

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

«СОГЛАСОВАНО»

Заведующий кафедрой
«Информационные технологии
и системы»



«30» 01 2026 г.

Д.В. Моисеев

«УТВЕРЖДАЮ»

Директор института
Высшая технологическая школа
«Севастопольский
приборостроительный
институт»



«30» 01 2026 г.

В.В. Мешков

РАБОЧАЯ ПРОГРАММА ПРАКТИКИ

Учебная

(вид практики)

Б2.О.01 (У) Технологическая практика

(тип практики)

09.03.03 Прикладная информатика

(код и наименование направления подготовки)

Геоинформационные технологии и 3D-моделирование

(наименование профиля подготовки)

бакалавриат

(уровень высшего образования)

очная, 2026

(форма обучения)

Севастополь

2026

Рабочая программа учебной (Технологической) практики составлена на основе Федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению подготовки 09.03.03 Прикладная информатика, утвержденного приказом Минобрнауки РФ от 19.09.2017 г. № 922.

Рабочая программа учебной (Технологической) практики рассмотрена и одобрена на заседании кафедры «Информационные технологии и системы» от «30» 01 2026 г. протокол № 7

Руководитель образовательной программы:  И.П. Шумейко

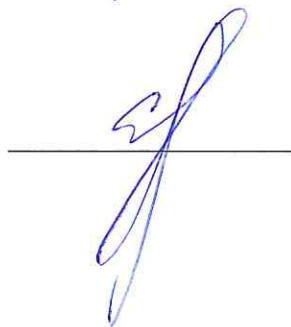
Рабочая программа составлена:

Профессор кафедры ИТиС,
доцент ИОЦ «Центр ИИ
СевГУ», к.ф.-м.н., доцент



И.П. Шумейко

Ст. преподаватель кафедры
«Информационные технологии
и системы»



М.В. Собченко

СОДЕРЖАНИЕ

1.	Цели практики.....	4
2.	Задачи практики	4
3.	Место практики в структуре ООП	4
4.	Тип практики, способ и форма ее проведения	5
5.	Планируемые результаты обучения при прохождении учебной практики, соотнесенные с планируемыми результатами освоения образовательной программы.....	6
5.1.	Предварительные и дополнительные условия.....	6
5.2.	Формируемые компетенции	6
6.	Структура и содержание учебной практики.....	8
7.	Форма отчетности по практике	10
8.	Фонд оценочных средств для проведения учебной практики.....	10
9.	Учебно-методическое и информационное обеспечение учебной практики.....	16
10.	Материально-техническое обеспечение учебной практики..	18
	Приложение А. Структура и требования к оформлению отчета о практике.....	19
	Приложение Б. Пример оформления титульного листа отчета о практике.....	20
	Приложение В. Перечень основного оборудования и программного обеспечения специализированных аудиторий и лабораторий для проведения учебной практики	21

1 Цели практики

Учебная Технологическая практика (далее – учебная практика) является обязательным элементом основных образовательных программ бакалавриата по направлению подготовки 09.03.03 «Прикладная информатика», профиль «Геоинформационные технологии и 3D-моделирование», квалификация (степень) – бакалавр.

Целью учебной практики является: закрепление и углубление теоретических знаний, полученных при изучении дисциплин математического и естественнонаучного цикла и профессионального цикла в области информатики, алгоритмизации, программирования, прикладных информационных технологий, а также приобретение практических навыков и компетенций разработки и отладки программ на языках высокого уровня.

2 Задачи практики

Основными задачами учебной практики является приобретение и совершенствование практических навыков и компетенций в областях:

- изучения современного состояния и направлений развития компьютерной техники и информационных технологий;
- работы с разнообразными источниками информации, методами сбора исходных данных, необходимых для решения задач;
- выбора компьютерных методов сбора, хранения и обработки информации в соответствии с поставленными задачами;
- разработок алгоритмов и программных решений поставленных задач;
- изучение и использование современных программных средств и технологий создания, отладки и исследования программных продуктов при решении поставленных задач;
- обобщение и представление результатов разработок и исследований созданных программных продуктов.

Задачами практики являются:

- углубление и закрепление полученных теоретических знаний;
- ознакомление на практике с методологиями управления проектами разработки программного обеспечения, такими как Scrum, Kanban, WaterFall;
- получение навыков управления компьютером с помощью работы в командной строке;
- практическое применение интерактивной среды разработки IDE для улучшения процесса написания программного кода;
- изучение возможностей систем сборки проектов, на примере Gradle;

- получение навыков в использовании системы контроля версии (типа Git) для хранения, отслеживания и внесения изменений в программный проект;
- получение навыков совместной работы над проектом через Github/Gitlab/Bitbucket/Azure (по выбору команды студентов);
- изучение возможностей Pull Requests / Merge Requests, как способов командной работы с 1 кодом;
- получение навыков тестирования программ с помощью unit-тестов (на примере Unit-5 для Java);
- автоматический запуск тестов в CI/CD в выбранной платформе совместной работы;
- получение практических навыков разработки программного проекта;
- получение навыков взаимодействия в командной работе над учебным программным проектом;
- получение навыков в подготовке презентаций и технических отчетов

3 Место практики в структуре ООП

Учебная практика Б2.О.01(У) является обязательной составляющей учебного процесса, входящей в блок Б2. Для ее успешного освоения требуются знания и умения, полученные в дисциплинах обязательной части образовательной программы. Пререквизитами данной дисциплины являются следующие дисциплины обязательной части блока Б1: «Алгоритмизация и программирование», «Цифровые технологии и медиабезопасность», «Дискретная математика», «Объектно-ориентированное программирование».

Для успешного прохождения учебной практики необходимы следующие знания, умения и навыки, формируемые предшествующими дисциплинами.

Из курса «Дискретная математика»:

знания: основных положений алгебры логики, логические функции и способы их преобразования;

умения: составлять и минимизировать логические выражения;

владения: анализом логических функций и методами их преобразования.

Из курса «Цифровые технологии и медиабезопасность»:

знания: начальные знания в области информационных систем и технологий, оформления технической документации;

умения: использования системного подхода к изучению профессиональных и общеобразовательных дисциплин;

владения: опытом практического применения текстового процессора Word, табличного процессора Excel и системы построения презентаций PowerPoint.

Из курса «Алгоритмизация и программирование», «Объектно-ориентированное программирование»:

обучающийся должен свободно владеть персональным компьютером, знать языки программирования Си, Си++, уметь пользоваться библиотеками

процедурных языков программирования, владеть знаниями особенностей модульного программирования и обработки сложных структур данных (файлы, динамические структуры).

Знания и умения, полученные в результате прохождения учебной практики, являются основой для дальнейшей теоретической и практической подготовки по дисциплинам: «Инфокоммуникационные системы и сети», «Системный анализ и проектирование информационных систем» и др.

4 Тип практики, способ и форма ее проведения

Тип учебной практики – технологическая практика. Форма проведения практики – концентрированная. Способ проведения данного типа учебной практики: стационарная.

Учебная практика представляет собой вид учебных занятий, ориентированных на профессионально-практическую подготовку обучающихся, путем:

- непосредственного участия студента в деятельности подразделений организации;
- непосредственного участия студента в деятельности учреждений и организаций различных форм собственности;
- путем выполнения индивидуальных заданий, направленных на решение конкретных задач.

5 Планируемые результаты обучения при прохождении учебной практики, соотнесенные с планируемыми результатами освоения образовательной программы

В соответствии с Федеральным государственным образовательным стандартом по направлению 09.03.03 «Прикладная информатика» в результате освоения программы учебной практики должны быть сформированы следующие компетенции: УК-1; УК-2; УК-3; УК-4; УК-5; УК-6; УК-9; ОПК-1; ОПК-2; ОПК-3; ОПК-4; ОПК-5; ОПК-6; ОПК-7; ОПК-8; ОПК-9; SS 1; SS 2; SS 3

Формируемые компетенции (код компетенции, уровень(этап) освоения)	Планируемые результаты обучения при прохождении практики
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1. Знает принципы сбора, отбора и обобщения информации, методики системного подхода для решения профессиональных задач. УК-1.2. Умеет анализировать и систематизировать разнородные данные, оценивать эффективность процедур анализа проблем и принятия решений в профессиональной деятельности. УК-1.3. Владеет навыками научного поиска и практической работы с информационными источниками; методами принятия решений.
УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1. Знает необходимые для осуществления профессиональной деятельности правовые нормы и методологические основы принятия управленческого решения. УК-2.2. Умеет анализировать альтернативные варианты решений для достижения намеченных результатов; разрабатывать план, определять целевые этапы и основные направления работ. УК-2.3. Владеет методиками разработки цели и задач проекта; методами оценки продолжительности и стоимости проекта, а также потребности в ресурсах.
УК-3 Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	УК-3.1. Знает типологию и факторы формирования команд, способы социального взаимодействия. УК-3.2. Умеет действовать в духе сотрудничества; принимать решения с соблюдением этических принципов их реализации; проявлять уважение к мнению и культуре других; определять цели и работать в направлении личностного, образовательного и профессионального роста. УК-3.3. Владеет навыками распределения ролей в условиях командного взаимодействия; методами оценки своих действий, планирования и управления временем
УК-4 Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(ых) языке(ах)	УК-4.1. Знает принципы построения устного и письменного высказывания на государственном и иностранном языках; требования к деловой устной и письменной коммуникации. УК-4.2. Умеет применять на практике устную и письменную деловую коммуникацию.

Формируемые компетенции (код компетенции, уровень(этап) освоения)	Планируемые результаты обучения при прохождении практики
	УК-4.3. Владеет методикой составления суждения в межличностном деловом общении на государственном и иностранном языках, с применением адекватных языковых форм и средств
УК-5 Способен воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах	УК-5.1. Знает основные категории философии, законы исторического развития, основы межкультурной коммуникации. УК-5.2. Умеет вести коммуникацию в мире культурного многообразия и демонстрировать взаимопонимание между обучающимися – представителями различных культур с соблюдением этических и межкультурных норм. УК-5.3. Владеет практическими навыками анализа философских и исторических фактов, оценки явлений культуры; способами анализа и пересмотра своих взглядов в случае разногласий и конфликтов в межкультурной коммуникации.
УК-6 Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни	УК-6.1. Знает основные принципы самовоспитания и самообразования, исходя из требований рынка труда. УК-6.2. Умеет демонстрировать умение самоконтроля и рефлексии, позволяющие самостоятельно корректировать обучение по выбранной траектории. УК-6.3. Владеет способами управления своей познавательной деятельностью и удовлетворения образовательных интересов и потребностей.
УК-9 Способен принимать обоснованные экономические решения в различных областях жизнедеятельности	УК-9.1. Знает экономические основы, необходимые для осуществления социальной и профессиональной деятельности. УК-9.2. Умеет воспринимать и анализировать информацию, необходимую для принятия обоснованных экономических решений. УК-9.3. Имеет практический опыт применения экономических законов и основ финансовой грамотности при планировании личного бюджета и профессиональной деятельности
ОПК-1 Способен применять естественнонаучные и общетехнические знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности;	ОПК-1.1. Знает основы математики, физики, вычислительной техники и программирования. ОПК-1.2. Умеет решать стандартные профессиональные задачи с применением естественнонаучных и общетехнических знаний, методов математического анализа и моделирования. ОПК-1.3. Владеет навыками теоретического и экспериментального исследования объектов профессиональной деятельности.
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности;	ОПК-2.1. Знает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности. ОПК-2.2. Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства

