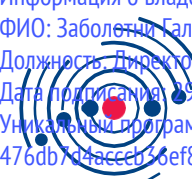


Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Заболотни Галина Ивановна  
Должность: Директор филиала  
Дата подписания: 09.04.2026 16:24:02  
Уникальный программный ключ:  
476db7d4acc6b30ef81301b7be235473473d63457266ce26b7e9e40f733b8b08



**САМАРСКИЙ  
ПОЛИТЕХ**  
Спортивный университет

**МИНОБРНАУКИ РОССИИ**  
федеральное государственное бюджетное образовательное  
учреждение высшего образования  
**«Самарский государственный технический университет»**  
(ФГБОУ ВО «СамГТУ»)

**Методические указания  
по выполнению лабораторных работ  
по дисциплине:**

**ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ**

для обучающихся очной, очно-заочной и заочной форм обучения  
направления подготовки 09.03.01 Информатика и вычислительная техника  
профиль Информатика и вычислительная техника в нефтехимическом производстве

Методические указания разработаны на кафедре «Информатика и системы управления» в соответствии с требованиями ФГОС ВО по направлению подготовки **09.03.01 Информатика и вычислительная техника**, утвержденного приказом Министерства образования и науки РФ от № 929 от 19.09.2017 и рабочей программой дисциплины «Языки и методы программирования».

Методические указания предназначены для обучающихся очной, заочной и очно-заочной форм обучения и содержат указания по выполнению лабораторных работ, а также требования к оформлению отчетов по их выполнению.

**Разработчик(и):** Рубцова Т.П., Волкодаева А.В.

## Содержание

1. Общие положения .....	4
2. Правила работы в лаборатории .....	5
3. Алгоритм проведения лабораторной работы .....	7
4. Материально-техническое обеспечение выполнения лабораторных работ.....	8
5. Учебно-методическое обеспечение выполнения лабораторных работ .....	9
6. Содержание лабораторных работ.....	11
7. Критерии и показатели оценки результата выполнения лабораторных работ .....	19
Приложение А.....	20
Приложение Б.....	21

## 1. Общие положения

**Лабораторные занятия** – одна из форм практического занятия, являющаяся эффективной формой учебных занятий в образовательной организации. Лабораторные занятия имеют выраженную специфику в зависимости от учебной дисциплины, углубляют и закрепляют теоретические знания. Лабораторные занятия дают наглядное представление об изучаемых процессах, обучающиеся осваивают постановку и ведение эксперимента, учатся оценивать полученные результаты, делать выводы и обобщения.

**Лабораторная работа** – это форма организации учебного процесса, в рамках которой обучающиеся по заданию и под руководством преподавателя самостоятельно выполняют специально разработанные задания. Лабораторная работа как вид учебного занятия должна проводиться в специально оборудованных учебных лабораториях. Продолжительность – не менее двух академических часов. Необходимыми структурными элементами лабораторной работы, помимо самостоятельной деятельности обучающегося, являются инструктаж, проводимый преподавателем, а также организация обсуждения итогов выполнения лабораторной работы.

### **Цели лабораторного занятия:**

- углубление и закрепление знания теоретического курса путем практического изучения в лабораторных условиях изложенных в лекциях законов и положений;

- приобретение навыков в научном экспериментировании, анализе полученных результатов;

- формирование первичных навыков организации, планирования и проведения практических работ и исследований.

В зависимости от задач, решаемых на лабораторных занятиях, различают:

- ознакомительные лабораторные занятия, которые проводятся с целью закрепления и конкретизации изученного теоретического материала, а также для изучения конструктивных особенностей, устройство средств производственной деятельности (оборудования, инструментов приспособлений и т.д.) и средств исследовательской деятельности (испытательных установок, приборов и т.д.), а также их наладки и настройки;

- экспериментальные лабораторные занятия, которые проводятся с целью получение новой информации на основе формализованных методов, обеспечивающих накопление знаний, умений и практического опыта и включают экспериментальные и исследовательские задания (по изучению и отработке методики проведения различных исследований, по конструированию, переконструированию и доконструированию различных схем и приспособлений, по исследованию влияния различных факторов на свойства объектов, по определению степени соответствия экспериментальных и расчетных данных, по проверке, иллюстрации, подтверждению законов, закономерностей и т.д.;

- творческие лабораторные занятия (проблемно-поисковые работы), которые ставят своей целью получение новой информации на основе формализованных методов и обеспечивают накопление знаний, умений и практического опыта, а также включают постановку и проведение экспериментов и отличаются они только степенью проблемности экспериментальных задач, при этом речь идет об уровнях проблемности этих задач: новизне объектов, условий, в которых проводится эксперимент по сравнению с известными ранее (к этой группе лабораторных работ относятся и работы по проверке различных гипотез учебного и научного уровня проблемности).

При проведении лабораторных занятий учебная группа может делиться на подгруппы численностью не менее 8 человек, а в случае индивидуальной подготовки и менее.

## **2. Правила работы в лаборатории**

Правила работы в лаборатории обязательны для исполнения всеми обучающимися, преподавателями и сотрудниками, находящимися в лаборатории. Нарушение правил влечет за собой предупреждение, отстранение от работы и/или другие дисциплинарные меры, предусмотренные уставом образовательной организации. Администрация лаборатории не несет ответственности за несчастные случаи, произошедшие в результате несоблюдения настоящих правил.

### **Одежда и защитные средства при выполнении лабораторной работы**

При проведении лабораторных работ по дисциплине «Языки и методы программирования» не предусмотрены специальная одежда и защитные средства. Обучающимся запрещается находиться в аудитории в верхней одежде при работе за компьютером.

### **Инструктаж перед выполнением лабораторной работы**

Перед началом выполнения лабораторной работы обучающиеся должны пройти инструктаж по работе в лаборатории, оснащенной персональными компьютерами. Инструктаж может быть как общим (в начале семестра), так и индивидуальным (перед каждой работой, при необходимости). Инструктаж включает:

1. Общие правила работы в лаборатории.
2. Меры безопасности при работе с компьютерной техникой (электробезопасность, правильная посадка за рабочим местом, перерывы).
3. Правила работы с программным обеспечением.
4. Правила хранения и использования данных (конфиденциальность, резервное копирование).
5. Подтверждение прохождения инструктажа – подпись обучающегося в журнале (при необходимости).

