МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет

имени академика С.П. Королева» (Самарский университет)

Институт информатики, математики и электроники Факультет информатики

Кафедра программных систем

ОТЧЕТ ПО ПРАКТИКЕ

Вид практики: учебная

Тип практики: Научно-исследовательская работа (получение первичных навыков научно-исследовательской работы)

Сроки прохождения практики: с 01.07.2023 г. по 14.07.2023 г. (в соответствии с календарным учебным графиком)

по направлению подготовки 02.03.02 «Фундаментальная информатика и информационные технологии» (уровень бакалавриата) направленность (профиль) «Информационные технологии»

Обучающийся группы № 6204-020302D Д.О. Колбанов

Руководитель практики

от университета, доцент кафедры

программных систем, к.т.н., доцент

Д.А. Попова-Коварцева

Дата сдачи 14.07.2023 г. Дата защиты 14.07.2023 г.

Оценка

Самара 2023

СОДЕРЖАНИЕ

[Задание (я) по практике для выполнения определенных видов работ, 3](#_bookmark0)

[связанных с будущей профессиональной деятельностью 3](#_bookmark1)

[(сбор и анализ данных и материалов, проведение исследований) 3](#_bookmark2)

[Введение 5](#_bookmark3)

1. [Часть 1. Обзор источников по теме «](#_bookmark4)Компьютерное зрение в автомобильной индустрии[». 6](#_bookmark4)
   1. [Источник 1 – Статья «The Role of Computer Vision in the Automotive Industry» 7](#_bookmark5)
   2. [Источник 2 – Статья «Перспективы применения искусственного интеллекта и компьютерного зрения в транспортных системах и подключенных автомобилях» 7](#_bookmark6)
   3. [Источник 3 – Статья «The Role of Computer Vision in the Automotive Industry»](#_bookmark7) 9
   4. [Источник 4 – Статья ««Машинное» зрение: что и как видят автомобили»](#_bookmark8) 10
2. [Часть 2. Формирование и преобразование бинарного дерева 1](#_bookmark9)2
   1. [Задание 1 – Формирование дерева (задача Tree 32) 1](#_bookmark10)3
      1. [Формулировка задания 1](#_bookmark11)3
      2. [Алгоритм решения 1](#_bookmark12)3
      3. [Визуализация работы алгоритма 1](#_bookmark13)3
   2. [Задание 2 – Преобразование дерева (задача Tree 43) 1](#_bookmark14)4
      1. [Формулировка задания 1](#_bookmark15)4
      2. [Алгоритм решения 1](#_bookmark16)4
      3. [Визуализация работы алгоритма 1](#_bookmark17)5

[Заключение 1](#_bookmark18)6

[Список использованных источников 1](#_bookmark19)7

[Приложение А Блок-схема алгоритма формирования дерева 1](#_bookmark21)8

[Приложение Б Листинг программы](#_bookmark22) 19

[Приложение В Блок-схема алгоритма преобразования дерева 2](#_bookmark23)1

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет

имени академика С.П. Королева» (Самарский университет)

Институт информатики, математики и электроники Факультет информатики

Кафедра программных систем

Задание (я) по практике для выполнения определенных видов работ, связанных с будущей профессиональной деятельностью

(сбор и анализ данных и материалов, проведение исследований)

Обучающемуся Колбанову Дмитрию Олеговичу группы 6204-020302D

Направлен на практику приказом по университету от 15.06.2023 г. № 289-ПР в Самарский университет на кафедру программных систем.