Формируемые компетенции <i>(код компетенции, уровень(этап) освоения)</i>	Планируемые результаты обучения при прохождении практики
	при решении задач профессиональной деятельности. ОПК-2.3. Владеет навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности.
ОПК-3 Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;	ОПК-3.1. Знает принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности ОПК-3.2. Умеет решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности, создавать прототипы ИС ОПК-3.3. Владеет навыками подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с учетом требований информационной безопасности.
ОПК-4 Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью;	ОПК-4.1. Знает основные стандарты оформления технической документации на различных стадиях жизненного цикла информационной системы. ОПК-4.2. Умеет применять стандарты оформления технической документации на различных стадиях жизненного цикла информационной системы. ОПК-4.3. Владеет навыками составления технической документации на различных этапах жизненного цикла информационной системы.
ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем;	ОПК-5.1. Знает основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем. ОПК-5.2. Умеет выполнять параметрическую настройку информационных и автоматизированных систем ОПК-5.3. Владеет навыками установки программного и аппаратного обеспечения информационных и автоматизированных систем
ОПК-6 Способен анализировать и разрабатывать организационно-технические и экономические процессы с применением методов системного анализа и математического моделирования;	ОПК-6.1. Знает основы теории систем и системного анализа, дискретной математики, теории вероятностей и математической статистики, методов оптимизации и исследования операций, нечетких вычислений, математического и имитационного моделирования. ОПК-6.2. Умеет применять методы теории систем и системного анализа, математического, статистического и имитационного моделирования

Формируемые компетенции (код компетенции, уровень(этап) освоения)	Планируемые результаты обучения при прохождении практики
	для автоматизации задач принятия решений, анализа информационных потоков, расчета экономической эффективности и надежности информационных систем и технологий. ОПК-6.3. Владеет навыками проведения инженерных расчетов основных показателей результативности создания и применения информационных систем и технологий.
ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения;	ОПК-7.1. Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий. ОПК-7.2. Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. ОПК-7.3. Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач.
ОПК-8 Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла;	ОПК-8.1. Знает основные технологии создания и внедрения информационных систем, стандарты управления жизненным циклом информационной системы. ОПК-8.2. Умеет осуществлять организационное обеспечение выполнения работ на всех стадиях и в процессах жизненного цикла информационной системы. ОПК-8.3. Владеет навыками составления плановой и отчетной документации по управлению проектами создания информационных систем на стадиях жизненного цикла.
ОПК-9 Способен принимать участие в реализации профессиональных коммуникаций с заинтересованными участниками проектной деятельности и в рамках проектных групп.	ОПК-9.1. Знает инструменты и методы коммуникаций в проектах; каналы коммуникаций в проектах; модели коммуникаций в проектах; технологии межличностной и групповой коммуникации в деловом взаимодействии, основы конфликтологии, технологии подготовки и проведения презентаций. ОПК-9.2. Умеет осуществлять взаимодействие с заказчиком в процессе реализации проекта; принимать участие в командообразовании и развитии персонала. ОПК-9.3. Владеет навыками проведения презентаций, переговоров, публичных выступлений.
SS-1 Способен осуществлять свою трудовую деятельность с учетом определения корректной роли ИИ в различных процессах, критического	SS-1.2 Учитывает в разработке и эксплуатации систем ИИ философские основания концепций интеллекта, языка, знания, агентности Уровень освоения индикатора – средний Умеет выявлять ценностные конфликты в

Формируемые компетенции (код компетенции, уровень(этап) освоения)	Планируемые результаты обучения при прохождении практики
анализа последствий применения ИИ-технологий, этических принципов	конкретных рабочих ситуациях (например, при разработке рекомендательной системы, системы распознавания изображений и т.п.). Соотносит технические характеристики модели с потенциальными рисками её применения.
SS-2 Способен осуществлять свою трудовую деятельность с учётом необходимости эффективной коммуникации и взаимодействия в рамках коллективной проектной работы в сфере ИИ	SS-2.1 Эффективно коммуницирует с участниками проектной команды при планировании, реализации и анализе результатов работы в контексте гибридной команды "Человек+ИИ", включая постановку задач людям и ИИ-агентам, фиксацию договорённостей и критериев качества Уровень освоения индикатора – средний Активно участвует в обсуждении постановки задач с учётом требований заказчика и особенностей данных, формулирует предложения по задачам и распределению ролей. Вносит предложения по отбору релевантных метрик по проекту. Согласовывает цели проекта с возможностями проектируемой ИИ-системы и ограничениями имеющейся инфраструктуры. Принимает участие в приоритизации задач и распределении ответственности. Переводит цели в формализованные задачи для членов команды и ИИ-агентов с указанием контекста, входов/выходов, сроков и критериев качества; осуществляет проверку понимания SS-2.2 Учитывает профессиональные и ролевые особенности коллег и контур ИИ-компонентов: адаптирует язык под аудиторию (tech/product/C-level), распределяет ответственность (RACI) и представляет результаты в понятном формате Уровень освоения индикатора – средний Совместно с командой определяет, как представить ключевые компоненты ИИ-решения (например, пайплайн, выбор моделей, оценка качества). Учитывает уровень подготовленности аудитории (например, заказчик, который не разбирается в ИИ-технологиях). Координирует связность выступления с другими участниками.

Формируемые компетенции (код компетенции, уровень(этап) освоения)	Планируемые результаты обучения при прохождении практики
	Адаптирует сообщение под аудиторию (tech/product/C-level), распределяет ответственности (с применением матрицы RACI) и специфицирует интерфейсы «люди ↔ ИИ» (форматы данных, входные/выходные требования, критерии качества). SS-2.0.1 Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла Знает способы мониторинга и управления работами проекта в соответствии с установленными регламентами; умеет определять первоначальные требования заказчика к ИС и возможности их реализации в ИС на этапе предконтрактных работ; имеет навыки планирования проекта в соответствии с полученным заданием.
SS-3 Способен осуществлять свою трудовую функцию с учетом неопределенности как сущностной черты функционирования искусственного интеллекта	SS-3.2 Определяет релевантность применения ИИ для решения конкретных задач, анализирует поведение ИИ в техническом, социальном и правовом контекстах, переносит идеи и методы за пределы исходной предметной области Уровень освоения индикатора – средний Оценивает целесообразность и ограничения применения ИИ для различных задач с учётом технических, социальных и правовых условий; сравнивает подходы из разных областей и адаптирует методы к новому контексту.

6 Структура и содержание учебной практики

6.1 Структура учебной практики

Учебная (технологическая) практика проводится в 4-м семестре. Объем – 3 зачетных единицы (продолжительность – 2 недели). Способ проведения данного типа учебной практики: стационарная. Результаты практики оцениваются дифференциальным зачетом.

6.2. Содержание учебной практики

№ п/ п	Разделы (этапы) практики	Виды учебной деятельности на практике и трудоемкость (в часах)		Формы текущего контроля
		Виды работы	Сум.	
1	Вводное занятие.	Инструктаж по ТБ.	12	Проверка руководителем
2	Изучение структуры предприятия. Информационный поиск	Экскурсия, Лекции (3 часа)	3	Раздел отчета
3	Изучение информационных систем и технологий.	Работа с технической литературой, в сети Интернет.	24	Раздел отчета
4	Выполнение индивидуальных заданий.	Индивидуальная работа.	48	Раздел отчета
5	Участие в выполнении производственных заданий на рабочих местах.	Работа в коллективе предприятия/орг анизации	12	Характеристика руководителя от предприятия.
6	Изучение нормативных документов и оформление и защита отчета по практике.	Оформление отчета и его защита.	9	Защита отчета
	Всего часов:	3 ЗЕ	108	

Раздел 1. Вводное занятие.

Инструкция по технике безопасности. Обзор задач, выносимых на практическое изучение. Выдача вариантов задания. Рекомендации по поиску дополнительного теоретического материала.

Раздел 2. Изучение структуры предприятия\ Информационный поиск.

Экскурсия по предприятию*, знакомство с его структурой и функциональными обязанностями ИТ-отдела (группы). Поиск и анализ литературных источников по теме индивидуального задания.

Раздел 3. Изучение информационных систем и технологий.

Изучение информационных систем и технологий, которые могут быть использованы для выполнения индивидуального задания / Изучение информационных систем и технологий, используемых на предприятии, программного и информационного обеспечения, технических средств конкретной информационной системы*.

Раздел 4. Выполнение индивидуальных заданий.

Выполнение индивидуальных заданий под руководством руководителя практики / Выполнение индивидуальных заданий под руководством специалиста от предприятия и руководителя от университета*.

Раздел 5. Участие в выполнении производственных заданий на рабочих местах.

Участие в выполнении производственных заданий на рабочих местах. Выполнение индивидуальных заданий под руководством руководителя практики.

Раздел 6. Изучение нормативных документов.

Изучение документов ЕСПД и ЕСКД. Оформление и защита отчета по практике.

* – для студентов, проходящих практику в сторонних организациях.

7 Форма отчетности по практике

Результаты прохождения практики оцениваются посредством промежуточной аттестации – защитой отчета о прохождении практики в форме собеседования практиканта с руководителем практики с выставлением оценки. Отчет должен быть представлен не позднее 3 дней с даты окончания практики. Отчета по практике должен содержать следующие подразделы:

- титульный лист;
- индивидуальное задание на практику;
- отзыв руководителя практики от профильной организации;
- содержание;
- введение;
- основная часть;
- заключение;
- список использованных источников;
- приложения.

Оформление отчета о практике должно соответствовать требованиям к текстовым учебным документам соответствующих ГОСТов.

Текстовая часть отчета о практике выполняется с использованием печатающих и графических устройств на одной стороне листа белой бумаги формата А4 с параметрами: междустрочный интервал – 15; кегль – 14; шрифт – TimesNewRoman, обычный; цвет шрифта – черный; поля, не менее:

верхнее - 20 мм; левое – 30 мм;
нижнее - 20 мм; правое – 15 мм.

Иллюстрационно-графический материал в зависимости от специфики программы может включать: чертежи, схемы (технологические, кинематические, гидравлические, автоматизации, алгоритмов и др.), плакаты, диаграммы, макеты, фотографии, аудио- и видеоматериалы, натурные образцы и др.

Иллюстрационно-графический материал может быть представлен на бумажном, электронном или ином виде носителя. Возможно представление иллюстрационно-графического материала в виде брошюр.

Отчет должен быть предоставлен в переплетенном виде.

8 Фонд оценочных средств для проведения учебной практики

Оценивание результатов освоения программы практики осуществляется в результате собеседования практиканта с преподавателем, руководителем практики, а также в процессе защиты отчета по практике.

Составление плана прохождения учебной практики, выдача индивидуального задания и рекомендаций по выполнению заданий осуществляется руководителем практики (в случае прохождения в сторонней организации руководителем от организации и согласуются с руководителем практики от университета).

В период учебной практики студент ведет дневник, в котором описывает полученные задания, проводимые мероприятия и виды работ, а также ход выполнения индивидуального задания и подготовки отчета, с указанием фактических сроков выполнения отдельных этапов работы, оценками и подписями руководителя по каждому этапу

Основными документами, характеризующим качество работы студента во время учебной практики, являются: дневник; отчет о прохождении практики и выполнении индивидуального задания (реферат и отчет по практическому заданию).

Дневник должен содержать: перечень полученных заданий; описание проведенных мероприятий и видов работ; сроки выполнения, оценки и подписи руководителя по каждому этапу, в случае прохождения практики в сторонней организации руководитель от предприятия дает характеристику студенту в дневнике – в письменном виде, заверив ее подписью и печатью предприятия.

Отчет оформляется в соответствии с установленными требованиями (ГОСТ 7.32-2001) в форме реферата объемом до 25 страниц (формат бумаги – А-4), а также представлен на электронном носителе. Отчет включает изложение всех вопросов, предусмотренных программой учебной практики и индивидуальным заданием, полученным до начала практики.

После окончания учебной практики дневник и отчет сдается на кафедру. Отчёт рецензируется руководителем практики от кафедры, в которой решается вопрос о допуске его к защите. По результатам защиты отчета ставится дифференцированная оценка, приравниваемая к оценке (зачетам) по теоретическому обучению и учитываемая при подведении итогов общей успеваемости студентов.

