.out.println(i + " ");  } **while** (i < 5);  **Практическое задание** **№5** на тему «Коллекции»  Создать коллекцию ArrayList, инициализировать ее тридцатью элементами. Идентифицировать десятый и двадцатый элемент коллекции. Добавить в коллекцию 5 элементов. Удалить первый элемент коллекции и заменить одиннадцатый.  Методы коллекции ArrayList:   * add(элемент)– добавление элемента в конец коллекции. Элемент может переменной любого типа Java, экземпляром класса Javaили агентом AnyLogic. * get(индекс элемента) –получение элемента коллекции по его индексу. Нумерация индексов в коллекции начинается так же как и у массивов – с нуля. * size() –получение числа элементов коллекции. * indexOf(элемент) – получение индекса элемента в коллекции. * contains(элемент)– определение наличия элемента коллекции. Если элемент присутствует, то результатом выполнения метода будут true, в противном случае – false. * remove(индекс)илиremove(элемент) – удаление элемента из коллекции, соответственно, по его индексу или по содержимому элемента. * set(индекс, элемент) – замена элемента в коллекции с индексом «индекс» на новый элемент, указанный в качестве параметра метода set. * clear() – очистка коллекции.   Создать коллекцию LinkedList, инициализировать ее тридцатью элементами. Идентифицировать десятый и двадцатый элемент коллекции. Добавить в коллекцию 5 элементов. Удалить первый элемент коллекции.  Дополнительные методы коллекции LinkedList:   * addLast(элемент) и addFirst(элемент) – добавление элемента, соответственно, в конец и начало списка. * add(индекс, элемент) – добавление элемента в список ПЕРЕД элементом, с указанным индексом. * removeLast()и removeFirst()–удаление элемента, находящегося, соответственно, в конце и начале списка.   **Практическое задание** **№6** на тему «Взаимодействие объектов»  Получить доступ к другим элементам модели.    Объявить модификаторы.  **public int** i;  **private double** j, k;  **privateint** createMethod(int a) {...};  **publicclass**Cat{} |
| Владеть | * методами обработки информации; * методами описания информационных процессов на алгоритмическом языке; * инструментами аппарата управления на транспорте на новейших информационных технологий. * методами создания и использования СУБД; | **Перечень теоретических вопросов:**   1. Инструменты построения имитационных моделей. 2. Методы работы с коллекциями Java. 3. Методы работы с массивами данных. |

**б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:**

Промежуточная аттестация по дисциплине «Основы информатики в транспортных приложениях» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические и комплексные задания, выявляющие степень сформированности умений и владений, проводится в форме зачёта.

Зачет по данной дисциплине проводится в устной форме.

**Показатели и критерии оценивания зачета:**

– на оценку **«зачтено»** – обучающийся демонстрирует достаточный уровень сформированности компетенций, основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.

– на оценку **«не зачтено»** – обучающийся демонстрирует знания не более 20% теоретического материала, допускает существенные ошибки, не может показать интеллектуальные навыки решения простых задач.