|  |  |  |
| --- | --- | --- |
| Планируемые результаты освоения образовательной программы *(компетенции)/индикаторы)* | Выполнение определенных видов работ, связанных с будущей профессиональной деятельностью (сбор и анализ данных и материалов, проведение исследований) | Результаты практики *(планируемые результаты обучения при прохождении практики)* |
| **ОПК-1 Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности**  **ОПК- 1.1** Использует основные положения и концепции в области математических и естественных наук, базовые теории коммуникации; знает основную терминологию  **ОПК-1.2** Осуществляет первичный сбор и анализ материала, интерпретирует различные математические объекты  **ОПК-1.3** Осуществляет первичный сбор и анализ материала, интерпретирует различные математические объекты | 1. Ознакомиться с концепциями и терминологией по теме обзора научно-исследовательской работы. 2. Осуществить поиск материала по теме обзора в сети Интернет, электронных библиотечных системах и базах данных. 3. Ознакомиться с рекомендациями по подготовке презентационного материала. 4. Подготовить презентацию по теме обзора в соответствии с общими правилами оформления презентаций. 5. Ознакомиться со стандартом оформления текстовых учебных документов Самарского университета. 6. Оформить отчет по результатам прохождения практики в строгом соответствии со стандартом оформления текстовых учебных документов. | 1. Ознакомлен с концепциями и терминологией по теме обзора научно-исследовательской работы. 2. Осуществлен поиск материала по теме обзора в сети Интернет, электронных библиотечных системах и базах данных. 3. Ознакомлен с рекомендациями по подготовке презентационного материала. 4. Подготовлена презентация по теме обзора в соответствии с общими правилами оформления презентаций. 5. Ознакомлен со стандартом оформления текстовых учебных документов Самарского университета. 6. Оформлен отчет по результатам прохождения практики в строгом соответствии со стандартом оформления текстовых учебных документов. |
| **ОПК-2 Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности**  **ОПК-2.1** Использует основные положения и концепции в области программирования, архитектуру языков программирования, теорию коммуникации, основную терминологию, знаком с содержанием единого реестра российских программ  **ОПК-2.2** Анализирует код на типовых языках программирования, может составлять программы  **ОПК-2.3** Применяет опыт решения задач анализа, интеграции различных типов программного обеспечения, анализа типов коммуникаций | 1. Разработать алгоритмы решения индивидуальных заданий по формированию и преобразованию бинарных деревьев. 2. Описать созданные алгоритмы. | 1. Разработаны алгоритмы решения индивидуальных заданий по формированию и преобразованию бинарных деревьев. 2. Описаны созданные алгоритмы. |
| **ОПК-3 Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям**  **ОПК-3.1** Понимает методы теории алгоритмов, методы системного и прикладного программирования, основные положения и концепции в области математических, информационных и имитационных моделей  **ОПК-3.2** Соотносит знания в области программирования, интерпретирует прочитанное, определяет и создает информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем  **ОПК-3.3** Имеет практический опыт применения разработки программного обеспечения | 1. С помощью выбранного языка программирования, создать программные коды, соответствующие разработанным алгоритмам создания и преобразования бинарных деревьев. 2. Реализовать визуализацию формирования и обработки бинарных деревьев в соответствии с вариантом задания. | 1) С помощью языка программирования Python созданы программные коды, соответствующие разработанным алгоритмам создания и преобразования бинарных деревьев.  2) Реализована визуализация формирования и обработки бинарных деревьев в соответствии с вариантом задания. |

Срок представления на кафедру отчета о практике 14.07.2023 г.

Руководитель практики от университета, доцент кафедры программных

систем, к.т.н., доцент Д.А. Попова-Коварцева

*(подпись)*

Задание принял к исполнению

обучающийся группы № 6204-020302D Д.О. Колбанов

*(подпись)*

ВВЕДЕНИЕ

Данная практика направлена на закрепление и углубление теоретической и практической подготовки по программированию, приобретение навыков самостоятельной и научно-исследовательской работы, а также оформления учебных текстовых документов.

В ходе выполнения практики необходимо составить обзор найденных источников по выбранной теме НИР «Компьютерное зрение в автомобильной индустрии». Затем подготовить презентацию по теме обзора.

Также предлагается решить задачи на формирование и преобразование дерева из задачника по бинарным деревьям [1]. Решение задачи может быть представлено в любой среде программирования и должно включать отрисовку дерева. Для решения задач была выбран язык программирования Python.

Отчёт по практике требуется оформить согласно стандарту оформления учебных текстовых документов Самарского университета [2].

1Часть 1. Обзор источников по теме «Компьютерное зрение в автомобильной индустрии».

Технологии машинного зрения развиваются на протяжении последних десятилетий. Ученые, занимающиеся этим вопросом, хотят научить компьютер видеть и анализировать полученную информацию.

У машинного зрения существует большое количество применений — такие технологии можно использовать, например, для получения 3D-моделей из фото и видеоматериалов в процессе создания цифровых карт, для создания архитектурных моделей зданий или в охранных системах для распознавания лиц. Но наибольшее влияние на повседневную жизнь миллионов людей окажет внедрение подходов машинного зрения в автомобилестроении.