Таблица соответствия параметров оценивания результатам контроля знаний по разным шкалам

Сумма баллов по 100-балльной системе оценивания	Оценка ECTS	Параметры оценивания	Уровень владения компетенциями	Оценка по пятибалльной системе оценивания	
				для экзамена, КП(КР), практики	для зачета
90-100	A	Обучающийся демонстрирует способность к полной самостоятельности в выборе способа решения неизвестных или нестандартных заданий в рамках выполнения плана	Высокий	отлично	зачтено
82-89	B	Обучающийся свободно владеет изученным объемом материала, применяет его на практике, свободно решает поставленные задачи в стандартных случаях, самостоятельно исправляет допущенные незначительного объема ошибки.	Достаточный	хорошо	
75-81	C	Обучающийся умеет сопоставлять, обобщать и систематизировать информацию под руководством наставников; в целом самостоятельно применять её на практике; контролировать собственную деятельность; исправлять ошибки, среди которых есть существенные, подбирать аргументы для подтверждения мыслей			
69-74	D	Обучающийся воспроизводит значительную часть теоретического материала, обнаруживает знание и понимание основных положений; с помощью преподавателя может анализировать учебный материал и исправлять ошибки среди которых есть изрядно существенных	Средний	удовлетворительно	
60-68	E	Обучающийся владеет учебным материалом на уровне «выше начального», значительная часть которого воспроизводит на репродуктивном уровне, наличие существенных ошибок при планировании и выполнении заданий практики			
35-59	FX	Обучающийся владеет материалом на уровне отдельных фрагментов, которые составляют лишь незначительную часть учебного материала, не умеет решать задачи в соответствии с основными требованиями	Низкий	неудовлетворительно	не зачтено
1-34	F	Обучающийся владеет материалом на уровне элементарного распознавания и воссоздания отдельных фактов, элементов, объектов			

Типовые контрольные задания

Тип 1 – Программные проекты повышенной сложности

Задание по практике предполагает разработку программной системы, моделирующей работу простого компьютера. Программная модель состоит из трех частей, которые должны выполняться тремя студентами в команде. Первая часть состоит в разработке программы-ассемблера, которая будет переводить программу в мнемокодах в машинные коды. Вторая часть – создание программы-симулятора компьютера, которая должна выполнять машинную программу, состоящую из команд, входящих в систему команд данного компьютера. Третья часть – разработка графического интерфейса, позволяющего управлять программами, созданными в первой и второй частях, современным способом. Варианты заданий на практику отличаются набором команд (системой команд компьютера), структурой, размером и разрядностью оперативной памяти, структурой процессора (приложение А).

Тип 2 – Программные проекты средней сложности (мини-проекты)

Перечисленные ниже мини-проекты могут выполняться как командой, состоящей из нескольких студентов, так и одним студентом (индивидуально).

Калькулятор. Реализовать простейший калькулятор, который дает возможность пользователю складывать, умножать, вычитать, делить числа, находить их квадратный корень.

Генератор паролей. Создать генератор случайных паролей для своих друзей и семьи, чтобы обезопасить их учётные записи!

Игра-викторина. Приложения-викторины задают пользователям серию вопросов и дают им возможность ответить. Затем викторина даёт пользователю результаты. Реализовать программу-викторину на любую интересную тему.

Игра "Угадай число". В данном мини-проекте необходимо реализовать игру, в которой компьютер пытается угадать целое число от 1 до 1000. Пользователь загадывает число, а программа задает ему вопросы из разряда Да или Нет в попытках угадать число. Если программа угадает число, она спрашивает у пользователя, то ли число она угадала. Если да - то «урааа!». Если нет, программа показывает грустный смайлик и предлагает новую игру.

Программа-шифровщик. В данном мини-проекте необходимо реализовать программу, которая позволяет как шифровать сообщение, так и дешифровать сообщение. Предполагается, что обладая знанием ключа/ключей, студент может зашифровать сообщение, а другой студент - расшифровать, и наоборот. Описание вышеперечисленных мини-проектов детализировано в Приложении Б.

Интеллектуальная матрица. Создать программу, реализующую объект «Числовая матрица», способный выполнять заданные математические операции над инкапсулированной матрицей.

9 Учебно-методическое и информационное обеспечение учебной практики

9.1 Основная и дополнительная литература

№	Наименование и полное библиографическое описание	Количество экземпляров в библиотеке
Основная литература		
1	Гниденко, И. Г. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 248 с. — (Высшее образование). — ISBN 978-5-534-18130-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/536903 (дата обращения: 30.01.2025).	Индивидуальный доступ без ограничений числа пользователей, регистрация по IP-адресам СевГУ
2	Огнева, М. В. Программирование на языке C++: практический курс : учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина, А. А. Казачкова. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 342 с. — (Высшее образование). — ISBN 978-5-534-18949-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/555533 (дата обращения: 30.01.2025).	Индивидуальный доступ без ограничений числа пользователей, регистрация по IP-адресам СевГУ
3	Трофимов, В. В. Алгоритмизация и программирование : учебник для вузов / В. В. Трофимов, Т. А. Павловская. — 4-е изд. — Москва : Издательство Юрайт, 2025. — 108 с. — (Высшее образование). — ISBN 978-5-534-20430-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/562040 (дата обращения: 30.01.2025).	Индивидуальный доступ без ограничений числа пользователей, регистрация по IP-адресам СевГУ
4	Барков, И. А. Объектно-ориентированное программирование / И. А. Барков. — 2-е изд., стер. — Санкт-Петербург : Лань, 2023. — 700 с. — ISBN 978-5-507-47113-3. — Текст : электронный // Лань : электронно-	Индивидуальный доступ без ограничений числа пользователей,

№	Наименование и полное библиографическое описание	Количество экземпляров в библиотеке
	библиотечная система. — URL: https://e.lanbook.com/book/329549	регистрация по IP-адресам СевГУ
Дополнительная литература		
1	Свердлов, С. З. Языки программирования и методы трансляции : учебное пособие для вузов / С. З. Свердлов. — 5-е изд., стер. — Санкт-Петербург : Лань, 2025. — 564 с. — ISBN 978-5-507-50570-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/447398 (дата обращения: 30.01.2025). — Режим доступа: для авториз. пользователей.	Индивидуальный доступ без ограничений числа пользователей, регистрация по IP-адресам СевГУ
2	Солдатенко, И. С. Практическое введение в язык программирования Си : учебное пособие / И. С. Солдатенко, И. В. Попов. — Санкт-Петербург : Лань, 2022. — 132 с. — ISBN 978-5-8114-3150-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/213149	Индивидуальный доступ без ограничений числа пользователей, регистрация по IP-адресам СевГУ
3	Андрианова, А. А. Алгоритмизация и программирование. Практикум : учебное пособие / А. А. Андрианова, Л. Н. Исмагилов, Т. М. Мухтарова. — Санкт-Петербург : Лань, 2022. — 240 с. — ISBN 978-5-8114-3336-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/206258	Индивидуальный доступ без ограничений числа пользователей, регистрация по IP-адресам СевГУ
4	Платонова, О. В. Алгоритмические основы обработки данных : методические указания / О. В. Платонова, Ю. С. Асадова, М. М. Расулов. — Москва : РТУ МИРЭА, 2022. — 73 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/256586 . — Режим доступа: для авториз. пользователей.	Индивидуальный доступ без ограничений числа пользователей, регистрация по IP-адресам СевГУ

9.2 Интернет-ресурсы

№	Адрес сайта и его описание	Перечень материалов, представленных на сайте
1.	Национальный открытый университет ИНТУИТ http://www.intuit.ru/studies/courses/628/484/info	Программирование на языке высокого уровня Паскаль. Автор курса Павловская Т.А.
2	Национальный открытый университет ИНТУИТ http://www.intuit.ru/studies/courses/626/482/info	Программирование на языке высокого уровня C/C++. Автор курса Павловская Т.А. В лекциях кратко изложены основные особенности языка C/C++ в соответствии со стандартом ISO/IEC 14882

9.3 Методические указания по практике

Учебно-методическое обеспечение учебной практики рекомендуется руководителем практики индивидуально в зависимости от темы проекта или темы исследования.

9.4 Информационные технологии, используемые при учебной практике

Наименование информационной технологии/ программного продукта	Назначение (базы и банки данных, тестирующие программы, практикум, деловые игры и т.д.)	Тип продукта (полная лицензионная версия, учебная версия, демоверсия и т.п.)
Borland Pascal, PascalABC.NET	Интегрированная среда разработки с отладчиком и компилятор для языка Pascal.	Полная версия, LGPLv3, свободное ПО
Dev-C++ (Borland C/C++)	Интегрированная среда разработки для языков C, C++ в комплекте с компиляторами GCC (MinGW)	Полная версия, GNU General Public License3.0 (GPLv3), свободное ПО
Интегрированная среда разработки Eclipse CDT (MinGW) http://www.eclipse.org/platform	Интегрированная среда разработки для языка C/C++ и компилятор GCC	Свободное ПО – общественная лицензия – Eclipse Public License (EPL). Полная версия.
MS Office	Создание и редактирования документов и презентаций	Microsoft Windows Professional 7/8.1/10 Microsoft Office Professional 2010/2013/2016

10 Материально-техническое обеспечение учебной практики

Для полноценного прохождения производственной практики в распоряжение студентов предоставлены компьютерные классы и учебные лаборатории кафедры «Информационные технологии и компьютерные системы»

и Центра коллективного пользования «Афалина», укомплектованные современным вычислительным оборудованием и периферией, специализированные учебные и научно-исследовательские лаборатории различного профиля. Облачные вычислительные ресурсы, предоставляемые якорным индустриальным партнёром ООО «ИТ ГЛОБАЛ КОМ»:

- NVIDIA A800. Предназначен для высокопроизводительных вычислений и анализа данных. Имеет 6912 ядер CUDA и 40 ГБ памяти HBM2e, что делает его эффективным для задач глубокого обучения и машинного обучения;

- NVIDIA A16. Оптимизирован для виртуализации графических рабочих станций. Содержит четыре графических процессора на одной плате, каждый с 1280 ядрами CUDA и 16 ГБ памяти GDDR6, что обеспечивает высокую плотность виртуальных рабочих мест и подходит для удаленной работы с графически интенсивными приложениями;

- NVIDIA H100. Представляет собой передовую видеокарту для задач глубокого обучения и высокопроизводительных вычислений. С поддержкой технологии Transformer Engine и высокой пропускной способностью памяти, H100 обеспечивает значительное ускорение обучения больших языковых моделей и других сложных ИИ-задач;

- NVIDIA L40S. Предназначена для универсальных задач, включая генеративный ИИ, 3D-рендеринг и обработку видео. С 48 ГБ памяти GDDR6 и поддержкой новейших технологий, таких как DLSS 3 и третье поколение RT-ядер, L40S обеспечивает высокую производительность в графических и ИИ-приложениях;

- Также, на GPU Cloud доступен чип NVIDIA A100. Чип оснащен 6912 ядрами CUDA и 432 тензорными ядрами, а также поддерживает технологию Multi-Instance GPU (MIG).

В библиотеке вуза студентам обеспечивается доступ к справочной, научной и учебной литературе, монографиям и периодическим научным изданиям по специальности.

Необходимое для проведения практики материально-техническое обеспечение предоставляется базами практик и представляет собой:

- специально оборудованные компьютерные классы и лаборатории образовательного учреждения, соответствующие действующим санитарным и противопожарным нормам, а также требованиям техники безопасности при проведении учебных и научно производственных работ;

- полигоны, лаборатории, специально оборудованные кабинеты, измерительные и вычислительные комплексы, транспортные средства, бытовые помещения, предоставляемые базами практики, соответствующие действующим санитарным и противопожарным нормам, а также требованиям техники безопасности при проведении работ.

Перечень основного оборудования и программного обеспечения специализированных аудиторий и лабораторий для проведения

производственной практики на базе кафедры «Информационные технологии и системы» приведен в приложении В.

Структура и требования к оформлению отчета о практике

Структурными элементами отчета о практике являются:

- титульный лист;
- индивидуальное задание на практику;
- отзыв руководителя практики от профильной организации;
- содержание;
- введение;
- основная часть;
- заключение;
- список использованных источников;
- приложения.

Требования к оформлению отчета о практике

Оформление отчета о практике должно соответствовать требованиям к текстовым учебным документам соответствующих ГОСТов.

Текстовая часть отчета о практике выполняется с использованием печатающих и графических устройств на одной стороне листа белой бумаги формата А4 с параметрами: междустрочный интервал – 15; кегль – 14; шрифт – TimesNewRoman, обычный; цвет шрифта – черный; поля, не менее:

верхнее - 20 мм; левое – 30 мм;
нижнее - 20 мм; правое – 15 мм.