1. Общие правила работы в лаборатории:
  - поддерживать чистоту и порядок на рабочем месте;
  - не оставлять личные вещи на проходах и рабочих столах;
  - не употреблять пищу и напитки за компьютерами;
  - не использовать постороннее программное обеспечение без разрешения преподавателя;
  - сообщать о любых неисправностях оборудования или программного обеспечения лаборанту или преподавателю.
  - соблюдать правильную осанку при работе за компьютером;
  - регулярно делать перерывы для отдыха глаз и разминки.
2. Меры безопасности при работе с компьютерной техникой:
  - запрет на эксплуатацию поврежденного оборудования: необходимо немедленно прекратить использование компьютера, если имеются повреждения корпуса или силовых кабелей, а также если в розетке отсутствует заземление;
  - избежание посторонних предметов на системном блоке: на корпусе системного блока не должно быть посторонних предметов, так как это может привести к вибрациям и сбоям в работе оборудования;
  - работа в сухих условиях: следует избегать работы с компьютером в условиях повышенной влажности или при открытом корпусе;
  - правильное расположение проводов и кабелей: провода должны располагаться так, чтобы исключить риск наступления на них или перегрузки тяжелыми предметами;

- не пытаться самостоятельно ремонтировать компьютерную технику;
- использовать средства защиты от излучения монитора (при необходимости);
- не перекрывать вентиляционные отверстия на системном блоке и мониторе;
- не прикасаться к экрану и корпусу монитора, а также изменять местоположение системного блока и монитора.

### 3. Правила работы с программным обеспечением:

- необходимо использовать только лицензионное программное обеспечение;
- запрещается устанавливать программное обеспечение без разрешения преподавателя;
- запрещается модернизировать или изменять параметры (настройки) операционной системы.

### 4. Правила хранения и использования данных

- при необходимости создавать резервные копии своих данных;
- не распространять персональные данные без разрешения;
- не посещать сайты, содержащие вирусы или вредоносное программное обеспечение.

### **Меры безопасности при выполнении лабораторной работы**

При выполнении лабораторной работы обучающиеся обязаны соблюдать следующие меры безопасности:

- включать, отключать, работать за компьютером без разрешения преподавателя или лаборанта;
- подключать и отключать любые периферийные устройства, за исключением флэш-накопителей; прикасаться к соединительным кабелям и их разъемам.
- прикасаться к соединительным кабелям и их разъемам;
- открывать корпус системного блока, монитора, периферийных устройств;
- размещать какие-либо предметы (тетради, дискеты, книги и др.) на элементах оборудования персонального компьютера;
- продолжать работу при наличии сбоев и неполадок функционирования, нехарактерного шума компьютера или признаков возникновения пожара (запаха гари).
- открывать корпус системного блока, монитора, периферийных устройств;

Перед началом работы обучающиеся должны убедиться в отсутствии видимых повреждений на компьютерах (нарушении целостности корпуса, нарушении изоляции проводов, неисправность индикации включения питания, признаки электрического напряжения на корпусе и т. д.), начинать работу с персональным компьютером только по указанию преподавателя.

В процессе работы обучающиеся должны при непроизвольном отключении персонального компьютера, сбоев в работе, нехарактерного шума или запаха гари немедленно сообщить преподавателю или лаборанту.

По окончании работы необходимо выполнить операцию выхода из системы (компьютер не выключать!), поставить преподавателя в известность об окончании работы с персональным компьютером.

### **Ответственность обучающихся при выполнении лабораторной работы**

Обучающиеся несут ответственность за сохранность оборудования и программного обеспечения, предоставленных для выполнения лабораторной работы. В случае повреждения по вине обучающегося, он обязан возместить ущерб в установленном порядке. Обучающиеся несут ответственность за нарушение конфиденциальности данных.

### 3. Алгоритм проведения лабораторной работы

**Цель проведения лабораторных работ** по дисциплине «Языки и методы программирования» заключается в формировании у обучающихся практических навыков применения технологий программирования, при решении задач профессиональной деятельности, разработки алгоритмов и программ, пригодных для практического применения, использования технологий программирования при оснащении отделов, лабораторий, офисов компьютерным и сетевым оборудованием, тестирования работоспособности программ.

Содержание лабораторных работ определяется требованиями к результатам обучения по дисциплине «Языки и методы программирования» (таблица 1) в виде умений и навыков в соответствии с **компетенциями**.

Таблица 1

Планируемые результаты обучения по дисциплине «Языки и методы программирования»

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения (знать, уметь, владеть, соотнесенные с индикаторами достижения компетенции)
ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.1 Составляет алгоритмы, пишет и отлаживает коды на языке программирования	<b>Владеть</b> навыками составления алгоритмов и программ, пригодных для практического применения <b>Знать</b> методы составления алгоритмов и программ, пригодных для практического применения <b>Уметь</b> составлять алгоритмы, писать и отлаживать коды, пригодные для практического применения
	ОПК-8.2 Применяет языки программирования, современные программные среды разработки информационных систем и технологии для решения прикладных задач различного класса	<b>Владеть</b> навыками применения языков программирования, современных технологии для решения прикладных задач различного класса <b>Знать</b> методы применения языков программирования, современных технологии для решения прикладных задач различного класса <b>Уметь</b> применять языки программирования, современные технологии для решения прикладных задач различного класса

Алгоритм проведения лабораторной работы по дисциплине «Языки и методы программирования» включает шесть основных этапов. Последовательность и содержание выполнения лабораторной работы представлено в таблице 2.