Одно из наиболее очевидных применений технологий машинного зрения в автомобилях — это создание систем предупреждения аварийных ситуаций. Сейчас активно развивается технология Мобилай (Mobileye), которая используется в машинах BMW, General Motors, и, с недавнего времени, Tesla Motors. Также свою технологию City Safety разрабатывает шведский автопроизводитель Volvo.

Эта система отслеживает траекторию движения машины и предупреждает водителя, если он слишком близко приблизится к впереди едущему автомобилю, или в случае его резкого торможения. При этом возможно не только получение предупреждения, но и автоматическое торможение в критических ситуациях — за это отвечают автономные системы экстренного торможения (Autonomous Emergency Braking, AEB), которые используются и в беспилотных автомобилях.

Существуют и другие способы применения машинного зрения — например, компания Ford научила фары автомобиля реагировать на информацию, поступающую с фронтальной камеры. Электроника расширяет луч света фар при приближении к знаку перекрестка или кругового движения, чтобы водитель мог раньше увидеть объекты, движущиеся сбоку. Кроме того, система может распознавать и внезапно появляющиеся на дороге объекты, заранее освещая их — в General Motors уверены, что это помогает снизить вероятность наезда на препятствие.

Технологии машинного зрения также используются для создания систем навигации — с помощью камер и сенсоров машина может определять свое местоположение, сверяться с цифровой картой и прокладывать маршруты движения. Такой проект разрабатывают, к примеру, инженеры Mercedes-Benz (система Route Pilot).

Машинное зрение можно использовать не только для улучшения существующих автомобилей, но и при проектировании новых — к примеру, с помощью фото и видео можно строить 3D-модели создаваемой машины.

В свою очередь страховые компании могут использовать современные инструменты для фиксации повреждений автомобиля, который попал в ДТП. Кроме того, если в аварию попадут автомобили, оборудованные технологиями машинного зрения, впоследствии это может облегчить реконструкцию сцены происшествия.

Далее можно ознакомиться с выбранными по теме источниками.

* 1. Источник 1 – Статья «The Role of Computer Vision in the Automotive Industry»

Генеральный директор и соучредитель Shaip Ватсал Гийс в статье [3] поделился некоторыми соображениями о роли компьютерного зрения в автомобильной промышленности, концепции обучения интеллектуальных моделей, области применения компьютерного зрения в автомобильной промышленности и намного больше.

Начиная с решения по снижению трафика и заканчивая возможностью вождения без водителя, производители автомобилей широко внедряют ИИ как неотъемлемую часть своего бизнес-процесса. Чтобы сделать использование ИИ более широким в автомобильной промышленности, производители могут обучать эти интеллектуальные модели в соответствии со своими бизнес-требованиями и повышать эффективность процессов наряду с контролем качества.

Более того, для обучения этих моделей предприятия могут использовать модели компьютерного зрения для идентификации, классификации и концептуализации изображений с высокой точностью. Автопроизводители используют модели компьютерного зрения не только для создания качественного продукта, но и для ускорения обслуживания клиентов беспилотными автомобилями.

В автомобильной промышленности модели компьютерного зрения могут широко использоваться в таких процессах, как профилактическое обслуживание, создание автономного мониторинга, интеллектуальная помощь водителю, обнаружение дефектов продукции, автоматизированные производственные операции и многие другие. Дорога к успешной реализации может показаться далекой, но прямо сейчас ведется активная разработка.

* 1. Источник 2 – Статья «Перспективы применения искусственного интеллекта и компьютерного зрения в транспортных системах и подключенных автомобилях»

В статье [4] описаны технологии компьютерного зрения в автомобильной индустрии, а также основные их потребители.

Автор статьи Постолит А.В., доктор технических наук, профессор, директор компании Smart Information Systems, описывая технологии систем компьютерного зрения, по своему функциональному назначению делит их на следующие группы:

* оценка состояния водителя;
* оценка действий водителя;
* оценка окружающей обстановки.

Что касается оценки состояния водителя, то современные системы способны

решать следующие задачи:

* фиксация засыпания водителя и выдача сигналов для пробуждения;
* фиксация отвлечения от дороги (курение, разговор по телефону, еда, направление

взгляда не на дорогу, нехарактерная поза и т. п.);

* распознавание личности водителя, находящегося за рулём.