Иллюстрационно-графический материал в зависимости от специфики программы может включать: чертежи, схемы (технологические, кинематические, гидравлические, автоматизации, алгоритмов и др.), плакаты, диаграммы, макеты, фотографии, аудио- и видеоматериалы, натурные образцы и др.

Иллюстрационно-графический материал может быть представлен на бумажном, электронном или ином виде носителя. Возможно представление иллюстрационно-графического материала в виде брошюр.

Отчет должен быть переплетен доступным способом.

Пример оформления титульного листа отчета о практике

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра «Информационные системы»

№	Дата поступления на кафедру	Подпись отв. за регистрацию	Подпись преподавателя

ОТЧЕТ

об учебной (Технологической (проектно-технологической) практике
(вид практики) (тип практики)

в _____
(наименование организации)

Выполнил _____
(Фамилия И.О. обучающегося)

(шифр группы)

Направление / специальность _____

(код, название)

Руководитель практики от Университета

(должность)

(Фамилия И.О. руководителя)

Севастополь

20 ____ г.

**Перечень основного оборудования и программного обеспечения
специализированных аудиторий и лабораторий для проведения
учебной практики на базе кафедры «Информационные системы»**

№ п/п	Наименование помещений для проведения всех видов учебной деятельности, предусмотренной учебным планом, в том числе помещения для самостоятельной работы	Перечень основного оборудования, учебно-наглядных пособий и используемого программного обеспечения
1	Учебная аудитория Лаборатория геоинформационных систем, 299053, г. Севастополь, ул. Университетская, 33. 24 посадочных места. Место преподавателя (из них 14 компьютеризированных)	Автоматизированные рабочие места (14 мест и место преподавателя): – ПК Tiger V-210 с монитором (Конфигурация 3): системный блок: AMD Ryzen 5 1400\ RAM 16ГБ\SSD NVIDIA GeForce GTX 1050; монитор:ASUS VP247\клавиатура\мышь; – APM GIGANT MT\23 с монитором, системный блок Pro-223 Intel-Core i5 3,2Гц\DDR3 8ГБ 1600МГц\SSD\ монитор LGA 1150\ клавиатура\мышь; – мультимедийный комплекс №5 (интерактивная доска) (стационарный); – проектор NEC M362XG (стационарный); – оборудование для видеоконференций №1 (мобильный); – пульт дистанционного управления для презентаций Logitech (мобильный); <i>Программное обеспечение (лицензионное):</i> – Windows 10 pro, Microsoft Office 2016, Kaspersky Endpoint Security; <i>Программное обеспечение (свободно распространяемое):</i> QGIS, PostgreSQL, OpenServer, GeoServer, SasPlanet, MS SQL, Visual Studio 2017 Community (C#, Unity, XNA), R – Studio, R, Apache, Deductor studio academic, Firebird, Python.
2	Учебная аудитория Лаборатория мультимедийной обработки данных и кроссплатформенного программирования, 299053, г. Севастополь, ул. Университетская, 33. 29 посадочных мест (из них 19 компьютеризированных)	Автоматизированные рабочие места (19 мест и место преподавателя): – Aquarius W60K37 (процессор Intel(R) Core(TM) i5-9400; оперативная память - 16GB; монитор Samsung: диагональ 24 дюйма; ОС - Microsoft Windows 10 Pro 64-bit) – 1 шт.; – DEPO Race VT55 (Процессор Intel(R) Core(TM) i7-9700F; Оперативная память - 16 GB;Монитор : диагональ 24 дюйма; ОС - Microsoft Windows 10 Pro 64-bit) – 10 шт; – APM (Рабочая станция 1 (DeepLearning) (видеокарта - NVIDIA® GeForce® RTX 30xx (CUDA); процессор CPU Intel Core I7; оперативная память - 32GB; накопители: SSD – 1 Tb; монитор: диагональ не менее 24 дюйма; ОС - Microsoft Windows 10 Pro 64-bit)) – 19 шт.; Стационарный мультимедийный комплекс) Пульт дистанционного управления для презентаций Logitech (мобильный)

№ п/п	Наименование помещений для проведения всех видов учебной деятельности, предусмотренной учебным планом, в том числе помещения для самостоятельной работы	Перечень основного оборудования, учебно-наглядных пособий и используемого программного обеспечения
		оборудование для видеоконференций №2 Logitech ConferenceCam CC3000e (мобильный) <i>Программное обеспечение (лицензионное):</i> Windows 10 pro, Microsoft Office 2016, Kaspersky Endpoint Security <i>Программное обеспечение (свободно распространяемое):</i> Visual Studio 2017 Community (C#, Unity, XNA), Dos Box, Eclipse (Lisp, Prolog, C, IDE for Java, JDK (Oracle), Visual Studio Code, Firebird, MySQL6, Android Studio, Python, Scilab, Unreal Engine, Unity.
3	Учебная аудитория Лаборатория получения и обработки данных дистанционного зондирования Земли 29 посадочных мест, место преподавателя; 299053, г. Севастополь, ул. Университетская, 33. Автоматизированные рабочие места (19 мест и место преподавателя):	– Графическая станция ThinkStation P300 TWR – 1 шт.; – АРМ (Рабочая станция 1 (DeepLearning) (видеокарта - NVIDIA® GeForce® RTX 30xx (CUDA); процессор CPU Intel Core I7; оперативная память - 32GB; накопители: SSD – 1 Tb; монитор: диагональ не менее 24 дюйма; ОС - Microsoft Windows 10 Pro 64-bit)) – 19 шт.; – Оборудование для видеоконференций №2 Logitech ConferenceCam CC3000e (960-000983) (мобильный); – Проектор NEC M362XG в комплекте с кабелем HDMI 25м; – Мультимедийное оборудование для компьютерного класса (ист.соб.пос) 3770; – пульт дистанционного управления для презентаций Logitech (мобильный); – Аппаратно-программный комплекс для приема, обработки и распространения данных наблюдения Земли с низкоорбитальных КА в X-диапазоне частот «Унискан-М» в составе: Земная приемная станция; – Устройство обработки и регистрации сигнала («Приемный канал»), совместимое с земной приемной станции УниСкан-М, входящей в состав ПАК; – Программно-аппаратный комплекс инфраструктурной дифференциальной базовой станции; – Доска магнитно-маркерная BRAUBERG Россия. <i>Программное обеспечение (лицензионное):</i> – Windows 10 pro, Microsoft Office 2016, Kaspersky Endpoint Security; – Программное обеспечение (Интерактивная 3D модель земного шара с функциями интеграции геоданных) «Цифровой глобус»; – Программное обеспечение ScanEx Web GeoMixer - функционал доступа к открытым данным космической съемки и продуктам на их основе; – Программное обеспечение для обработки GNSS данных инфраструктурной дифференциальной базовой

№ п/п	Наименование помещений для проведения всех видов учебной деятельности, предусмотренной учебным планом, в том числе помещения для самостоятельной работы	Перечень основного оборудования, учебно-наглядных пособий и используемого программного обеспечения
		станции Trimble Business Center (версия Advanced, USB ключ); Программное обеспечение СКАНЭКС Терминал: Модуль генерации изображений СКАНЭКС Терминал, Модуль веб-интерфейса пользователя СКАНЭКС Терминал и активации обработки данных, принимаемых ПАК с КА серии JPSS
4	Учебная лаборатория систем передачи данных и компьютерных сетей 299053, г. Севастополь ул. Университетская, 33	25 посадочных мест (из них 12 компьютеризированных), рабочее место преподавателя: АРМ: GIGANT MT\23 с монитором (системный блок Pro-223 Intel-Core i5 3,2Гц\ DDR3 8Гб\1600МГц\SSD\монитор LGA 1150\клавиатура\мышь) (6 шт.) ПК: Tiger V-510 (конфигурация 1) (системный блок: Intel Core i5 6400 @ 2.70GHz\ RAM 8,0 0ГБ\SSD \Intel HD Graphics 530; монитор: DELL E2417H\клавиатура\мышь) (6 шт.) Стационарный мультимедийный комплекс (проектор, экран) Оборудование для видеоконференций №2 Logitech ConferenceCam CC3000e (мобильный) Программное обеспечение (лицензионное): Windows 10 pro, Microsoft Office 2016, Kaspersky Endpoint Security, CommView Не включает модуль VoIP TamoSoft, MatLab R2017a —
5	Учебная аудитория Лаборатория объектно-ориентированного программирования и web-технологий 299053, г. Севастополь, ул. Университетская, 33 20 посадочных мест (из них 10 компьютеризированных) Автоматизированное рабочее место (АРМ) с 2мя мониторами (10 шт.)	Автоматизированные рабочие места (10 мест и место преподавателя): — АРМ (Процессор I5-7400; Оперативная память - 8 GB; 2 монитора: диагональ 24 дюйма; ОС - Microsoft Windows 10 Pro 64-bit) – 10 шт; — Aquarius W60K37 (процессор Intel(R) Core(TM) i5-9400; оперативная память - 16GB; монитор Samsung: диагональ 24 дюйма; ОС - Microsoft Windows 10 Pro 64-bit) – 1 шт.; — мультимедийный комплект (тип2) стационарный; — проектор NEC M362XG (стационарный); — оборудование для видеоконференций №1 (мобильный); — пульт дистанционного управления для презентаций Logitech (мобильный); — Доска маркерная 100x200; <i>Программное обеспечение (лицензионное):</i> — Windows 10 pro, Microsoft Office 2016, Kaspersky Endpoint Security; <i>Программное обеспечение (свободно распространяемое):</i> Visual Studio 2017 Community (C#, Unity, XNA), Dos Box, Eclipse (Lisp, Prolog, C, IDE for Java, JDK (Oracle), Visual Studio Code, Firebird, AnyLogic PLE (Personal Learning Edition), MySQL, Gimp, Android Studio, , Python, Scilab

№ п\п	Наименование помещений для проведения всех видов учебной деятельности, предусмотренной учебным планом, в том числе помещения для самостоятельной работы	Перечень основного оборудования, учебно-наглядных пособий и используемого программного обеспечения
		<i>Программное обеспечение</i> (лицензионное): Windows 10 pro, Microsoft Office 2016, Kaspersky Endpoint Security, CommView Не включает модуль VoIP TamoSoft, MatLab R2017a <i>Программное обеспечение</i> (свободно распространяемое): Cisco Packet Tracer Student, Work Bench, Proteus, MySQL, OS Linux, Linux CentOS 6.0, VirtualBox

ПРИЛОЖЕНИЕ А

Варианты задания, описанного в п. 8.2.1 – Описание структуры процессора, системы команд и способов кодирования

Структура 1 (рисунок А.1). Описание и система команд (таблица А.1)

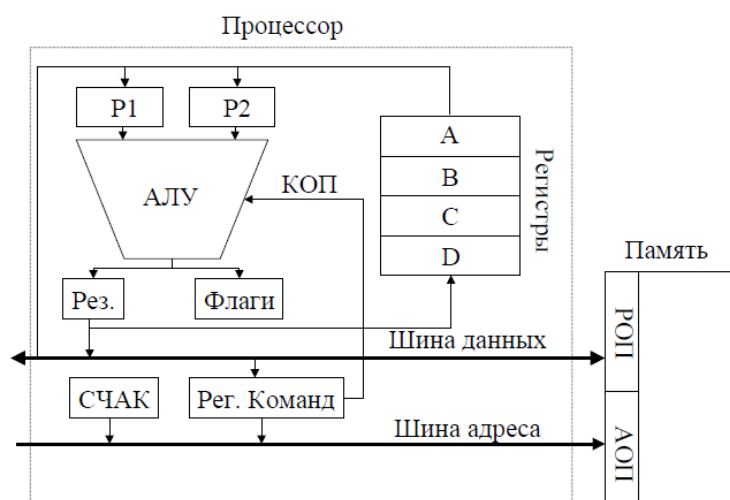


Рисунок А.1 – Структура 1

Объем оперативной памяти – 256 байт.

Длина команд – один или два байта. В первом байте биты 0..3 являются кодом операции, а следующие интерпретируются в зависимости от принадлежности команд к той или иной группе:

- для арифметических и логических команд содержат номера РОН (0-А, 1-В, 2-С, 3-Д), над которыми производятся операции;
- для команд пересылки информации – номер регистра, с которым осуществляется операция;
- не используются для команд передачи управления и управления процессом.