Таблица 2

Последовательность и содержание выполнения лабораторной работы по дисциплине «Языки и методы программирования»

Этап	Содержание
1. Подготовка к лабораторной работе	Получение задания от преподавателя (описание задачи, данные, требования к отчету). Изучение теоретического материала (конспекты лекций, учебная литература, методические указания). Ознакомление с используемым программным обеспечением. Подготовка плана выполнения работы.
2. Начало лабораторной работы	Запуск необходимого программного обеспечения. Загрузка предоставленных данных или подготовка данных самостоятельно. Создание новых файлов для хранения результатов работы.
3. Выполнение работы	Выполнение анализа данных в соответствии с заданием. Очистка и подготовка данных. Выбор и применение подходящих методов выполнения поставленных задач. Интерпретация результатов анализа. Фиксация промежуточных результатов. Создание таблиц, графиков, диаграмм для визуализации данных. Формулировка выводов и рекомендаций на основе результатов анализа.
4. Завершение работы	Сохранение всех рабочих файлов. Закрытие программного обеспечения. Удаление временных файлов (при необходимости).
5. Оформление отчета	В отчете должны быть четко сформулированы цель работы, описание использованных методов, результаты анализа (с таблицами, графиками), выводы и рекомендации. Отчет должен быть оформлен в соответствии с требованиями методических указаний. Указание источников данных и использованного программного обеспечения.
6. Защита лабораторной работы	Демонстрация преподавателю результатов работы. Ответы на вопросы по методике анализа, интерпретации результатов и сделанным выводам. Объяснение ограничений использованных методов. Предложения по дальнейшему развитию анализа. Обоснование практической значимости полученных результатов лабораторной работы.

#### **4. Материально-техническое обеспечение выполнения лабораторных работ**

Для проведения лабораторных занятий и выполнения лабораторных работ по дисциплине «Языки и методы программирования», образовательная организация располагает материально-технической базой, соответствующей действующим санитарным и противопожарным правилам и нормам. Помещения оснащены оборудованием персональными компьютерами, программным обеспечением, комплектом мебели для обучающихся и преподавателя, а также другими техническими средствами обучения.

Каждый обучающийся обеспечен индивидуальным неограниченным доступом к электронной информационно-образовательной среде образовательной организации из любой точки, в которой имеется доступ к информационно-телекоммуникационной сети «Интернет», как на территории образовательной организации, так и вне ее.

Компьютерная техника оснащена подключения к информационно-телекоммуникационной сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду образовательной организации.

Образовательная организация обеспечена необходимым комплектом лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства (в соответствии с программой дисциплины) и подлежит обновлению при необходимости.

Оборудование и программное обеспечение, используемое при проведении лабораторных работ по дисциплине «Языки и методы программирования»:

- программное обеспечение: среда разработки для C++ (например, Code::Blocks, Visual Studio Community) и/или Anaconda 3 (Python), браузер, доступ в ЭИОС и Интернет;
- документация по языку C++/Python, [cplusplus.com](http://cplusplus.com), [cpreference.com](http://cpreference.com).

## **5. Учебно-методическое обеспечение выполнения лабораторных работ**

Учебно-методическое обеспечение выполнения лабораторных работ по дисциплине «Языки и методы программирования» включает:

1. Перечень литературы по дисциплине.
2. Методические указания по выполнению лабораторных работ.
3. Описание процесса проведения лабораторных работ.

### **1. Перечень литературы по дисциплине.**

Основная литература:

1. Алгоритмизация и программирование; Вузовское образование, 2019.- Режим доступа: [https://elib.samgtu.ru/getinfo?uid=els\\_samgtu||iprbooks||80539](https://elib.samgtu.ru/getinfo?uid=els_samgtu||iprbooks||80539)

2. Введение в программирование; Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020.- Режим доступа: [https://elib.samgtu.ru/getinfo?uid=els\\_samgtu||iprbooks||97539](https://elib.samgtu.ru/getinfo?uid=els_samgtu||iprbooks||97539)

3. Программирование: основы языка C++; Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2020.- Режим доступа: [https://elib.samgtu.ru/getinfo?uid=els\\_samgtu||iprbooks||102464](https://elib.samgtu.ru/getinfo?uid=els_samgtu||iprbooks||102464)

Дополнительная литература

1. Программирование в примерах и задачах; Лаборатория знаний, 2020.Режим доступа: [https://elib.samgtu.ru/getinfo?uid=els\\_samgtu||iprbooks||99863](https://elib.samgtu.ru/getinfo?uid=els_samgtu||iprbooks||99863)

2. Программирование на языке высокого уровня C/C++; МИСИ-МГСУ, ЭБС АСВ, 2018.- Режим доступа: [https://elib.samgtu.ru/getinfo?uid=els\\_samgtu||iprbooks||76390](https://elib.samgtu.ru/getinfo?uid=els_samgtu||iprbooks||76390)

3. Программирование: типовые задачи, алгоритмы, методы; Лаборатория знаний, 2020.- Режим доступа: [https://elib.samgtu.ru/getinfo?uid=els\\_samgtu||iprbooks||12264](https://elib.samgtu.ru/getinfo?uid=els_samgtu||iprbooks||12264)

Доступ обучающихся к ЭР НТБ СамГТУ ([elib.samgtu.ru](http://elib.samgtu.ru)) осуществляется посредством электронной информационной образовательной среды университета и сайта НТБ СамГТУ по логину и паролю.

Методические материалы размещены на сайте филиала ФГБОУ ВО «СамГТУ» в г. Новокуйбышевске, в разделе «Сведения об образовательной организации», подраздел «Образование», таблица «Информация по образовательным программам» в ячейке «Ссылка на иные компоненты, оценочные и методические материалы, а также в предусмотренных ФЗ от 29.12.2012 № 273-ФЗ «Об образовании в РФ» случаях в виде рабочей программы воспитания, календарного плана воспитательной работы, форм аттестации в виде электронного документа».

### **2. Методические указания по выполнению лабораторных работ.**

Методические указания по выполнению лабораторных работ по дисциплине «Языки и методы программирования» содержат общие положения, правила работы в лаборатории, алгоритм проведения лабораторной работы, материально-техническое обеспечение выполнения лабораторных работ, учебно-методическое обеспечение выполнения лабораторных работ, содержание лабораторных работ, критерии и показатели оценки результата выполнения лабораторных работ.

### **3. Описание процесса проведения лабораторных работ**

Процесс проведения лабораторных работ по дисциплине «Языки и методы программирования» направлен на формирование у обучающихся практических навыков в соответствии с указанной выше целью. Процесс проведения лабораторных работ состоит из шести этапов, перечисленных выше. Рассмотрим

содержание и ход выполнения работ в соответствии с этапами проведения лабораторной работы.