С использованием искусственного интеллекта и систем компьютерного зрения можно оценить следующие действия водителя:

* зафиксировать выезд за пределы дорожной разметки или пересечение сплошных линий;
* оценить дистанцию до впереди идущего транспорта и предупредить об опасном сближении;
* определить скорость движения собственного автомобиля и предупредить об опасном превышении;
* зафиксировать агрессивную манеру движения (частая смена полосы – «игра в шашечки»).

Что касается оценки окружающей обстановки, то с помощью видеокамеры, направленной в сторону движения, можно:

* распознать светофоры и их текущие сигналы;
* распознать дорожные знаки с получением подсказки о допустимом режиме

движения;

* зафиксировать транспорт, внезапно появляющийся слева или справа с

прилегающих дорог;

* распознать пешеходов, велосипедистов, мотоциклистов и другие объекты;
* перед автомобилем и на обочине, получить предупреждение об опасном

сближении;

* оценить качество разметки дорожного полотна;
* зафиксировать скорость движения обгоняющего транспорта;
* распознать и зафиксировать автомобильные номера попутного транспорта;
* зафиксировать ДТП или наличие на дороге посторонних предметов, влияющих

на безопасность движения (покрышки, большие лужи, ямы, камни, трещины).

Основной вывод статьи заключается в следующем: компьютерное зрение и искусственный интеллект является, пожалуй, самым востребованным направлением развития информационных технологий.

Количество подключенных автомобилей постоянно растёт, появляются автомобили с полуавтономным и автономным вождением, соответственно востребованность инновационных сервисов на основе компьютерного зрения для этой группы автомобилей будет только возрастать.

Для успешного создания и развития новых видов сервисов для подключенных и беспилотных автомобилей требуется пересмотр и совершенствование системы подготовки кадров, реализация новых видов образовательных программ с упором на изучение специальных разделов математики, нейронных сетей, машинного обучения и компьютерного зрения.

* 1. Источник 3 – Статья «The Role of Computer Vision in the Automotive Industry»

В статье [5] описано, как машинное зрение используется в автомобильной индустрии.

Подробно рассматривается автомобильная промышленность, как движущая сила автоматизации. В Соединенных Штатах, Японии, Южной Корее, Германии и других европейских странах автомобильная индустрия играет важную роль в производственном секторе. Технологии машинного зрения используются во всех видах промышленного применения для быстрого и однозначного обнаружения объектов, что приводит к автоматизации и ускорению производственных процессов. Эти методы используются для точного контроля поверхностей компонентов, а также для выявления и автоматической сортировки дефектных изделий. Аналогично, технология также может быть использована для определения комплектности. Особенностью машинного зрения является его высокая скорость — его алгоритмам требуются всего миллисекунды для обнаружения и обработки информации с изображения. Следовательно, большое количество продуктов может проверяться автоматически и в быстрой последовательности.

В статье [5] подробно описано, как компьютерное зрение определяет направление движения объекта в трехмерном пространстве. До недавнего времени обработка данных 2D-изображений была стандартом. Оно использовалось для отслеживания движущихся объектов в последовательности изображений и определения их положения — но только в 2D-пространстве, например, продуктов на горизонтальной конвейерной ленте. Трехмерная координация объектов в более сложных процессах движения невозможна, что значительно ограничивает возможности применения технологии в автомобильной производственной среде. MVTec решает эту проблему с помощью своей технологии “3D scene flow”, многообещающей технологии 3D vision, также используемой в автомобильной промышленности. В этом решении используется так называемая структура камеры с несколькими изображениями, которая предполагает установку двух или более камер, которые записывают информацию об изображении под разными углами. Каждой точке изображения присваиваются координаты X, Y и Z. Используя последовательность нескольких изображений, теперь можно перемещать, отслеживать и точно локализовывать объекты, которые движутся в трехмерном пространстве. Технология определяет не только точное местоположение объектов, но также направление их движения и скорость в пространстве.

Особое внимание уделяется плавному и безопасному взаимодействию между человеком и машиной на производстве. Если человек входит в зону действия промышленных роботов, датчики автоматически останавливают рабочие процессы робота, что приводит к дорогостоящим сбоям в производственном процессе. 3D scene flow предлагает решение этой проблемы с помощью нескольких камер, которые контролируют рабочую зону робота. Если кто-то приближается к этой области, программное обеспечение машинного зрения, используя несколько изображений, точно знает, где находится человек и в каком направлении он движется. Робот останавливается только в том случае, если предполагается, что пути человека и машины пересекутся и столкновение неизбежно. Эта способность реагировать на траектории движения человека уменьшает количество громоздких остановок машины и перерывов в производстве. Меньшее количество остановок означает большую экономию для компании.

