

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский ядерный университет «МИФИ»

Обнинский институт атомной энергетики –

филиал федерального государственного автономного образовательного учреждения высшего образования

«Национальный исследовательский ядерный университет «МИФИ»

(ИАТЭ НИЯУ МИФИ)

ОТДЕЛЕНИЕ ЯДЕРНОЙ ФИЗИКИ И ТЕХНОЛОГИЙ

Одобрено на заседании

Ученого совета ИАТЭ НИЯУ МИФИ

Протокол от 24.04.2023 № 23.4

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Информатика

название дисциплины

для направления подготовки

12.03.01 Приборостроение

код и название направления подготовки

образовательная программа

Приборы и методы контроля качества и диагностики

Форма обучения: заочная

г. Обнинск 2023 г.

Область применения

Фонд оценочных средств (ФОС) – является неотъемлемой частью учебно-методического комплекса учебной дисциплины «Информатика» и предназначен для контроля и оценки образовательных достижений обучающихся, освоивших программу данной дисциплины.

Цели и задачи фонда оценочных средств

Целью Фонда оценочных средств является установление соответствия уровня подготовки обучающихся требованиям федерального государственного образовательного стандарта.

Для достижения поставленной цели Фондом оценочных средств по дисциплине «Информатика» решаются следующие задачи:

- контроль и управление процессом приобретения обучающимися знаний, умений и навыков предусмотренных в рамках данного курса;
- контроль и оценка степени освоения компетенций предусмотренных в рамках данного курса;
- обеспечение соответствия результатов обучения задачам будущей профессиональной деятельности через совершенствование традиционных и внедрение инновационных методов обучения в образовательный процесс в рамках данного курса.

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

1.1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Коды компетенций	Результаты освоения ООП Содержание компетенций	Перечень планируемых результатов обучения по дисциплине
ОПК-4	Способен использовать современные информационные технологии и программное обеспечение при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	Знать: -общие понятия теории информации; -основные методы представления и обработки информации в современных ЭВМ. Уметь: -работать и информацией в глобальных компьютерных сетях. Владеть: -современными офисными пакетами, стандартными библиотеками; -навыками работы с информацией в глобальных компьютерных сетях.

1.2. Этапы формирования компетенций в процессе освоения ООП бакалавриата

Компоненты компетенций, как правило, формируются при изучении нескольких дисциплин, а также в немалой степени в процессе прохождения практик, НИР и во время самостоятельной работы обучающегося. Выполнение и защита ВКР являются видом учебной деятельности, который завершает процесс формирования компетенций.

Место дисциплины и соответствующий этап формирования компетенций в целостном процессе подготовки по образовательной программе можно определить по матрице компетенций, которая приводится в Приложении.

Этапы формирования компетенции в процессе освоения дисциплины:

- **начальный** этап – на этом этапе формируются знаниевые и инструментальные основы компетенции, осваиваются основные категории, формируются базовые умения. Студент воспроизводит термины, факты, методы, понятия, принципы и правила; решает учебные задачи по образцу;
- **основной** этап – знания, умения, навыки, обеспечивающие формирование компетенции, значительно возрастают, но еще не достигают итоговых значений. На этом этапе студент осваивает аналитические действия с предметными знаниями по дисциплине, способен самостоятельно решать учебные задачи, внося коррективы в алгоритм действий, осуществляя коррекцию в ходе работы, переносит знания и умения на новые условия;
- **завершающий** этап – на этом этапе студент достигает итоговых показателей по заявленной компетенции, то есть осваивает весь необходимый объем знаний, овладевает всеми умениями и навыками в сфере заявленной компетенции. Он способен использовать эти знания, умения, навыки при решении задач повышенной сложности и в нестандартных условиях.

Этапы формирования компетенций в ходе освоения дисциплины отражаются в тематическом плане (см.п. 4 рабочей программы дисциплины).