#### 1. Подготовка к лабораторной работе

Преподаватель предоставляет обучающим описание лабораторной работы, включающее цель работы, постановку задачи, требования к программному обеспечению, требования к отчету по результатам проведения лабораторной работы, критерии оценки (параметры, по которым будет оцениваться выполненная работа).

Обучающиеся выбирают и осваивают программные средства, которые будут использоваться для выполнения работы, знакомятся с интерфейсом, основными функциями и инструментами, примерами выполнения учебных заданий, осуществляют поиск дополнительных материалов и документации для выполнения лабораторных задач. Обучающиеся разрабатывают план выполнения лабораторной работы, включающий определение последовательности действий, распределение времени на выполнение каждого этапа работы, подготовку необходимых ресурсов (данные, программное обеспечение, инструменты).

#### 2. Выполнение лабораторной работы (начало лабораторной работы, выполнение работы, завершение работы).

Обучающиеся собирают необходимые для выполнения лабораторной работы данные из указанных источников, подготавливают данные для анализа, осуществляет преобразование данных, масштабирование данных. В процессе анализа данных обучающиеся применяют выбранные методы для решения поставленной задачи, интерпретируют результаты анализа данных, делая выводы о решении поставленной задачи. Выводы должны быть обоснованы результатами анализа. Обучающиеся оформляют результаты работы в виде таблиц, графиков, диаграмм. Формат представления результатов должен быть понятным и наглядным.

#### 3. Оформление и сдача отчета

Обучающиеся оформляют отчет по лабораторной работе. Отчет включает следующие элементы:

- титульный лист (Приложения А);
- цель работы;
- индивидуальный вариант задания;
- листинг программы (исходный код с комментариями, с указанием структуры и специальной функции);
- результаты выполнения программы (скриншот консоли с выводом данных и результата);
- выводы по лабораторной работе.

Отчет должен быть оформлен в соответствии с требованиями. Текст должен быть четким и лаконичным. Таблицы и графики должны быть подписаны и пронумерованы. Формулы должны быть набраны в редакторе формул. Отчет должен быть предоставлен на проверку преподавателю в установленный срок. Образец отчета по лабораторной работе представлен в Приложении Б.

#### 4. Защита лабораторной работы

Обучающиеся готовятся к защите, повторяя материал и анализируя результаты своей работы. Необходимо быть готовым ответить на вопросы преподавателя, касающиеся цели и задач работы, методики выполнения работы, результатов работы и сформулированных выводов.

В процессе защиты, обучающийся кратко излагает цель, задачи и результаты своей работы, отвечает на вопросы преподавателя, демонстрирует понимание результатов анализа и их практической значимости. Преподаватель оценивает качество выполненной работы, уровень знаний обучающегося и его способность применять эти знания на практике.

Обучающиеся должны активно участвовать в выполнении лабораторных работ, проявлять самостоятельность и инициативу. При анализе данных необходимо проявлять критическое мышление, оценивать достоверность информации и обоснованность выводов.

## 6. Содержание лабораторных работ

В рамках дисциплины «Языки и методы программирования» предусмотрено 16 академических часов на выполнение лабораторных работ согласно учебному плану. Лабораторные работы предусмотрены в рамках разделов дисциплины:

- структурное программирование;
- объектно-ориентированное программирование.

Темы лабораторных работ и соответствующие им планируемые результаты обучения представлены в таблице 3.

Таблица 3

Темы лабораторных работ и соответствующие планируемые результаты обучения

Темы лабораторных работ	Кол-во часов	Код индикатора достижения компетенции	Планируемые результаты обучения (умения, навыки)
1. Основы объектно-ориентированного программирования	2	ОПК-8.1	<b>Владеть</b> навыками составления алгоритмов и программ, пригодных для практического применения <b>Знать</b> методы составления алгоритмов и программ, пригодных для практического применения <b>Уметь</b> составлять алгоритмы, писать и отлаживать коды, пригодные для практического применения
2. Отношения между объектами и/или классами	4	ОПК-8.1	<b>Владеть</b> навыками составления алгоритмов и программ, пригодных для практического применения <b>Знать</b> методы составления алгоритмов и программ, пригодных для практического применения <b>Уметь</b> составлять алгоритмы, писать и отлаживать коды, пригодные для практического применения
3. Программирование статических и движущихся изображений	4	ОПК-8.2	<b>Владеть</b> навыками применения языков программирования, современных технологии для решения прикладных задач различного класса <b>Знать</b> методы применения языков программирования, современных технологии для решения прикладных задач различного класса <b>Уметь</b> применять языки программирования, современные технологии для решения прикладных задач различного класса
4.Связь структур данных с управляющими структурами языков программирования	6	ОПК-8.2	<b>Владеть</b> навыками применения языков программирования, современных технологии для решения прикладных задач различного класса <b>Знать</b> методы применения языков программирования, современных

Темы лабораторных работ	Кол-во часов	Код индикатора достижения компетенции	Планируемые результаты обучения (умения, навыки)
			технологии для решения прикладных задач различного класса <b>Уметь</b> применять языки программирования, современные технологии для решения прикладных задач различного класса

Содержание лабораторных работ отражает цель, код и наименование индикатора достижения компетенции, виды работ, время выполнения лабораторной работы, информацию о подготовке к лабораторной работе, общее задание для всех вариантов ответов, индивидуальные варианты заданий, методику выполнения, содержание отчета (шаблон), выводы, контрольные вопросы (для допуска и защиты).

### **Лабораторная работа №1. Основы объектно-ориентированного программирования**

**Цель:** Освоить создание классов и объектов, конструкторов и деструкторов, методов доступа и других элементов объектно-ориентированного программирования.

**Код и наименование индикатора достижения компетенции:** ОПК-8.1  
Составляет алгоритмы, пишет и отлаживает коды на языке программирования

**Время выполнения:** 2 часа(-ов).

**Подготовка к работе (повторить):**

- понятия класс и объект;
- инкапсуляция, методы доступа, модификаторы доступа;
- конструкторы и деструкторы;
- создание и использование объектов.

**Общее задание для всех вариантов:**

1. Разработайте класс в соответствии с вариантом.
2. Определите поля, конструкторы, методы для доступа к полям и вывода информации.
3. Создайте массив (или список) объектов класса и реализуйте операцию поиска/вычисления согласно варианту.
4. Выведите результаты работы программы.