Второй байт используется для задания адреса памяти, где находится операнд, адреса перехода для команд передачи управления или константы.

Таблица А.1 – Система команд для структуры 1

Код	Мнемокод и операнды	Описание
0000	ADD регистр1,регистр2	регистр1=регистр1+регистр2
0001	ADD регистр, память	регистр=регистр+память
0010	AND регистр1,регистр2	регистр1=регистр1&регистр2
0011	OR регистр1,регистр2	регистр1=регистр1 регистр2
0100	NOT регистр	инверсия битов регистра
0101	MOV регистр,константа	занести константу в регистр

0110	LOAD регистр, адрес	загрузка данных в регистр из памяти
0111	STORE регистр, адрес	запись данных из регистра в память
1000	JMP адрес	безусловный переход
1001	JZ адрес	переход, если 0
1010	JO адрес	переход, если переполнение
1011	JL адрес	переход, если меньше
1100	JNZC регистр, адрес	переход, если не ноль, декремент регистра
1101	CLF	установка регистра флагов в 0
1110	CALL адрес	переход к подпрограмме, адрес возврата в регистре D
1111	RET	возврат из подпрограммы

Структура 2 (рисунок А.2). Описание и система команд (таблица А.2)

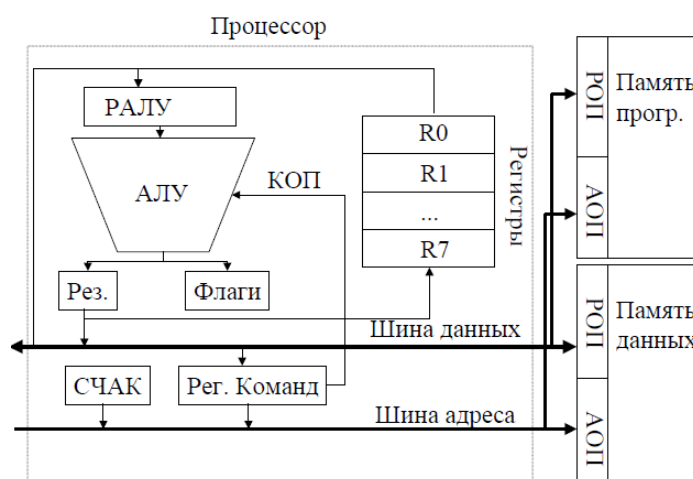


Рисунок А.2 – Структура 2

Объем оперативной памяти данных – 256 байтов, объем оперативной памяти программ – 256 байтов. Данные и программы хранятся отдельно.

Таблица А.2 – Система команд для структуры 2

...бит определяет код команды в соответствии с таблицей команд (бит 0-4), а также номер регистра-операнда (биты 5-7), указывающего на один из РОНов процессора.

Второй байт определяет адрес операнда в памяти, или задает константу.

Арифметические и логические операции выполняются над операндом, находящимся в памяти

или регистре, и результатом предыдущей операции, который находится в регистре результатов. Флаг переполнения учитывается как единица младшего разряда, если используется в арифметических командах.

Код	Мнемокод и операнды	Описание
00000	ADD PОН	рез=PОН+рез
00001	ADD адрес	рез=память[адрес]+рез
00010	ADC PОН	рез=PОН+рез+флаг переполнения
00011	SUB PОН	рез=рез-PОН
00100	SUB адрес	рез=рез-память[адрес]
00101	SBB PОН	рез=рез-PОН-флаг переполнения
00110	AND PОН	рез=PОН&рез
00111	AND адрес	рез=память[адрес]&рез
01000	OR PОН	рез=PОН рез.
01001	OR адрес	рез=память[адрес] рез
01010	XOR PОН	рез=PОН⊗рез.
01011	XOR адрес	рез=память[адрес]⊗рез
01100	MUL PОН	рез=рез*PОН
01101	DIV PОН	рез=рез/PОН (деление нацело, остаток – в PОН+1)
01110	LDR PОН	рез=PОН
01111	SVR PОН	PОН=рез
10000	CALL PОН,адрес	переход к подпрограмме; адрес возврата сохраняется в PОНе;
10001	RET PОН	возврат из подпрограммы; адрес возврата находится в PОНе
10010	INC PОН	увеличение PОНа на 1
10011	DEC PОН	уменьшение PОНа на 1
10100	MOV PОН,константа	занести в PОН константу
10101	MOV PОН,адрес	PОН=память[адрес]
10110	MOV память,PОН	память[адрес]=PОН
10111	LOAD PОН	рез=память[адрес=PОН]
11000	SAVE PОН	память[адрес=PОН]=рез
11001	JMP адрес	безусловный переход
11010	JZ адрес	переход, если 0
11011	JNZ адрес	переход, если не 0
11110	HALT	останов
11111	CLR	установка регистра результата в 0

Структура 3 (рисунок А.3).
Описание и система команд
(таблица А.3)

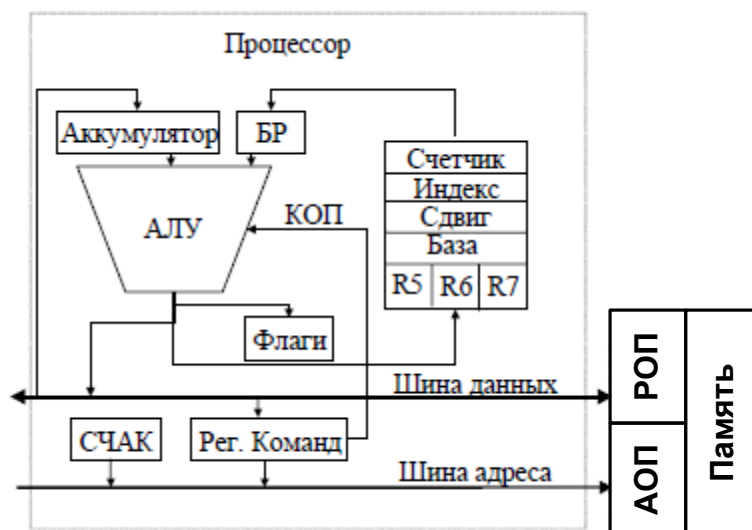


Рисунок А.3 – Структура 3

Объем оперативной памяти – 256 байт. Микропроцессор имеет 8 регистров, 5 из них – специального назначения:

РОН0 (AC). аккумулятор (в него заносятся результаты выполнения всех операций);

РОН 1 (CNT) . счетчик (для операций увеличения (уменьшения) на 1);

РОН 2 (SH) . сдвиговый регистр (для выполнения команд сдвига);

РОН 3 (BASE) . регистр базы (содержит базовый адрес операнда в ОП);

РОН 4 (IND) . регистр индекса (для формирования смещения операнда в ОП).

Таблица А.3 – Система команд для структуры 3

Код	Мнемокод	Описание
0000	ADD регистр	аккумулятор= аккумулятор + регистр
0001	ADC регистр	аккумулятор= аккумулятор + регистр + флаг переполнения
0010	SUB регистр	аккумулятор= аккумулятор - регистр
0011	SBB регистр	аккумулятор= аккумулятор - регистр - флаг переполнения
0100	AND регистр	аккумулятор=аккумулятор & регистр
0101	OR регистр	аккумулятор=аккумулятор регистр
0110	NOT	аккумулятор=не аккумулятор
0111	MOV регистр, константа	регистр=константа
1000	STOS	память[BASE+IND]= аккумулятор IND=IND+1
1000	LODS*	Аккумулятор =память[BASE+IND], IND=IND+1
1001	OUTR регистр, адрес	память[адрес]= регистр
1001	INR* регистр, адрес	регистр= память[адрес]
1010	LOOP адрес	переход по адресу, если CNT>0, CNT=CNT-1
1011	JNZ адрес	переход, если не 0
1100	JB адрес	переход, если результат меньше 0
1101	SWAP регистр	Обмен данными между аккумулятором и регистром
1110	SHR	Сдвиг вправо регистра SH на число разрядов в регистре CNT
1111	SHL	Сдвиг влево регистра SH на число разрядов в регистре CNT

Коды команд:

Команды имеют длину один или два байта. Биты 0-3 первого байта определяют код команды в соответствии с таблицей команд. Номер регистра хранится в битах 4-6 первого байта. Бит 7 указывает направление перемещения данных: 0 означает перемещение данных в оперативную память, в остальных случаях он равен 1. Второй байт определяет константу или адреса памяти.

Арифметические и логические операции выполняются над операндами, находящимися в аккумуляторе и (или) регистре.

Структура 4 (рисунок А.4). Описание и система команд (таблица А.4)

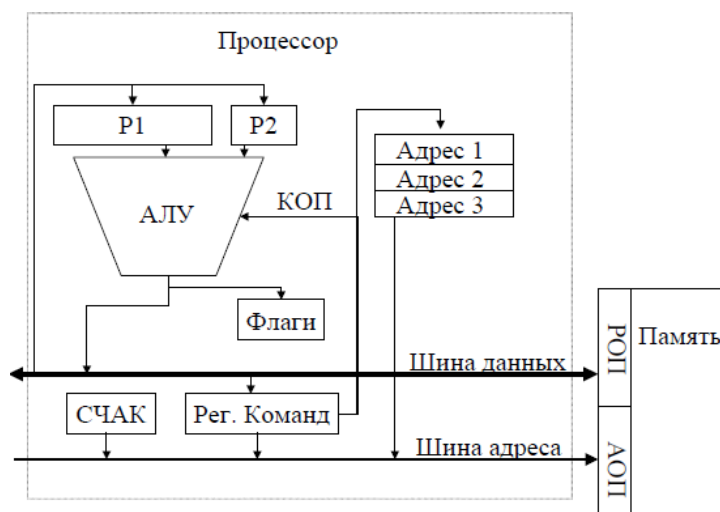


Рисунок А.4 – Структура 4

Объем оперативной памяти . 2048 байт (8 страниц по 256 байт).

Регистр СЧАК имеет 12 битов, но старший бит не используется.

Микропроцессор имеет три адресных регистра. Регистры АДРЕС1 (A1) и АДРЕС2 (A2) содержат адреса операндов команд, регистр АДРЕС3 (A3) . адрес результата операции. Эти регистры имеют по 8 битов и определяют смещение в диапазоне 0-255 внутри страницы. Номер страницы (целое число от 0 до 7) задан в битах 5-7 регистра команд.

Команды имеют длину один или два байта. Биты 0-4 первого байта определяют код команды в соответствии с таблицей команд. Биты 5-7 первого байта задают номер страницы ОП, в которой располагаются данные. Второй байт определяет константу.