Как итог, технологии машинного зрения коренным образом изменят процессы в автомобильном секторе. Различные детали будут быстро и надежно обнаруживаться, а производственные процессы контролироваться, в то время как технологии машинного зрения улучшат техническое обслуживание производственных машин. Решения машинного зрения будут автоматически обнаруживать компоненты с мобильных устройств и отображать соответствующую информацию об обращении. Машинное зрение ускорит и оптимизирует логистику международных цепочек поставок. И, наконец, решения сделают взаимодействие между людьми и машинами в производственных процессах более безопасным.

* 1. Источник 4 – Статья ««Машинное» зрение: что и как видят автомобили»

В статье [6] подробно описано, что «видят» автомобили и как они это делают, используя базовые алгоритмы и методы.

Читателя знакомят с тем, как, используя массивы датчиков и камер, автомобили научились распознавать вокруг себя бортики, деревья, столбы и припаркованный транспорт. Принцип определения расстояния до объектов основывается на параллаксе движения. Техника получила название structure from motion, в основе которой лежит анализ стереопары. Стереопара позволяет идентифицировать пары сопряженных точек в пространстве, на основании которых строится карта расстояний. На ней дистанция до объекта обозначается оттенками серого — чем дальше, тем темнее. На основании полученных карт алгоритмы строят маршруты и оценивают траектории: уход от столкновений, маневрирование. Алгоритмы строят вероятностные модели для предсказания маршрутов объектов вокруг.

Также в статье описывается работы других алгоритмов, которые основаны не на использовании камер, а на системах связи «автомобиль-автомобиль» и «автомобиль-инфраструктура». Такими являются DRCA-алгоритм и метод искусственных силовых полей.

Алгоритм Decentralized Reactive Collision Avoidance предполагает, что, если автомобили находятся на большом расстоянии друг от друга, они следуют по намеченным маршрутам. Если машины сближаются, то система выполняет маневр по избеганию столкновения. Алгоритм помогает объехать и статичные объекты, однако не годится, если транспортные средства двигаются в формации. DRCA предложит автомобилям набрать безопасное расстояние относительно друг друга и «разобьёт» строй.

Метод искусственных силовых полей может применяться в комбинации с DRCA. Он основывается на вычислении искусственных силовых полей. Автомобили — это заряженные частицы. Частицы с одинаковым зарядом отталкиваются, а с противоположным — притягиваются. Поэтому автомобилям назначаются отрицательные заряды, а их желаемым маршрутам — положительные. Силовые векторы вычисляются на основании этих зарядов, а затем используются для определения направления движения. Вокруг транспортного средства формируется овальная «зона безопасности», за пределами которой влияние «зарядов» других ТС становится незначительным. Сила притяжения точки назначения считается константой. Все это позволяет оценить силу притяжения, которую «ощущают» автомобили, и на основании этих данных корректировать маршрут.

Особое внимание в статье отводится тому, как автомобили, используя компьютерное зрение, определяют полосы движения. Первый — когда инфраструктура является частью интеллектуальных систем (используются умные маячки), однако он связан со сложностью и дороговизной перестройки дорог. Второй — автомобили сами анализируют дорогу и принимают решения. Такие системы используют камеры и методы выделения характерных признаков для обозначения границ полосы. Сперва на картинке с камеры находят линии, расположенные под определенным углом к объективу. После чего у выделенных полос выделяют крайние пиксели с использованием троичного поиска. Также в статье описаны такие подходы, как проективное преобразование, «Птичий глаз».

2Часть 2. Формирование и преобразование бинарного дерева

Дерево — это иерархическая структура данных, состоящая из элементов (вершин, или узлов), которые связаны между собой отношениями типа «родительская вершина – дочерняя вершина».

Определить дерево с вершинами типа T можно следующим образом:  это либо пустое дерево, не содержащее ни одной вершины  либо некоторая вершина типа T, соединенная ветвями с конечным числом отдельных деревьев с вершинами типа T (эти деревья называются поддеревьями).