**Индивидуальные варианты заданий:**

**Вариант 1.**

Разработайте класс Student с полями: id, имя, специальность, средний балл. Реализуйте метод отображения данных и метод определения студента с наивысшим средним баллом.

**Вариант 2.**

Разработайте класс Book с полями: название, автор, год издания, количество страниц. Реализуйте метод вычисления общего числа страниц и поиска самой старой книги.

**Вариант 3.**

Разработайте класс Car с полями: модель, производитель, год, мощность. Найдите автомобиль с максимальной мощностью.

**Вариант 4.**

Разработайте класс Employee с полями: id, имя, отдел, зарплата. Подсчитайте суммарную зарплату по отделу.

**Вариант 5.**

Разработайте класс Course с полями: код, название, кредиты, преподаватель. Найдите курс с наибольшим числом кредитов.

**Вариант 6.**

Разработайте класс Product с полями: id, название, категория, цена. Найдите товар с минимальной ценой.

**Вариант 7.**

Разработайте класс Client с полями: id, имя, адрес, телефон. Реализуйте поиск клиента по номеру телефона.

**Вариант 8.**

Разработайте класс Movie с полями: название, режиссёр, год, рейтинг. Выведите фильмы с рейтингом выше заданного значения.

**Вариант 9.**

Разработайте класс City с полями: название, страна, население, площадь. Вычислите плотность населения и найдите город с максимальной плотностью.

**Вариант 10.**

Разработайте класс BankAccount с полями: номер счёта, владелец, баланс, процентная ставка. Реализуйте операции пополнения/снятия и поиска счёта с наибольшим балансом.

**Методика выполнения:**

1. Выберите вариант задания и определите структуру класса (поля и методы).
2. Опишите класс в исходном файле, реализуйте конструктор по умолчанию и параметрический конструктор, деструктор (при необходимости).
3. Реализуйте методы для чтения и изменения полей (getters/setters) и дополнительный метод для вывода информации.
4. Создайте массив или список объектов класса. Сформируйте тестовые данные.
5. Напишите функцию для поиска или вычисления требуемой величины (например, поиск максимального значения).
6. Выведите полученные результаты на экран в удобном формате.

**Содержание отчёта (шаблон):**

1. Титульный лист.
2. Цель и задание.
3. Описание класса и его реализация (листинг).
4. Результаты выполнения программы (таблицы, скриншоты).
5. Выводы.

**Выводы:**

6. Объектно-ориентированный подход позволяет объединять данные и методы в единую сущность.

7. Конструкторы и деструкторы обеспечивают корректную инициализацию и освобождение ресурсов.
8. Инкапсуляция обеспечивает скрытие внутреннего состояния объекта и доступ через интерфейс.

#### **Контрольные вопросы:**

1. Что такое класс и объект?
2. Какие виды конструкторов вы знаете и для чего они служат?
3. Чем отличается метод от обычной функции?
4. Что такое инкапсуляция и как она реализуется в языке программирования?

#### **Лабораторная работа №2. Отношения между объектами и/или классами**

**Цель:** Освоить различные типы отношений между объектами и классами (ассоциация, агрегация, композиция, наследование) и научиться проектировать программы, состоящие из нескольких взаимодействующих классов.

**Код и наименование индикатора достижения компетенции:** ОПК-8.1 Составляет алгоритмы, пишет и отлаживает коды на языке программирования

**Время выполнения:** 4 часа(-ов).

**Подготовка к работе** (повторить):

- понятия ассоциация, агрегация, композиция и наследование;
- реализация связей «один-к-многим», «многие-ко-многим»;
- использование указателей и контейнеров для связи объектов;
- классы-контейнеры и ассоциативные классы.

Общее задание для всех вариантов:

1. Спроектируйте систему из нескольких классов в соответствии с вариантом.
2. Определите типы связей между классами (ассоциация, агрегация, композиция).
3. Реализуйте классы и методы для создания, хранения и обработки связанных объектов.
4. Реализуйте функцию для вычисления требуемого показателя (например, подсчёт количества элементов или среднее значение).

Индивидуальные варианты заданий:

Вариант 1.

Система «Библиотека»: классы Author и Book (отношение один-ко-многим).

Определите количество книг для каждого автора и найдите автора с наибольшим количеством книг.

Вариант 2.

Система «Учебный процесс»: классы Student и Course связаны отношением многие-ко-многим через класс Enrollment (содержит оценку). Рассчитайте средний балл по каждому курсу.

Вариант 3. Система «Компания»: классы Employee и Department (один-ко-многим). Подсчитайте суммарную зарплату сотрудников в каждом отделе.

Вариант 4.

Система «Научная деятельность»: классы Professor и Publication (композиция). Найдите профессора с наибольшим числом публикаций.

Вариант 5.

Система «Каталог товаров»: классы Product и Supplier (ассоциация). Подсчитайте общую стоимость товаров для каждого поставщика.

Вариант 6.

Система «Социальная сеть»: класс User с отношением дружбы (двусторонняя ассоциация). Выведите список друзей для каждого пользователя.

Вариант 7.

Система «Поликлиника»: классы Doctor, Patient и Appointment. Найдите доктора с максимальным числом приёмов.

Вариант 8.

Система «Прокат автомобилей»: классы Car, Customer и Rental. Определите, сколько раз каждый клиент брал машину в аренду.

Вариант 9.

Система «Банк»: классы Customer, Account и Transaction. Рассчитайте общее количество транзакций по каждому счёту.

Вариант 10.

Система «Городской транспорт»: классы Route, Bus и Driver. Определите нагрузку на каждого водителя (количество маршрутов).

Методика выполнения:

1. Выберите вариант и проанализируйте сущности и связи между ними.
2. Определите классы, их атрибуты и методы, укажите тип связей (агрегация, композиция и т.д.).
3. Реализуйте классы в коде, используйте указатели, списки или словари для хранения связанных объектов.
4. Создайте тестовые объекты и свяжите их в соответствии с выбранным вариантом.
5. Напишите функцию для вычисления требуемого показателя (например, подсчёт количества объектов, среднее значение) и выведите результат.