1.3. Паспорт фонда оценочных средств по дисциплине

№ п/п	Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции (или её части) / и ее формулировка	Наименование оценочного средства
Текущий контроль, 1 курс			
1.	Первый семестр, Темы 1-6	ОПК-4	Сдача отчетов по лабораторным работам
2.	Первый семестр, Темы 7-12	ОПК-4	Сдача отчетов по лабораторным работам
Промежуточный контроль, 1 курс			
	зачет	ОПК-4	Вопросы на зачет
Всего:			
Текущий контроль, 1 курс			
1.	Второй семестр, Темы 1-6	ОПК-4	Сдача отчетов по лабораторным работам
2.	Второй семестр, Темы 7-12	ОПК-4	Сдача отчетов по лабораторным работам
Промежуточный контроль, 1 курс			
	зачет	ОПК-4	Вопросы на зачет
Всего:			

2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Конечными результатами освоения программы дисциплины являются сформированные когнитивные дескрипторы «знать», «уметь», «владеть», расписанные по отдельным компетенциям, которые приведены в п.1.1. Формирование этих дескрипторов происходит в процессе изучения дисциплины по этапам в рамках различного вида учебных занятий и самостоятельной работы.

Выделяются три уровня сформированности компетенций на каждом этапе: пороговый, продвинутый и высокий.

Уровни	Содержательное описание уровня	Основные признаки выделения уровня	БРС, % освоения	ECTS/Пятибалльная шкала для оценки экзамена/зачета
Высокий <i>Все виды компетенций сформированы на высоком уровне в соответствии с целями и задачами дисциплины</i>	Творческая деятельность	<i>Включает нижестоящий уровень.</i> Студент демонстрирует свободное обладание компетенциями, способен применить их в нестандартных ситуациях: показывает умение самостоятельно принимать решение, решать проблему/задачу теоретического или прикладного характера на основе изученных методов, приемов, технологий	90-100	A/ Отлично/ Зачтено
Продвинутый <i>Все виды компетенций сформированы на продвинутом уровне в соответствии с целями и задачами дисциплины</i>	Применение знаний и умений в более широких контекстах учебной и профессиональной деятельности, нежели по образцу, большей долей самостоятельности и инициативы	<i>Включает нижестоящий уровень.</i> Студент может доказать владение компетенциями: демонстрирует способность собирать, систематизировать, анализировать и грамотно использовать информацию из самостоятельно найденных теоретических источников и иллюстрировать ими теоретические положения или обосновывать практику применения.	85-89	B/ Очень хорошо/ Зачтено
			75-84	C/ Хорошо/ Зачтено
Пороговый <i>Все виды компетенций сформированы на пороговом уровне</i>	Репродуктивная деятельность	Студент демонстрирует владение компетенциями в стандартных ситуациях: излагает в пределах задач курса теоретически и практически контролируемый материал.	65-74	D/Удовлетворительно/ Зачтено
			60-64	E/Посредственно /Зачтено
Ниже порогового	Отсутствие признаков порогового уровня: компетенции не сформированы. Студент не в состоянии продемонстрировать обладание компетенциями в стандартных ситуациях.		0-59	Неудовлетворительно/ Зачтено

Оценивание результатов обучения студентов по дисциплине осуществляется по регламенту текущего контроля и промежуточной аттестации.

Критерии оценивания компетенций на каждом этапе изучения дисциплины для каждого вида оценочного средства и приводятся в п. 4 ФОС. Итоговый уровень сформированности компетенции при изучении дисциплины определяется по таблице. При этом следует понимать, что граница между уровнями для конкретных результатов освоения образовательной программы может смещаться.

Уровень сформированности компетенции	Текущий контроль	Промежуточная аттестация
высокий	высокий	высокий
	<i>продвинутой</i>	<i>высокий</i>
	<i>высокий</i>	<i>продвинутой</i>
продвинутой	<i>пороговый</i>	<i>высокий</i>
	<i>высокий</i>	<i>пороговый</i>
	продвинутой	продвинутой
	<i>продвинутой</i>	<i>пороговый</i>
	<i>пороговый</i>	<i>продвинутой</i>
пороговый	пороговый	пороговый
ниже порогового	пороговый	ниже порогового
	ниже порогового	-

3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков или опыта деятельности, характеризующих этапы формирования компетенций.

Рейтинговая оценка знаний является интегральным показателем качества теоретических и практических знаний и навыков студентов по дисциплине и складывается из оценок, полученных в ходе текущего контроля и промежуточной аттестации.

Текущий контроль в семестре проводится с целью обеспечения своевременной обратной связи, для коррекции обучения, активизации самостоятельной работы студентов.