Чаще всего дерево изображается в виде графа, вершинами которого являются вершины дерева, а ребрами — его ветви.

Начальная вершина дерева, называемая корнем, изображается в верхней части графа, и считается, что она находится на нулевом уровне. Вершина Y, расположенная ниже вершины X и соединенная с ней ветвью, называется непосредственным потомком вершины X, или ее дочерней вершиной (а вершина X, соответственно, — непосредственным предком вершины Y, или ее родительской вершиной).

Если вершина X находится на уровне K, то считается, что все ее непосредственные потомки находятся на уровне K + 1. На графе, изображающем дерево, все вершины одного уровня располагаются на одной горизонтали.

Номер максимального уровня дерева называется глубиной (или высотой) дерева. Если вершина не имеет потомков, то она называется терминальной вершиной, или листом. Нетерминальная вершина называется внутренней. Число непосредственных потомков внутренней вершины дерева называется степенью этой вершины. Максимальная степень всех вершин дерева называется степенью дерева.

Дерево называется упорядоченным, если все непосредственные потомки любой вершины упорядочены.

Особым видом деревьев являются бинарные деревья. Бинарное дерево с вершинами типа T можно определить следующим образом:  это либо пустое дерево, не содержащее ни одной вершины  либо некоторая вершина типа T, соединенная ветвями с двумя бинарными деревьями с вершинами типа T (эти деревья называются левым и правым поддеревом).

Бинарные деревья нельзя считать просто упорядоченными деревьями степени 2. Различие между этими типами деревьев проявляется для вершин степени 1, то есть вершин, имеющих единственную дочернюю вершину. В «обычных» упорядоченных деревьях о

подобной дочерней вершине нельзя сказать, является ли она левой или правой, тогда как в бинарных деревьях единственная дочерняя вершина всегда будет либо левой, либо правой.

* 1. Задание 1 – Формирование дерева (задача Tree 32)
     1. Формулировка задания

Дано число N (> 0) и набор из N чисел. Создать идеально сбалансированное дерево из N вершин с заданными значениями (то есть дерево, для каждой вершины которого количество вершин в его левом и правом поддереве отличается не более чем на 1) и вывести указатель на его корень. Для создания дерева использовать рекурсивный алгоритм, который создает вершину дерева с очередным значением, после чего вызывается для создания левого поддерева с N/2 вершинами и правого поддерева с N – 1 – N/2 вершинами (символ «/» обозначает операцию деления нацело).

* + 1. Алгоритм решения

В данном разделе описаны основные шаги для создания и заполнения дерева.

1. Дерево формируется в конструкторе класса BalancedTree. В качестве параметров передаются количество вершин и список с их значениями;
2. В конструкторе создается корень дерева, являющийся объектом класса Node, который хранит числовое значение и указатели на левую и правую вершины;
3. Далее вызываются рекурсивные функции для левого и правого поддеревьев с количеством вершин соответственно N/2 и N-1-N/2. При этом при каждом вызове функций происходит проверка на N=0.

Объект сформирован. Работа алгоритма завершена.

Визуализация работы конструктора класса BalancedTree представлена на иллюстрации из приложения А. Листинг программы представлен в приложении Б.

* + 1. Визуализация работы алгоритма

Программа сопровождается вводом входных данных. Пример ввода входных данных показан на рисунке 1.

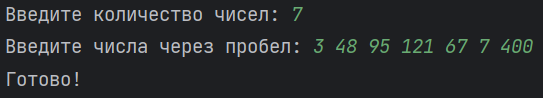


Рисунок 1 – Ввод данных для формирования дерева

Далее входные данные поступают в конструктор согласно алгоритму. Результирующее из входных данных на рисунке 1 дерево отражено на рисунке 2. Пунктирные линии соединяют с левой дочерней вершиной, сплошные – с правой. Работа программы завершена.

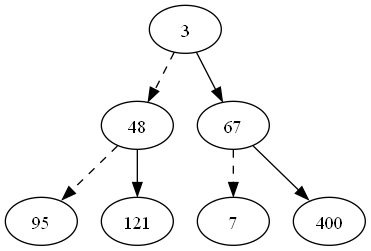


Рисунок 2 – Вывод сформированного бинарного дерева

* 1. Задание 2 – Преобразование дерева (задача Tree 43)
     1. Формулировка задания

Дан указатель P1 на корень непустого дерева. Для вершин дерева, имеющих две дочерние вершины, удалить одну из дочерних вершин: правую, если родительская вершина имеет четное значение, и левую в противном случае (вершины дерева перебирать в префиксном порядке, при удалении вершины удалять и всех ее потомков). Для удаленных вершин освобождать память, которую они занимали.