Содержание отчёта (шаблон):

- Титульный лист.
- Цель и задание.
- Диаграмма классов (при наличии) и описание связей.
- Листинг программы.
- Результаты выполнения и выводы.

**Выводы:**

1. Отношения между объектами позволяют строить более сложные модели реальных систем.
2. Композиция подразумевает жёсткую связь, при которой жизненный цикл объектов взаимосвязан.
3. Агрегация – более слабая связь, объекты могут существовать независимо.
4. Ассоциация описывает факт связи без указания её силы.

**Контрольные вопросы:**

1. Чем отличаются ассоциация, агрегация и композиция?
2. В каких случаях используется наследование?
3. Как реализовать отношение многие-ко-многим?

4. Чем отличается композиция от агрегации с точки зрения жизненного цикла объектов?

### **Лабораторная работа №3. Программирование статических и движущихся изображений**

**Цель:** Изучить основные подходы к программированию статических и движущихся изображений, освоить работу с графическими примитивами и создание простых анимаций.

**Код и наименование индикатора достижения компетенции:** ОПК-8.2 Применяет языки программирования, современные программные среды разработки информационных систем и технологии для решения прикладных задач различного класса

**Время выполнения:** 4 часа(-ов).

Подготовка к работе (повторить):

- координатная система и графические примитивы (точка, линия, прямоугольник, окружность);
- базовые функции вывода графики в выбранной библиотеке (например, Turtle, Pygame, SFML);
- основы построения событийных циклов и управления временем для анимации.

#### **Общее задание для всех вариантов:**

1. Разработайте программу, рисующую статическое изображение согласно варианту.
2. Добавьте возможность анимированного перемещения или изменения элементов изображения.
3. Обеспечьте корректное завершение работы программы и управление частотой кадров (при необходимости).

#### **Индивидуальные варианты заданий:**

Вариант 1. Нарисовать дом и солнце; реализовать анимацию восхода солнца.

Вариант 2. Нарисовать автомобиль на дороге; реализовать его движение слева направо.

Вариант 3. Нарисовать аналоговые часы; реализовать движение секундной стрелки.

Вариант 4. Нарисовать падающий и отскакивающий мяч; реализовать простую гравитацию.

Вариант 5. Нарисовать многоугольник; реализовать его вращение вокруг центра.

Вариант 6. Нарисовать звёздное небо; реализовать эффект мерцания звёзд.

Вариант 7. Нарисовать столбчатую диаграмму; анимировать рост столбцов.

Вариант 8. Нарисовать фрактальное дерево; анимировать его постепенный рост.

Вариант 9. Нарисовать логотип и реализовать его плавное масштабирование.

Вариант 10. Нарисовать ракету; реализовать анимацию её взлёта.

#### **Методика выполнения:**

1. Выберите вариант и определите необходимые графические элементы.
2. Настройте среду разработки и библиотеку для графики.
3. Создайте функции для рисования статического изображения.
4. Добавьте цикл анимации и изменяйте координаты/параметры элементов во времени.
5. Управляйте скоростью анимации с помощью задержек или функции таймера.

6. Проверьте корректность работы и обработку завершения программы.

#### **Содержание отчёта (шаблон):**

1. Титульный лист.
2. Цель и задание.
3. Описание используемой библиотеки и принципов построения графики.
4. Листинг программы.
5. Скриншоты или описание полученного изображения/анимации.
6. Выводы.

#### **Выводы:**

1. Статические изображения формируются из графических примитивов с заданными координатами.
2. Анимация требует управления временем и изменения параметров объектов в цикле.
3. Разные библиотеки имеют свои особенности, но общие принципы остаются схожими.

#### **Контрольные вопросы:**

1. Что такое графический примитив? Перечислите основные примитивы.
2. В чём состоит различие между статическим изображением и анимацией?
3. Как организовать цикл событий для анимации?
4. Почему важно управлять частотой кадров при создании анимации?

#### **Лабораторная работа №4. Связь структур данных и алгоритмов их обработки с управляющими структурами**

**Цель:** Изучить связь структур данных и алгоритмов их обработки с управляющими структурами языка программирования, освоить применение циклов, условных операторов и рекурсии при работе с данными.

**Код и наименование индикатора достижения компетенции:** ОПК-8.2 Применяет языки программирования, современные программные среды разработки информационных систем и технологии для решения прикладных задач различного класса

**Время выполнения:** 6 часа(-ов).

**Подготовка к работе** (повторить):

- линейные структуры данных: массивы, списки;
- структуры со связями: стек, очередь, двусвязный список;
- деревья и графы, базовые алгоритмы поиска и сортировки;
- итеративные и рекурсивные способы обработки данных;
- анализ трудоёмкости алгоритмов (Big O).

#### **Общее задание для всех вариантов:**

1. Реализуйте выбранную структуру данных и операции добавления, удаления, поиска.
2. Используйте циклические и условные конструкции языка программирования для обработки структуры.
3. Проанализируйте трудоёмкость реализованных операций.
4. Реализуйте требуемый алгоритм обработки данных в соответствии с вариантом.

Индивидуальные варианты заданий:

Вариант 1. Динамический массив: реализуйте вставку, удаление и поиск элемента; найдите максимальное и минимальное значение.

Вариант 2. Односвязный список: реализуйте добавление в начало и конец, удаление элемента по значению, вычислите сумму элементов.

Вариант 3. Стек на основе массива: реализуйте операции push/pop; используйте стек для вычисления постфиксного выражения.

Вариант 4. Очередь на основе списка: реализуйте операции enqueue/dequeue; используйте очередь для моделирования обслуживания клиентов.

Вариант 5. Бинарное дерево поиска: реализуйте вставку и поиск; выполните обход дерева (in-order) и найдите высоту дерева.

Вариант 6. Хеш-таблица с цепочками: реализуйте вставку, поиск и удаление; используйте таблицу для хранения записей студентов.

Вариант 7. Граф и обход в ширину: реализуйте граф с помощью списка смежности; выполните поиск пути между двумя вершинами.

Вариант 8. Рекурсивные функции: реализуйте вычисление факториала и числа Фибоначчи, сравните трудоёмкость с итеративными вариантами.