Таблица А.4 – Система команд для структуры 4

Код	Мнемокод	Описание
00000	MOVA1 константа	A1=константа
00001	MOVA2 константа	A2=константа
00010	MOVA3 константа	A3=константа
00100	ADD страница	Память[A3]= Память[A1]+ Память[A2]
00110	SUB страница	Память[A3]= Память[A1]- Память[A2]
00111	SBB страница	Память[A3]= Память[A1]- Память[A2] -флаг переп.
01000	MUL страница	Память[A3]= Память[A1]* Память[A2]
01001	DIV страница	Память[A3]= Память[A1]/ Память[A2]
01010	MOD страница	Память[A3]= Память[A1]% Память[A2]
01011	ABS страница	Память[A3]= Память[A1] **
01100	AND страница	Память[A3]=Память[A1]& Память[A2]
01101	OR страница	Память[A3]= Память[A1] Память[A2]
01110	XOR страница	Память[A3]= Память[A1] ^ Память[A2]
01111	NOT страница	Память[A3]=not Память[A1]
10000	JMP страница	Переход по адресу A3
10001	JB страница	Переход по адресу A3, если результат меньше 0
10010	JNZ страница	Переход по адресу A3, если результат не равен 0
10101	MOVS	Память[A3]=Память[A1], A3+=1, A1+=1
10101	CMPS	Сравнение данных Память[A1] и Память[A3] ***

** отрицательные числа представляются в дополнительном коде. Для нахождения модуля необходимо выполнить перевод в прямой код и обнулить старший бит

*** выполняется аналогично операции вычитания, но результат операции не заносится в память

Структура 5 (рисунок А.5). Описание и система команд (таблица А.5)

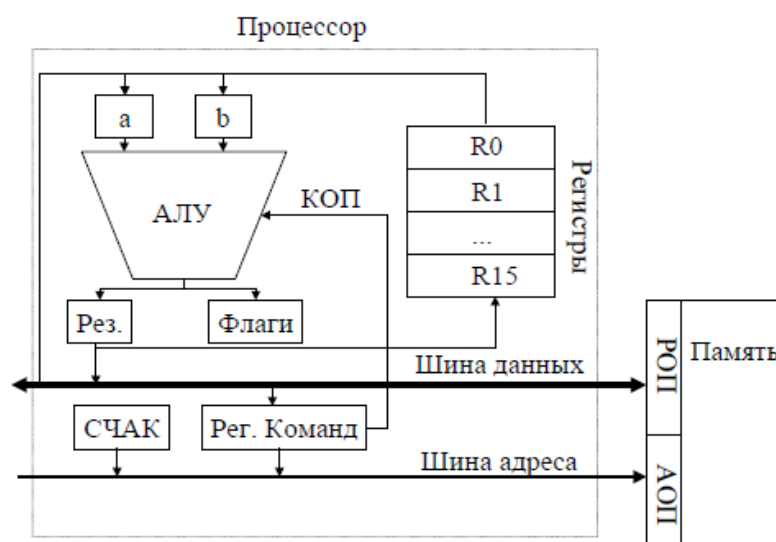


Рисунок А.5 – Структура 5

Объем оперативной памяти . 256 байт. Буферные регистры а и b предназначены для хранения операндов операций. Рабочий регистр Рез используется как регистр результата, полученного при обработке данных в АЛУ.

Коды команд:

Команды имеют длину один или два байта. Первый байт определяет код команды в соответствии с таблицей команд (биты 0-3). Следующие 4 бита предназначены для указания номера регистра, коммутируемого к буферному регистру а.

Второй байт используется для задания константы или адреса памяти, где находится операнд. Если в качестве второго операнда используется регистр, то его номер хранится в младших 4-х битах второго байта команды, а старшие 4 бита не используются.

Таблица А.5 – Система команд для структуры 5

Код	Мнемокод	Описание
0000	ADNAB регистр.регистр	$\bar{a} + b$
0001	ADANB регистр.регистр	$a + \bar{b}$
0010	ADAB регистр.регистр	$a + b$
0011	ANBONAB регистр.регистр	$a\bar{b} \vee \bar{a}b$
0100	ABONANB регистр.регистр	$ab \vee \bar{a}\bar{b}$
0101	NAB регистр.регистр	$\bar{a}b$
0110	ANB регистр.регистр	$a\bar{b}$
0111	JZ адрес	Переход, если результат равен 0

1001	JL адрес	Переход, если результат меньше 0
1010	JMP адрес	Безусловный переход
1100	MOVC регистр, константа	Регистр:=Константа
1101	MOVR регистр, адрес	Регистр:= Память[адрес]
1110	MOVM регистр, адрес	Память[адрес]:=регистр
1111	CMP регистр, регистр	Сравнение двух регистров. Установка флагов.

Детализация описания мини-проектов, представленных в п. 8.2.2

1. Калькулятор

Реализовать простейший калькулятор, который дает возможность пользователю складывать, умножать, вычитать, делить числа, находить их квадратный корень.

КАК пользователь

КОГДА я ввожу два числа

ТОГДА я увижу сумму двух чисел

КАК пользователь

КОГДА я ввожу два числа

ТОГДА я могу увидеть разность двух чисел

КАК пользователь

КОГДА я ввожу два числа

ТОГДА я увижу произведение двух чисел

КАК пользователь

КОГДА я ввожу два числа

И первое число является числителем

И второе является знаменателем и *не равно* нулю

ТОГДА я увижу частное двух чисел

КАК пользователь

КОГДА я ввожу два числа

И первое число является числителем

И второе является знаменателем и *равно* нулю

ТОГДА я должен увидеть ошибку

2. Генератор паролей

Создайте генератор случайных паролей для своих друзей и семьи, чтобы обезопасить их учётные записи!

Реализовать простейший генератор паролей, который дает возможность пользователю получить пароль, удовлетворяющий его требованиям.

КАК пользователь

КОГДА я открываю программу

И указываю количество обычных символов

ТОГДА я вижу сгенерированный пароль
И этот пароль содержит выбранное количество обычных символов

КАК пользователь
КОГДА я открываю программу
И указываю количество специальных символов
ТОГДА я вижу сгенерированный пароль
И этот пароль содержит выбранное количество специальных символов

КАК пользователь
КОГДА я открываю программу
И указываю количество букв в верхнем регистре
ТОГДА я вижу сгенерированный пароль
И этот пароль содержит выбранное количество букв в верхнем регистре

КАК пользователь
КОГДА я открываю программу
И указываю различные критерии(разное число обычных, специальных, в верхнем регистре символов и цифр)
ТОГДА я вижу сгенерированный пароль
И этот пароль должен удовлетворять *всем* требованиям

3. Игра-викторина

Приложения-викторины задают пользователям серию вопросов и дают им возможность ответить. Затем викторина даёт пользователю результаты. Реализовать программу-викторину на любую интересную вам тему. Да, можно про Наруто. Да, можно не про аниме. На любую тему :3

КАК пользователь
КОГДА я открываю программу
ТОГДА программа *предлагает мне сыграть в игру (с)* и ответить на 10 вопросов

КАК пользователь
КОГДА я не соглашаюсь сыграть в игру
ТОГДА программа благодарит за участие и завершается

КАК пользователь
КОГДА я соглашаюсь сыграть в игру
ТОГДА программа задает мне вопрос и дает возможность ответить

КАК пользователь
КОГДА я даю ответ на вопрос
ТОГДА программа задает мне следующий вопрос

КАК пользователь
КОГДА я даю ответ на последний (десятый) вопрос
ТОГДА программа показывает результаты(сколько очков заработано)
И программа предлагает сыграть снова

4. Игра "Угадай число"

В данном мини-проекте вам необходимо реализовать игру, в которой компьютер пытается угадать целое число от 1 до 1000. Пользователь загадывает число, а программа задает ему вопросы из разряда *Да* или *Нет* в попытках угадать число. Если программа угадает число, она спрашивает у пользователя, это ли число она угадала. Если да - то урааа! Если нет - программа показывает грустный смайлик и предлагает новую игру. *Предполагается, что пользователь играет и отвечает на вопросы честно, а не как всегда.*

КАК пользователь

КОГДА я открываю программу

ТОГДА программа *предлагает мне сыграть в игру (с)*

КАК пользователь

КОГДА я не соглашаюсь сыграть в игру

ТОГДА программа благодарит за участие и завершается

КАК пользователь

КОГДА я соглашаюсь сыграть в игру

ТОГДА программа предлагает мне загадать число от 1 до 1000

И после этого начинает задавать вопросы из-разряда *Да/Нет*

КАК пользователь

КОГДА я даю ответ на вопрос *Да*

ТОГДА программа задает мне следующий вопрос

КАК пользователь

КОГДА я даю ответ на вопрос *Нет*

ТОГДА программа задает мне следующий вопрос

КАК пользователь

КОГДА я даю ответ на вопрос, который не является *Да* или *Нет*

ТОГДА программа просит ввести ответ снова, так как у нее лапки

КАК пользователь

КОГДА я даю ответ на вопрос

И программа предполагает что угадала число

ТОГДА программа спрашивает - такое ли число было загадано

И предлагает дать ответ на вопрос *Да* или *Нет*

КАК пользователь

КОГДА я даю ответ на вопрос

И программа угадала число

ТОГДА программа показывает результаты - как быстро она угадала, сколько шагов и какое число было загадано

И программа предлагает сыграть снова

КАК пользователь

КОГДА я даю ответ на вопрос

И программа не число
ТОГДА программа показывает грустный смайлик
И программа предлагает сыграть снова

5. Программа-шифровщик

В данном мини-проекте вам необходимо реализовать программу, которая позволяет как шифровать сообщение, так и дешифровать сообщение. Предполагается, что обладая знанием ключа/ключей, вы можете зашифровать сообщение, а ваш друг - расшифровать, и наоборот.

КАК пользователь
КОГДА я открываю программу
ТОГДА программа предлагает либо зашифровать текст, либо дешифровать текст

КАК пользователь
КОГДА я ввожу текст для шифрования
И я ввожу ключ для шифрования
ТОГДА программа показывает зашифрованный текст

КАК пользователь
КОГДА я ввожу текст для дешифрования
И я ввожу ключ для дешифрования
ТОГДА программа показывает расшифрованный текст

ПРИЛОЖЕНИЕ А

Варианты задания - Описание структуры процессора, системы команд и способов кодирования

Структура 1 (рисунок А.1). Описание и система команд (таблица А.1)

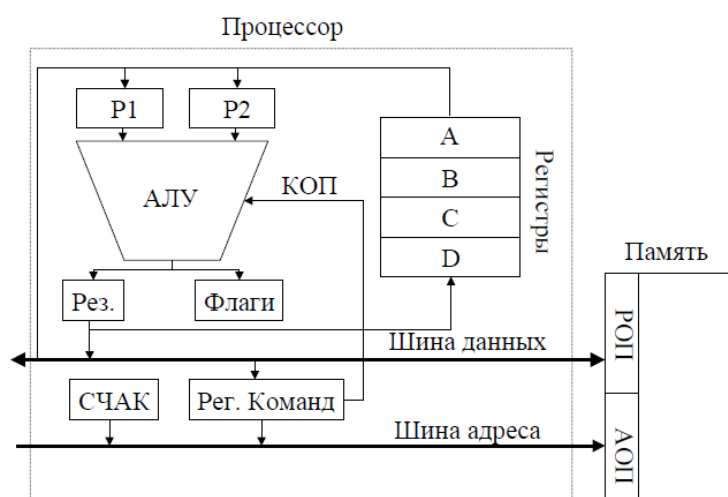


Рисунок А.1 – Структура 1

Объем оперативной памяти – 256 байт.

Длина команд – один или два байта. В первом байте биты 0..3 являются кодом операции, а следующие интерпретируются в зависимости от принадлежности команд к той или иной группе:

- для арифметических и логических команд содержат номера РОН (0-А, 1-В, 2-С, 3-Д), над которыми производятся операции;
- для команд пересылки информации – номер регистра, с которым осуществляется операция;
- не используются для команд передачи управления и управления процессом.

Второй байт используется для задания адреса памяти, где находится операнд, адреса перехода для команд передачи управления или константы.

Таблица А.1 – Система команд для структуры 1

Код	Мнемокод и операнды	Описание
0000	ADD регистр1,регистр2	регистр1=регистр1+регистр2
0001	ADD регистр, память	регистр=регистр+память
0010	AND регистр1,регистр2	регистр1=регистр1&регистр2
0011	OR регистр1,регистр2	регистр1=регистр1 регистр2
0100	NOT регистр	инверсия битов регистра
0101	MOV регистр,константа	занести константу в регистр
0110	LOAD регистр, адрес	загрузка данных в регистр из памяти
0111	STORE регистр, адрес	запись данных из регистра в память
1000	JMP адрес	безусловный переход
1001	JZ адрес	переход, если 0
1010	JO адрес	переход, если переполнение
1011	JL адрес	переход, если меньше
1100	JNZC регистр, адрес	переход, если не ноль, декремент регистра
1101	CLF	установка регистра флагов в 0
1110	CALL адрес	переход к подпрограмме, адрес возврата в регистре D
1111	RET	возврат из подпрограммы

Структура 2 (рисунок А.2). Описание и система команд (таблица А.2)

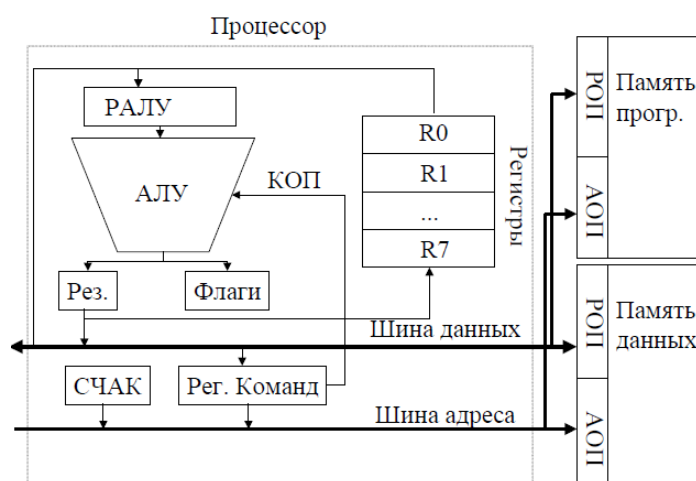


Рисунок А.2 – Структура 2

Объем оперативной памяти данных – 256 байтов, объем оперативной памяти программ – 256 байтов. Данные и программы хранятся отдельно.