* + 1. Алгоритм решения

Начинаем процедуру преобразования дерева.

1. Если у текущей вершины есть левая и правая дочерние вершины, то в зависимости от значения текущей вершины вызывается рекурсивная процедура удаления всех потомков левой или правой вершины дерева. Потом удаляется и сама дочерняя вершина;
2. Если у текущей вершины есть только левая или только правая дочерняя вершина, то начинается рекурсивная процедура преобразования дерева от соответственно левой или правой дочерней вершины;

Листинг программы представлен в приложении Б. Блок-схема алгоритма метода, преобразующего бинарное дерево согласно заданию, представлена в иллюстрации в приложении В.

* + 1. Визуализация работы алгоритма

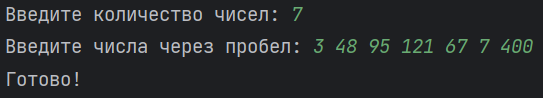


Рисунок 3 – Ввод данных для формирования дерева

Используя полученные входные данные, создается двоичное дерево поиска для удобства. Созданное дерево представлено на рисунке 4. Пунктирные линии соединяют с левой дочерней вершиной, сплошные – с правой.

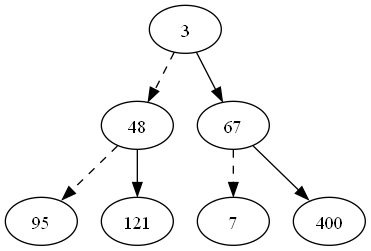


Рисунок 4 – Созданное двоичное дерево поиска

Результирующее из входных данных на рисунке 3 дерево отражено на рисунке. Работа программы завершена.

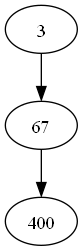


Рисунок 4 – Созданное двоичное дерево поиска

ЗАКЛЮЧЕНИЕ

В ходе выполнения учебной практики были получены следующие навыки:

* исследования различных источников по теме НИР «Компьютерное зрение в автомобильной индустрии» и составления презентации;
* работы со структурой данных «бинарное дерево» и выполнение задач по его формированию и преобразованию;
* работы со стандартом оформления текстовых документов Самарского Университета и подготовки учебного текстового документа на его основе.

Были успешно использованы и закреплены знания, полученные за 2 курса обучения в университете.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

– 71 с.

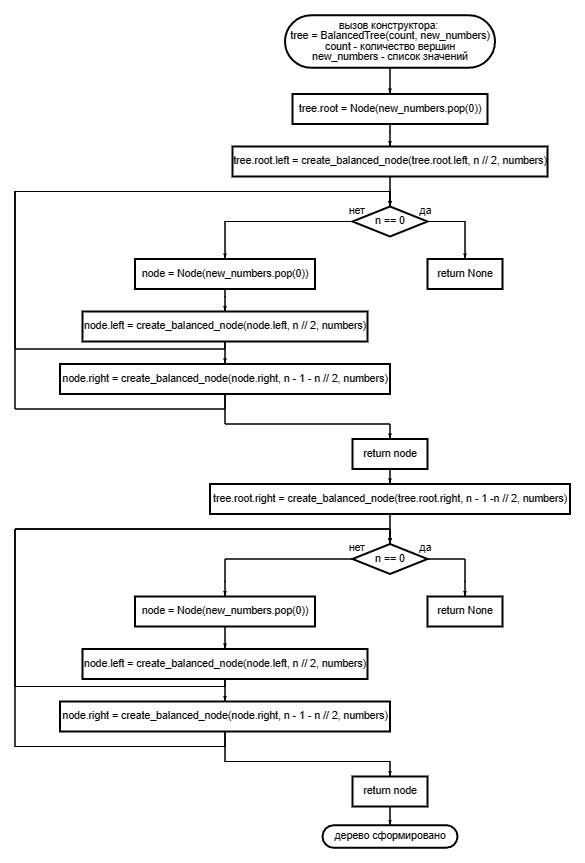
1. Абрамян, М.Э. Бинарные деревья. [Текст] / М.Э. Абрамян – Ростов-на-Дону, 2009.
2. СТО 02068410-004-2018 Общие требования к учебным текстовым документами

[Текст]. - Самара: Самарский университет, 2018. - 36 с.