Вариант 9. Сортировка: реализуйте пузырьковую сортировку и быструю сортировку, сравните их сложность.

Вариант 10. Конвертация инфиксного выражения в постфиксное с использованием стека.

Методика выполнения:

1. Выберите вариант и опишите структуру данных и требуемые операции.
2. Разработайте интерфейс (функции или методы) для работы со структурой.
3. Используйте циклы, условные операторы и/или рекурсию для реализации операций.
4. Создайте тестовый пример данных и проверьте корректность работы.
5. Проанализируйте асимптотическую трудоёмкость основных операций и сделайте выводы.

Содержание отчёта (шаблон):

1. Титульный лист.
2. Цель и задание.
3. Описание структуры данных и алгоритмов обработки.
4. Листинг программы.
5. Результаты тестирования и анализ трудоёмкости.
6. Выводы.

Выводы:

1. Выбор структуры данных влияет на сложность алгоритмов и эффективность работы программы.
2. Рекурсия и итерация могут решать одну и ту же задачу, но имеют различную трудоёмкость и особенности реализации.
3. Управляющие конструкции языка программирования (циклы, условия) являются основным инструментом обработки данных.

**Контрольные вопросы:**

1. Чем отличается динамический массив от статического?
2. В чем преимущества и недостатки списков по сравнению с массивами?
3. Что такое рекурсия и когда её использование оправдано?
4. Какова асимптотическая сложность сортировки пузырьком и быстрой сортировки?

## 7. Критерии и показатели оценки результата выполнения лабораторных работ

Оценка обучающемуся за выполненную лабораторную работу выставляется по четырехбалльной шкале по итогам проверки отчета и защиты работы. Основными **критериями оценки** лабораторной работы по дисциплине «Языки и методы программирования», являются:

- полнота и правильность выполнения задания (соответствие программы поставленному заданию, обработка всех требуемых случаев);
- качество кода (читаемость, структурированность, наличие комментариев, использование современных возможностей языка);
- оформление отчета (соответствие шаблону, наличие всех разделов, аккуратность, грамотность выводов);
- ответы на вопросы на защите отчета (понимание теоретических основ и практических аспектов выполненной работы);

В таблице 4 представлены критерии и показатели оценки выполнения лабораторной работы

Таблица 4

Критерии и показатели оценки выполнения лабораторной работы

Оценка	Критерии			
	Полнота и правильность выполнения	Качество кода	Оформление отчета	Ответы на контрольные вопросы/защита
<b>«Отлично»</b>	Задание выполнено полностью, код корректен.	Код чистый, хорошо структурирован, адекватно прокомментирован.	Отчет оформлен в соответствии с требованиями, выводы обоснованные.	Полные, уверенные, правильные ответы.
<b>«Хорошо»</b>	Задание выполнено с 1-2 незначительными недочетами.	Код читаем, но есть небольшие нарушения стиля.	Отчет оформлен с 1-2 замечаниями, выводы есть.	Ответы в целом правильные, но с неточностями.
<b>«Удовлетворительно»</b>	Задание выполнено частично, есть ошибки в логике.	Код плохо отформатирован, минимальные комментарии.	Отчет оформлен небрежно, выводы поверхностные.	Ответы неполные, требуются наводящие вопросы.
<b>«Неудовлетворительно»</b>	Задание не выполнено или выполнено неверно.	Код нечитаем.	Отчет не оформлен или оформление не соответствует требованиям.	Не может ответить на вопросы по теме.

Некоторые показатели, которые могут использоваться для оценки результата:

- функциональность – работающая реализация всех заявленных функций и возможностей.
- тестирование – наличие тестов и их результаты (покрытие кода тестами).

- креативность – оригинальные решения и подходы к реализации задачи.
- обработка ошибок – корректная обработка исключительных ситуаций и ошибок.
- пользовательский интерфейс – удобство и интуитивность интерфейса (если применимо).
- соответствие стандартам кодирования – следование принятым стандартам и стилям кодирования.

В качестве дополнительных критериев оценки лабораторной работы может быть: скорость выполнения программы, использование памяти и ресурсов, оптимизация алгоритмов и структур данных, выполнение требования к интерфейсу или взаимодействию с пользователем, используемые библиотеки и технологии, также учитывается, предоставлено ли описание работы программы, добавлены ли инструкции по запуску и использованию, документированы функции и классы в коде.

## **Приложение А**

Образец титульного листа отчета по лабораторной работе

---

### **МИНОБРНАУКИ РОССИИ**

федеральное государственное бюджетное образовательное учреждение высшего образования «Самарский государственный технический университет»  
(ФГБОУ ВО «СамГТУ»)

Филиал ФГБОУ ВО «СамГТУ» в г. Новокуйбышевске

Кафедра «Информатика и системы управления»

### **ОТЧЕТ**

по лабораторной работе № \_\_\_\_\_

по дисциплине «Языки и методы программирования»

Тема: \_\_\_\_\_

Выполнил:

обучающийся \_\_\_ курса \_\_\_\_\_ группы  
ФИО: \_\_\_\_\_

Принял:  
ФИО: \_\_\_\_\_

Оценка: \_\_\_\_\_

Дата: \_\_\_\_\_

Новокуйбышевск, 20\_\_\_

## Приложение Б

### Образец отчета по лабораторной работе

---

#### 1. Цель работы:

Сформировать навык использования структур для агрегации разнотипных данных, моделирующих сущность предметной области (нефтехимическое производство).

Научиться создавать структуры, инициализировать их, передавать в функции и производить над ними операции.

#### 2. Подготовка к работе (Повторение)

Перед выполнением работы был повторен следующий теоретический материал:

- понятие структуры (*struct*) в языке C/C++ как типа данных, объединяющего разнородные данные (поля);
- способы инициализации структур (списочная инициализация, через конструктор);
- доступ к полям структуры с помощью оператора точки;
- передача структур в функции по значению (копия) и по ссылке (&) для избежания копирования и возможности изменения.

#### 3. Ход работы (Программа выполнения)

3.1. Изучено задание, выбран вариант №1 (Насосное оборудование).

3.2. Создан новый проект в среде разработки.

3.3. Объявлена структура *Equipment* с базовыми полями и дополнительным полем *flow\_rate*.