Таблица А.2 – Система команд для структуры 2

Команды имеют длину один или два байта. Первый байт определяет код команды в соответствии с таблицей команд (бит 0-4), а также номер регистра-операнда (биты 5-7), указывающего на один из РОНов процессора.

Второй байт определяет адрес операнда в памяти, или задает константу.

Арифметические и логические операции выполняются над операндом, находящимся в памяти

или регистре, и результатом предыдущей операции, который находится в регистре результатов. Флаг переполнения учитывается как единица младшего разряда, если используется в арифметических командах.

Код	Мнемокод и операнды	Описание
00000	ADD РОН	рез=РОН+рез
00001	ADD адрес	рез=память[адрес]+рез
00010	ADC РОН	рез=РОН+рез+флаг переполнения
00011	SUB РОН	рез=рез-РОН
00100	SUB адрес	рез=рез-память[адрес]
00101	SBB РОН	рез=рез-РОН-флаг переполнения
00110	AND РОН	рез=РОН&рез
00111	AND адрес	рез=память[адрес]&рез
01000	OR РОН	рез=РОН рез.
01001	OR адрес	рез=память[адрес] рез
01010	XOR РОН	рез=РОН⊗рез.
01011	XOR адрес	рез=память[адрес]⊗рез
01100	MUL РОН	рез=рез*РОН
01101	DIV РОН	рез=рез/РОН (деление нацело, остаток – в РОН+1)
01110	LDR РОН	рез=РОН
01111	SVR РОН	РОН=рез
10000	CALL РОН,адрес	переход к подпрограмме; адрес возврата сохраняется в РОНе;
10001	RET РОН	возврат из подпрограммы; адрес возврата находится в РОНе
10010	INC РОН	увеличение РОНа на 1
10011	DEC РОН	уменьшение РОНа на 1
10100	MOV РОН,константа	занести в РОН константу
10101	MOV РОН,адрес	РОН= память[адрес]
10110	MOV память,РОН	память[адрес]=РОН
10111	LOAD РОН	рез=память[адрес=РОН]
11000	SAVE РОН	память[адрес=РОН]=рез
11001	JMP адрес	безусловный переход
11010	JZ адрес	переход, если 0
11011	JNZ адрес	переход, если не 0
11110	HALT	останов
11111	CLR	установка регистра результата в 0

Структура 3 (рисунок А.3).
Описание и система команд
(таблица А.3)

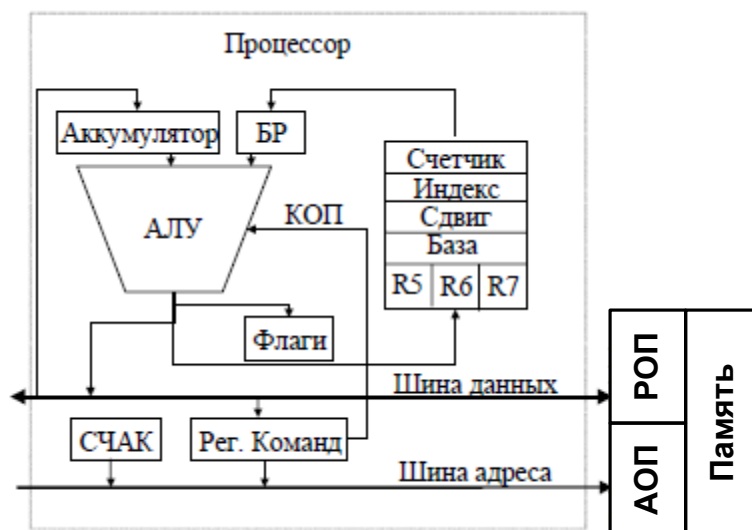


Рисунок А.3 – Структура 3

Объем оперативной памяти – 256 байт. Микропроцессор имеет 8 регистров, 5 из них – специального назначения:

РОН0 (AC). аккумулятор (в него заносятся результаты выполнения всех операций);

РОН 1 (CNT) . счетчик (для операций увеличения (уменьшения) на 1);

РОН 2 (SH) . сдвиговый регистр (для выполнения команд сдвига);

РОН 3 (BASE) . регистр базы (содержит базовый адрес операнда в ОП);

РОН 4 (IND) . регистр индекса (для формирования смещения операнда в ОП).

Таблица А.3 – Система команд для структуры 3

Код	Мнемокод	Описание
0000	ADD регистр	аккумулятор= аккумулятор + регистр
0001	ADC регистр	аккумулятор= аккумулятор + регистр + флаг переполнения
0010	SUB регистр	аккумулятор= аккумулятор - регистр
0011	SBB регистр	аккумулятор= аккумулятор - регистр - флаг переполнения
0100	AND регистр	аккумулятор=аккумулятор & регистр
0101	OR регистр	аккумулятор=аккумулятор регистр
0110	NOT	аккумулятор=не аккумулятор
0111	MOV регистр, константа	регистр=константа
1000	STOS	память[BASE+IND]= аккумулятор IND=IND+1
1000	LODS*	Аккумулятор =память[BASE+IND], IND=IND+1
1001	OUTR регистр, адрес	память[адрес]= регистр
1001	INR* регистр, адрес	регистр= память[адрес]
1010	LOOP адрес	переход по адресу, если CNT>0, CNT=CNT-1
1011	JNZ адрес	переход, если не 0
1100	JB адрес	переход, если результат меньше 0
1101	SWAP регистр	Обмен данными между аккумулятором и регистром
1110	SHR	Сдвиг вправо регистра SH на число разрядов в регистре CNT
1111	SHL	Сдвиг влево регистра SH на число разрядов в регистре CNT

Коды команд:

Команды имеют длину один или два байта. Биты 0-3 первого байта определяют код команды в соответствии с таблицей команд. Номер регистра хранится в битах 4-6 первого байта. Бит 7 указывает направление перемещения данных: 0 означает перемещение данных в оперативную память, в остальных случаях он равен 1. Второй байт определяет константу или адреса памяти.

Арифметические и логические операции выполняются над операндами, находящимися в аккумуляторе и (или) регистре.

Структура 4 (рисунок А.4). Описание и система команд (таблица А.4)

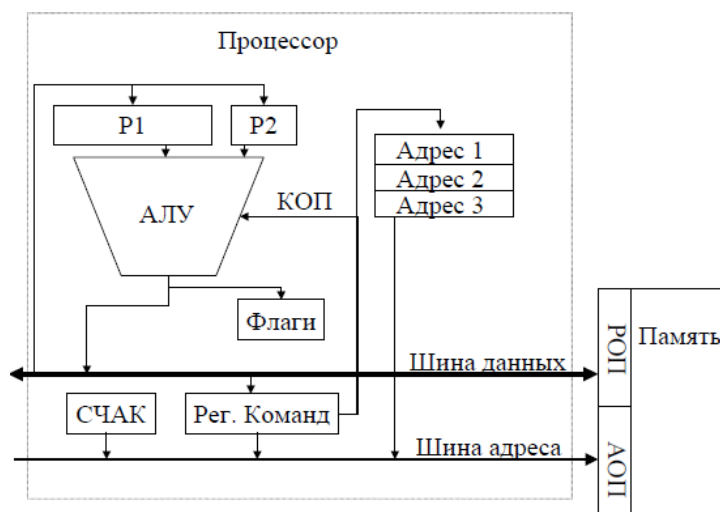


Рисунок А.4 – Структура 4

Объем оперативной памяти . 2048 байт (8 страниц по 256 байт).

Регистр СЧАК имеет 12 битов, но старший бит не используется.

Микропроцессор имеет три адресных регистра. Регистры АДРЕС1 (A1) и АДРЕС2 (A2) содержат адреса операндов команд, регистр АДРЕС3 (A3) . адрес результата операции. Эти регистры имеют по 8 битов и определяют смещение в диапазоне 0-255 внутри страницы. Номер страницы (целое число от 0 до 7) задан в битах 5-7 регистра команд.

Команды имеют длину один или два байта. Биты 0-4 первого байта определяют код команды в соответствии с таблицей команд. Биты 5-7 первого байта задают номер страницы ОП, в которой располагаются данные. Второй байт определяет константу.

Таблица А.4 – Система команд для структуры 4

Код	Мнемокод	Описание
00000	MOVA1 константа	A1=константа
00001	MOVA2 константа	A2=константа
00010	MOVA3 константа	A3=константа
00100	ADD страница	Память[A3]= Память[A1]+ Память[A2]
00110	SUB страница	Память[A3]= Память[A1]- Память[A2]
00111	SBB страница	Память[A3]= Память[A1]- Память[A2] -флаг переп.
01000	MUL страница	Память[A3]= Память[A1]* Память[A2]
01001	DIV страница	Память[A3]= Память[A1]/ Память[A2]
01010	MOD страница	Память[A3]= Память[A1]% Память[A2]
01011	ABS страница	Память[A3]= Память[A1] **
01100	AND страница	Память[A3]=Память[A1]& Память[A2]
01101	OR страница	Память[A3]= Память[A1] Память[A2]
01110	XOR страница	Память[A3]= Память[A1] ^ Память[A2]
01111	NOT страница	Память[A3]=not Память[A1]
10000	JMP страница	Переход по адресу A3
10001	JB страница	Переход по адресу A3, если результат меньше 0
10010	JNZ страница	Переход по адресу A3, если результат не равен 0
10101	MOVS	Память[A3]=Память[A1], A3+=1, A1+=1
10101	CMPS	Сравнение данных Память[A1] и Память[A3] ***

** отрицательные числа представляются в дополнительном коде. Для нахождения модуля необходимо выполнить перевод в прямой код и обнулить старший бит

*** выполняется аналогично операции вычитания, но результат операции не заносится в память

Структура 5 (рисунок А.5). Описание и система команд (таблица А.5)

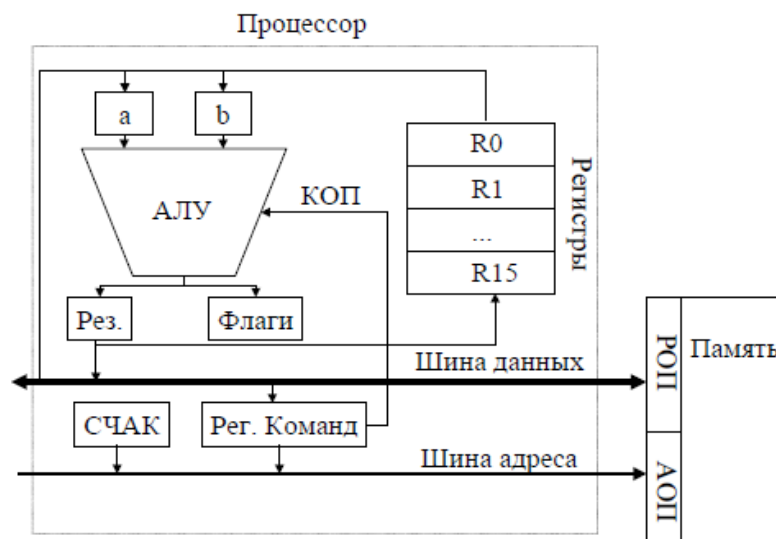


Рисунок А.5 – Структура 5

Объем оперативной памяти . 256 байт. Буферные регистры а и b предназначены для хранения операндов операций. Рабочий регистр Рез используется как регистр результата, полученного при обработке данных в АЛУ.