Промежуточная аттестация предназначена для объективного подтверждения и оценивания достигнутых результатов обучения после завершения изучения дисциплины.

Текущий контроль осуществляется два раза в семестр: контрольная точка № 1 (лабораторные работы 1-6) и контрольная точка № 2 (лабораторные работы 7-12).

Результаты текущего контроля и промежуточной аттестации подводятся по шкале балльно-рейтинговой системы.

Вид контроля	Этап рейтинговой системы Оценочное средство	Балл	
		Минимум	Максимум
Текущий	Контрольная точка № 1		
	Лабораторная работа 1	3	5
	Лабораторная работа 2	3	5
	Лабораторная работа 3	3	5
	Лабораторная работа 4	3	5
	Лабораторная работа 5	3	5
	Лабораторная работа 6	3	5
	Контрольная точка № 2		
	Лабораторная работа 7	4	5
	Лабораторная работа 8	4	5
	Лабораторная работа 9	4	5

	Лабораторная работа 10	3	5
	Лабораторная работа 11	3	5
	Лабораторная работа 12	4	5
Промежуточный	Зачет		
	Вопросы на зачет	20	40
ИТОГО по дисциплине		60	100

Определение бонусов и штрафов

Бонусы: поощрительные баллы студент получает к своему рейтингу в конце семестра за активную и регулярную работу на занятиях 5 баллов (но суммарно за семестр не больше чем 60)

Штрафы: за несвоевременное участие в коллоквиуме и контрольной работе максимальная оценка может быть снижена на 20%.

Процедура оценивания знаний, умений, владений по дисциплине включает учет успешности по всем видам заявленных оценочных средств. По окончании освоения дисциплины в 5 семестре проводится промежуточная аттестация в виде зачета, что позволяет оценить совокупность приобретенных в процессе обучения компетенций. При выставлении итоговой оценки применяется балльно-рейтинговая система оценки результатов обучения.

Зачет предназначен для оценки работы обучающегося в течение всего срока изучения дисциплины и призван выявить уровень, прочность и систематичность полученных обучающимся теоретических знаний и умений приводить примеры практического использования знаний (например, применять их в решении практических задач), приобретения навыков самостоятельной работы, развития творческого мышления.

Оценка сформированности компетенций на зачете для тех обучающихся, которые пропускали занятия и не участвовали в проверке компетенций во время изучения дисциплины, проводится после индивидуального собеседования с преподавателем по пропущенным или не усвоенным обучающимся темам с последующей оценкой самостоятельно усвоенных знаний на зачете.

В случае если суммарная оценка за текущий контроль позволяет проставить итоговую оценку за обучение в семестре автоматически, это делается по обоюдному согласию лектора и студента.

4. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков

4.1.1. Зачет

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«Национальный исследовательский ядерный университет «МИФИ»

Обнинский институт атомной энергетики –

филиал федерального государственного автономного образовательного учреждения высшего профессионального образования «Национальный исследовательский ядерный университет «МИФИ»

Отделение интеллектуальных кибернетических систем

Направление	<u>12.03.01 «Приборостроение»</u>
Профиль	<u>«Приборы и методы контроля качества и диагностики»</u>
Дисциплина	<u>«Информатика»</u>

ВОПРОСЫ К ЗАЧЕТУ

1. Что такое Google Classroom?
2. Операционные системы. Linux, Windows, Mac OS, Android, IOS, Windows Mobile.
3. Сервисы для создания онлайн почтового ящика?
4. Онлайн инструменты для офиса?
5. Программные продукты с открытым исходным кодом.
6. Причины возникновения и преимущества использования продуктов с открытым исходным кодом.
7. Свободные программные продукты в приборах и системах диагностики, примеры проектов.
8. Операционная система Linux. История возникновения.
9. Проект GNU. Понятие дистрибутива Linux. Лицензия GPL и метод copyleft.
10. Стандартные и служебные программы. Преимущества использования ОС Linux в сравнении с ОС Windows
11. Понятие пользователей и групп пользователей в ОС Linux, понятие суперпользователя root. Преимущества подобного разделения пользователей для администрирования и использования системы, сравнение с концепцией пользователей и групп в ОС Windows.
12. Desktop Environment(DE) – современный рабочий стол для Linux. Различные виды DE, основные характеристики и отличия.
13. Офисные приложения. Текстовые редакторы.
14. Офисные приложения. Редакторы для работы с таблицами.
15. Офисные приложения. Редакторы для работы с презентациями.
16. Офисные приложения. Редакторы для работы с изображениями.
17. АСУ – Предприятия,
18. 1С Bitrix 24