3.4. В функции *main()* создан массив (вектор) структур *Equipment* и заполнен данными согласно варианту.

3.5. Реализована функция *print\_equipment()* для вывода информации об одном объекте.

3.6. Реализована функция *find\_max\_flow\_rate\_pump()* для поиска насоса с максимальной производительностью.

3.7. Реализован вывод исходного списка и результата работы специальной функции.

#### 4. Программный код

```
cpp
#include <iostream>
#include <string>
#include <vector>
#include <limits> // Для numeric_limits

using namespace std;

// 1. Определение структуры Equipment
struct Equipment {
    int id;           // Уникальный идентификатор
    string name;     // Наименование оборудования
    string workshop; // Цех/установка
    int year_commissioned; // Год ввода в эксплуатацию
    double flow_rate; // Производительность, м³/ч (для варианта 1)
};

// 2. Функция для печати одного элемента оборудования
void print_equipment(const Equipment& item) {
    cout << "ID: " << item.id
         << " | Наименование: " << item.name
         << " | Цех: " << item.workshop
         << " | Год ввода: " << item.year_commissioned
         << " | Производительность: " << item.flow_rate << " м³/ч" << endl;
}

// 3. Специальная функция для варианта 1: поиск насоса с макс. производительностью
Equipment find_max_flow_rate_pump(const vector<Equipment>& pumps) {
    // Предполагаем, что вектор не пустой (по условию задачи)
    Equipment max_pump = pumps[0];
    for (const auto& pump : pumps) {
        if (pump.flow_rate > max_pump.flow_rate) {
            max_pump = pump;
        }
    }
    return max_pump;
}

int main() {
    // 4. Создание массива (вектора) структур с данными
    vector<Equipment> equipment_list = {
        {1, "Насос центробежный ЦН-500", "Цех №1", 2018, 500.0},
        {2, "Насос шестеренчатый ШН-200", "Цех №2", 2020, 200.0},
        {3, "Насос вакуумный ВН-150", "Цех №3", 2019, 150.0},
        {4, "Насос дозировочный НД-80", "Цех №1", 2022, 80.0},
        {5, "Насос винтовой ВН-350", "Цех №2", 2021, 350.0}
    };

    // 5. Вывод исходного списка оборудования
    cout << "--- Исходный список насосного оборудования ---" << endl;
    for (const auto& item : equipment_list) {
        print_equipment(item);
    }
    cout << endl;
}
```

```

// 6. Работа специальной функции
Equipment best_pump = find_max_flow_rate_pump(equipment_list);

// 7. Вывод результата
cout << "--- Результат поиска ---" << endl;
cout << "Насос с максимальной производительностью:" << endl;
print_equipment(best_pump);
cout << endl;

return 0;
}

```

## 5. Результаты работы

Вывод программы:

--- Исходный список насосного оборудования ---

ID: 1 | Наименование: Насос центробежный ЦН-500 | Цех: Цех №1 | Год ввода: 2018 | Производительность: 500 м³/ч

ID: 2 | Наименование: Насос шестеренчатый ШН-200 | Цех: Цех №2 | Год ввода: 2020 | Производительность: 200 м³/ч

ID: 3 | Наименование: Насос вакуумный ВН-150 | Цех: Цех №3 | Год ввода: 2019 | Производительность: 150 м³/ч

ID: 4 | Наименование: Насос дозировочный НД-80 | Цех: Цех №1 | Год ввода: 2022 | Производительность: 80 м³/ч

ID: 5 | Наименование: Насос винтовой ВН-350 | Цех: Цех №2 | Год ввода: 2021 | Производительность: 350 м³/ч

--- Результат поиска ---

Насос с максимальной производительностью:

ID: 1 | Наименование: Насос центробежный ЦН-500 | Цех: Цех №1 | Год ввода: 2018 | Производительность: 500 м³/ч

Анализ результатов:

Алгоритм корректно перебрал все элементы массива и сравнил значения поля *flow\_rate*. Максимальное значение (500 м³/ч) принадлежит объекту с ID 1, что и было выведено на экран.

## 6. Вывод

В ходе выполнения лабораторной работы:

- Была создана и использована структура *Equipment* для хранения данных о технологическом оборудовании, что позволило объединить разнотипную информацию (числовые идентификаторы, строки, год выпуска) в единую сущность.
- Освоены способы инициализации объектов структуры (при создании вектора). Реализованы функции для работы со структурами:
- `print_equipment` (передача по константной ссылке для экономии памяти и защиты данных);
- `find_max_flow_rate_pump` (принимает вектор и возвращает структуру).

Успешно решена прикладная задача, характерная для нефтехимического производства - поиск оборудования с экстремальным значением параметра (максимальная производительность насоса).

Таким образом, цель работы достигнута, навык использования структур сформирован.

## 7. Ответы на контрольные вопросы

Контрольные вопросы (для допуска и защиты):

1. Объясните, что такое композиция данных и как структура реализует этот принцип.

**Композиция** - объединение разнотипных данных в одну сложную единицу. **Структура** реализует это через группировку полей в один тип `Equipment`, представляющий целостную сущность.

2. В какой области памяти (стек или куча) будет размещен массив ваших структур, созданный как локальная переменная в `main()`?

**Сам объект вектора (`equipment_list`)** - в **стеке** (локальная переменная). **Данные структур**, хранящиеся в векторе - в **куче** (динамическая память).

3. Предложите, как можно модифицировать программу, чтобы данные об оборудовании загружались из текстового файла. Как при этом должна измениться структура?

Добавить `#include <fstream>` и функцию `loadFromFile()`. Читать файл построчно, парсить значения (например, через запятую), заполнять вектор. **Структуру менять не нужно**, меняется только способ инициализации данных.

4. Почему в реальных проектах для подобных целей часто используют не структуры (`struct`), а классы (`class`)? В чем ключевое различие применительно к вашей задаче?

**Инкапсуляция:** в классе поля по умолчанию **приватные**, доступ через методы. **Контроль данных:** можно добавить проверки в сеттеры. **Гибкость:** легче расширять (наследование, полиморфизм) и поддерживать код. **Структуры** используют для простых "контейнеров" данных, **классы** - для сложной логики и защиты.