Коды команд:

Команды имеют длину один или два байта. Первый байт определяет код команды в соответствии с таблицей команд (биты 0-3). Следующие 4 бита предназначены для указания номера регистра, коммутируемого к буферному регистру а.

Второй байт используется для задания константы или адреса памяти, где находится операнд. Если в качестве второго операнда используется регистр, то его номер хранится в младших 4-х битах второго байта команды, а старшие 4 бита не используются.

Таблица А.5 – Система команд для структуры 5

Код	Мнемокод	Описание
0000	ADNAB регистр.регистр	$\bar{a} + b$
0001	ADANB регистр.регистр	$a + \bar{b}$
0010	ADAB регистр.регистр	$a + b$
0011	ANBONAB регистр.регистр	$a\bar{b} \vee \bar{a}b$
0100	ABONANB регистр.регистр	$ab \vee a\bar{b}$
0101	NAB регистр.регистр	$\bar{a}b$
0110	ANB регистр.регистр	$a\bar{b}$
0111	JZ адрес	Переход, если результат равен 0

1001	JL адрес	Переход, если результат меньше 0
1010	JMP адрес	Безусловный переход
1100	MOVC регистр, константа	Регистр:=Константа
1101	MOVR регистр, адрес	Регистр:= Память[адрес]
1110	MOVM регистр, адрес	Память[адрес]:=регистр
1111	CMP регистр, регистр	Сравнение двух регистров. Установка флагов.

Детализация описания мини-проектов

1. Калькулятор

Реализовать простейший калькулятор, который дает возможность пользователю складывать, умножать, вычитать, делить числа, находить их квадратный корень.

КАК пользователь

КОГДА я ввожу два числа

ТОГДА я увижу сумму двух чисел

КАК пользователь

КОГДА я ввожу два числа

ТОГДА я могу увидеть разность двух чисел

КАК пользователь

КОГДА я ввожу два числа

ТОГДА я увижу произведение двух чисел

КАК пользователь

КОГДА я ввожу два числа

И первое число является числителем

И второе является знаменателем и *не равно* нулю

ТОГДА я увижу частное двух чисел

КАК пользователь

КОГДА я ввожу два числа

И первое число является числителем

И второе является знаменателем и *равно* нулю

ТОГДА я должен увидеть ошибку

2. Генератор паролей

Создайте генератор случайных паролей для своих друзей и семьи, чтобы обезопасить их учётные записи!

Реализовать простейший генератор паролей, который дает возможность пользователю получить пароль, удовлетворяющий его требованиям.

КАК пользователь

КОГДА я открываю программу

И указываю количество обычных символов

ТОГДА я вижу сгенерированный пароль

И этот пароль содержит выбранное количество обычных символов

КАК пользователь

КОГДА я открываю программу

И указываю количество специальных символов
ТОГДА я вижу сгенерированный пароль
И этот пароль содержит выбранное количество специальных символов

КАК пользователь
КОГДА я открываю программу
И указываю количество букв в верхнем регистре
ТОГДА я вижу сгенерированный пароль
И этот пароль содержит выбранное количество букв в верхнем регистре

КАК пользователь
КОГДА я открываю программу
И указываю различные критерии(разное число обычных, специальных, в верхнем регистре символов и цифр)
ТОГДА я вижу сгенерированный пароль
И этот пароль должен удовлетворять *всем* требованиям

3. Игра-викторина

Приложения-викторины задают пользователям серию вопросов и дают им возможность ответить. Затем викторина даёт пользователю результаты. Реализовать программу-викторину на любую интересную вам тему. Да, можно про Наруто. Да, можно не про аниме. На любую тему :3

КАК пользователь
КОГДА я открываю программу
ТОГДА программа *предлагает мне сыграть в игру (с)* и ответить на 10 вопросов

КАК пользователь
КОГДА я не соглашаюсь сыграть в игру
ТОГДА программа благодарит за участие и завершается

КАК пользователь
КОГДА я соглашаюсь сыграть в игру
ТОГДА программа задает мне вопрос и дает возможность ответить

КАК пользователь
КОГДА я даю ответ на вопрос
ТОГДА программа задает мне следующий вопрос

КАК пользователь
КОГДА я даю ответ на последний (десятый) вопрос
ТОГДА программа показывает результаты(сколько очков заработано)
И программа предлагает сыграть снова

4. Игра "Угадай число"

В данном мини-проекте вам необходимо реализовать игру, в которой компьютер пытается угадать целое число от 1 до 1000. Пользователь загадывает число, а программа задает ему вопросы из разряда *Да* или *Нет* в попытках угадать число. Если программа угадает число, она спрашивает у пользователя, это ли число она угадала. Если да - то урааа! Если нет - программа показывает грустный смайлик и предлагает новую игру. *Предполагается, что пользователь играет и отвечает на вопросы честно, а не как всегда.*

КАК пользователь

КОГДА я открываю программу

ТОГДА программа *предлагает мне сыграть в игру (с)*

КАК пользователь

КОГДА я не соглашаюсь сыграть в игру

ТОГДА программа благодарит за участие и завершается

КАК пользователь

КОГДА я соглашаюсь сыграть в игру

ТОГДА программа предлагает мне загадать число от 1 до 1000

И после этого начинает задавать вопросы из-разряда *Да/Нет*

КАК пользователь

КОГДА я даю ответ на вопрос *Да*

ТОГДА программа задает мне следующий вопрос

КАК пользователь

КОГДА я даю ответ на вопрос *Нет*

ТОГДА программа задает мне следующий вопрос

КАК пользователь

КОГДА я даю ответ на вопрос, который не является *Да* или *Нет*

ТОГДА программа просит ввести ответ снова, так как *у нее лапки*

КАК пользователь

КОГДА я даю ответ на вопрос

И программа предполагает что угадала число

ТОГДА программа спрашивает - такое ли число было загадано

И предлагает дать ответ на вопрос *Да* или *Нет*

КАК пользователь

КОГДА я даю ответ на вопрос

И программа угадала число

ТОГДА программа показывает результаты - как быстро она угадала, сколько шагов и какое число было загадано

И программа предлагает сыграть снова

КАК пользователь

КОГДА я даю ответ на вопрос

И программа не число

ТОГДА программа показывает грустный смайлик
И программа предлагает сыграть снова

5. Программа-шифровщик

В данном мини-проекте вам необходимо реализовать программу, которая позволяет как шифровать сообщение, так и дешифровать сообщение. Предполагается, что обладая знанием ключа/ключей, вы можете зашифровать сообщение, а ваш друг - расшифровать, и наоборот.

КАК пользователь

КОГДА я открываю программу

ТОГДА программа предлагает либо зашифровать текст, либо дешифровать текст

КАК пользователь

КОГДА я ввожу текст для шифрования

И я ввожу ключ для шифрования

ТОГДА программа показывает зашифрованный текст

КАК пользователь

КОГДА я ввожу текст для дешифрования

И я ввожу ключ для дешифрования

ТОГДА программа показывает расшифрованный текст

Приложение В

Варианты заданий для мини-проекта «Интеллектуальная матрица»

Вариант 1.

Разработать класс «Интеллектуальная матрица» (IntellectMatr).

Поля: 1) ссылка на прямоугольную (квадратную) матрицу (public), тип элементов матрицы – float.

Методы:

- 1) конструктор (параметр – ссылка на матрицу);
- 2) процедура вывода матрицы в окно терминала (без параметров);
- 3) функция, возвращающая значение суммы элементов главной диагонали матрицы (без параметров);
- 4) функция, возвращающая значение определителя матрицы (без параметров);
- 5) функция, возвращающая адрес вектора (одномерного массива), *i*-ый элемент которого равен количеству отрицательных элементов в *i*-ой строке матрицы (без параметров);
- 6) функция, возвращающая адрес транспонированной матрицы (без параметров).

Разработать класс MatrDemo, содержащий статические методы:

- 1) процедура вывода вектора (одномерного массива) в окно терминала (параметр – ссылка на вектор);
- 2) метод main, демонстрирующий применение методов класса «Интеллектуальная матрица» для двух матриц различной размерности.

Вариант 2. Разработать класс «Интеллектуальная матрица» (IntellectMatr).

Поля: 1) ссылка на прямоугольную (квадратную) матрицу (public), тип элементов матрицы – double.

Методы:

- 1) конструктор (параметр – ссылка на матрицу);
- 2) процедура вывода матрицы в окно терминала (без параметров);

- 3) функция, возвращающая значение суммы элементов под главной диагональю матрицы (без параметров);
- 4) функция, возвращающая значение определителя матрицы (без параметров);
- 5) функция, возвращающая адрес вектора (одномерного массива), i -ый элемент которого равен сумме положительных элементов i -ой строки матрицы (без параметров);
- 6) функция, возвращающая адрес матрицы-произведения инкапсулированной матрицы на другую матрицу (параметр – ссылка на матрицу).

Разработать класс `MatrDemo`, содержащий статические методы:

- 1) процедура вывода вектора (одномерного массива) в окно терминала (параметр – ссылка на вектор);
- 2) метод `main`, демонстрирующий применение методов класса «Интеллектуальная матрица» для двух матриц различной размерности.

Вариант 3. Разработать класс «Интеллектуальная матрица» (`IntellectMatr`).

Поля: 1) ссылка на прямоугольную (квадратную) матрицу (`public`), тип элементов матрицы – `int`.

Методы:

- 1) конструктор (параметр – ссылка на матрицу);
- 2) процедура вывода матрицы в окно терминала (без параметров);
- 3) функция, возвращающая значение произведения элементов над главной диагональю матрицы (без параметров);
- 4) функция, возвращающая значение определителя матрицы (без параметров);
- 5) функция, возвращающая адрес вектора (одномерного массива), i -ый элемент которого равен произведению положительных элементов i -го столбца матрицы (без параметров);
- 6) функция, возвращающая адрес транспонированной матрицы (без параметров).

Разработать класс `MatrDemo`, содержащий статические методы:

- 1) процедура вывода вектора (одномерного массива) в окно терминала (параметр – ссылка на вектор);
- 2) метод `main`, демонстрирующий применение методов класса «Интеллектуальная матрица» для двух матриц различной размерности.

Вариант 4. Разработать класс «Интеллектуальная матрица» (IntellectMatr).

Поля: 1) ссылка на прямоугольную (квадратную) матрицу (public), тип элементов матрицы – double.

Методы:

- 1) конструктор (параметр – ссылка на матрицу);
- 2) процедура вывода матрицы в окно терминала (без параметров);
- 3) функция, возвращающая значение максимального элемента под второй (не главной) диагональю матрицы (без параметров);
- 4) функция, возвращающая значение определителя матрицы (без параметров);
- 5) функция, возвращающая адрес вектора (одномерного массива), *i*-ый элемент которого равен числу положительных элементов в *i*-ом столбце матрицы (без параметров);
- 6) функция, возвращающая адрес матрицы-произведения инкапсулированной матрицы на другую матрицу (параметр – ссылка на матрицу).

Разработать класс MatrDemo, содержащий статические методы:

- 1) процедура вывода вектора (одномерного массива) в окно терминала (параметр – ссылка на вектор);
- 2) метод main, демонстрирующий применение методов класса «Интеллектуальная матрица» для двух матриц различной размерности.

Вариант 5. Разработать класс «Интеллектуальная матрица» (IntellectMatr).

Поля: 1) ссылка на прямоугольную (квадратную) матрицу (public), тип элементов матрицы – int.

Методы:

- 1) конструктор (параметр – ссылка на матрицу);
- 2) процедура вывода матрицы в окно терминала (без параметров);
- 3) функция, возвращающая значение минимального элемента над второй (не главной) диагональю матрицы (без параметров);
- 4) функция, возвращающая значение определителя матрицы (без параметров);

5) функция, возвращающая адрес вектора (одномерного массива), i -ый элемент которого равен единице, если в i -ом столбце матрицы есть положительные элементы, и нулю, в противном случае;

6) функция, возвращающая адрес транспонированной матрицы (без параметров).

Разработать класс `MatrDemo`, содержащий статические методы:

1) процедура вывода вектора (одномерного массива) в окно терминала (параметр – ссылка на вектор);

2) метод `main`, демонстрирующий применение методов класса «Интеллектуальная матрица» для двух матриц различной размерности.