1. The Role of Computer Vision in the Automotive Industry [Электронный ресурс] // Techexpert. 2021 – URL:  [The Role of Computer Vision in the Automotive Industry - Techiexpert.com](https://www.techiexpert.com/the-role-of-computer-vision-in-the-automotive-industry/) (дата обращения: 02.07.2023).
2. Перспективы применения искусственного интеллекта и компьютерного зрения в транспортных системах и подключенных автомобилях [Электронный ресурс] // Мир транспорта. 2021 – URL: [Перспективы применения искусственного интеллекта и компьютерного зрения в транспортных системах и подключенных автомобилях | Постолит | Мир транспорта (elpub.ru)](https://mirtr.elpub.ru/jour/article/view/2079) (дата обращения: 03.07.2023)
3. The Role of Computer Vision in the Automotive Industry [Электронный ресурс] // Vision Spectra. 2016 – URL: [The Role of Machine Vision in the Automotive Industry | Business | Jan 2016 | Vision Spectra (photonics.com)](https://www.photonics.com/Articles/The_Role_of_Machine_Vision_in_the_Automotive/a58196) (дата обращения: 03.07.2023).
4. «Машинное» зрение: что и как видят автомобили [Электронный ресурс] // Хабр. 2017 – URL: [«Машинное» зрение: что и как видят автомобили / Хабр (habr.com)](https://habr.com/ru/companies/wayray/articles/407775/) (дата обращения: 03.07.2023)

ПРИЛОЖЕНИЕ А

Блок-схема алгоритма формирования дерева



ПРИЛОЖЕНИЕ Б

Листинг программы

import graphviz

class Node:

def \_\_init\_\_(self, data):

self.left = None

self.right = None

self.data = data

class BalancedTree:

def \_\_init\_\_(self, n, numbers):

self.root = Node(numbers.pop(0))

self.root.left = create\_balanced\_node(self.root.left, n // 2, numbers)

self.root.right = create\_balanced\_node(self.root.right, n - 1 - n // 2, numbers)

def create\_balanced\_node(node, n, numbers):

if n == 0:

return None

else:

node = Node(numbers.pop(0))

node.left = create\_balanced\_node(node.left, n // 2, numbers)

node.right = create\_balanced\_node(node.right, n - 1 - n // 2, numbers)

return node

def convert\_tree(node: Node):

if node.left and node.right:

if node.data % 2 == 1:

delete\_nodes(node.left)

node.left = None

convert\_tree(node.right)

else:

delete\_nodes(node.right)

node.right = None

convert\_tree(node.left)

elif node.left:

convert\_tree(node.left)

elif node.right:

convert\_tree(node.right)

def delete\_nodes(node: Node):

if node.left:

delete\_nodes(node.left)

if node.right:

delete\_nodes(node.right)

node.left = None

node.right = None

def show\_tree(my\_nood, my\_dot):

if my\_nood.left:

new\_node = str(my\_nood.left.data)

old\_node = str(my\_nood.data)

my\_dot.node(new\_node, new\_node)

my\_dot.edge(old\_node, new\_node, style="dashed")

show\_tree(my\_nood.left, my\_dot)

if my\_nood.right:

new\_node = str(my\_nood.right.data)

old\_node = str(my\_nood.data)

my\_dot.node(new\_node, new\_node)

my\_dot.edge(old\_node, new\_node)

show\_tree(my\_nood.right, my\_dot)

count = int(input('Введите количество чисел: '))

s = input('Введите числа через пробел: ')

new\_numbers = [int(x) for x in s.split()]

new\_tree = BalancedTree(count, new\_numbers)

dot = graphviz.Digraph()

show\_tree(new\_tree.root, dot)

dot.format = 'png'

dot.render('Graph', view=True)

convert\_tree(new\_tree.root)

new\_dot = graphviz.Digraph()

show\_tree(new\_tree.root, new\_dot)

new\_dot.format = 'png'

new\_dot.render('Graph1', view=True)

print('Готово!')

ПРИЛОЖЕНИЕ В

Блок-схема алгоритма преобразования дерева