19. Облачные системы управления предприятием
20. Gmail, Создание почты, управление письмами и рассылками
21. Программы для управления контактами и задачами
22. Системы управления временем – календарь
23. Формирование групповых рассылок и форумов
24. Блогосфера, ведение личного блога и AdSense
25. Облачные инструменты для работы с текстовыми документами
26. Облачные инструменты для работы с таблицами
27. Облачные инструменты для работы с презентациями
28. Облачные инструменты для работы с формами
29. Облачные инструменты для работы с изображениями
30. Инструменты по разработке сайтов
31. Облачные инструменты ГИС
32. Облачные инструменты по работе с фотографиями
33. Какие библиотеки Python используются для анализа данных и работы с массивами?
Опишите их основные функции.
34. Что такое Google Colaboratory? Опишите основные функции и предназначение.
35. Инструменты для поиска научно-технической литературы
36. Среда разработки Microsoft Visual Studio.
37. Что такое условный оператор?
38. Для чего нужен цикл while и как он реализуется в языках программирования?
39. Какие бывают типы данных в программировании?
40. Для чего нужен цикл for и как он реализуется в языках программирования?
41. Что такое массивы? Какие бывают массивы?
42. Что такое рекурсия? Как создаются функции в языках программирования?
43. Какие бывают структуры для хранения данных?
44. Опишите стандартные алгоритмы STL?
45. Что такое ООП? Основные особенности ООП?
46. Что такое классы, методы и функции в ООП?

На зачёте студенту предлагаются три вопроса из представленного перечня – образец:

1. Какие возможности есть у google maps? (Знание)
2. Как сгруппировать элементы рисунка в единое целое и повернуть изображение? (Умение)
3. Какой тип адресации используется в EXCEL по умолчанию? (Владение)

б) критерии оценивания компетенций (результатов):

полнота ответа на вопросы.

в) описание шкалы оценивания:

Критерии оценки	Оценка
<p>Студент должен</p> <ul style="list-style-type: none"> – показать глубокое и прочное усвоение знаний по тематике вопросов; – построить грамотный, логический и исчерпывающий ответ; – правильно и полно формулировать определения; – привести примеры, иллюстрирующие теоретическую часть ответов; – сделать обоснованные выводы по излагаемому материалу 	36 – 40
<p>Студент должен:</p> <ul style="list-style-type: none"> – показать достаточно полное знание тематики вопросов; – дать достаточно грамотный и логический ответ; – правильно формулировать определения; – привести примеры для теоретической части ответов; – сделать выводы по излагаемому материалу. 	30 – 35
<p>Студент должен</p> <ul style="list-style-type: none"> – показать знания тематической части вопросов; – дать правильный ответ на большую часть вопросов; – знать основную рекомендованную учебную литературу по дисциплине. 	25 – 29
<p>Студент</p> <ul style="list-style-type: none"> – не знает значительной части материала вопросов; – не владеет понятийным аппаратом дисциплины; – дает более 50% ошибочных ответов на вопросы; – не умеет делать выводы по излагаемому материалу. 	≤ 24 (незачтено)

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«Национальный исследовательский ядерный университет «МИФИ»

Обнинский институт атомной энергетики –

филиал федерального государственного автономного образовательного учреждения высшего профессионального образования «Национальный исследовательский ядерный университет «МИФИ»

Отделение интеллектуальных кибернетических систем

Комплект лабораторных работ

по дисциплине «Информатика»

Первый семестр

Лабораторная работа 1

1. Учащимся необходимо перейти в google classroom по ссылке <https://classroom.google.com>
2. Войти в корпоративный аккаунт на домене oiate.ru и заново обновить страницу.
3. Получить приватный код приглашения и вступить в google класс по предмету «Информатика», используя символ + (**Присоединиться к классу**).
4. Настроить google аккаунт нажав по иконке вашего аккаунта и выбрав в выпадающем меню **Управление аккаунтом Google**.
5. Добавьте номер телефона для лучшей безопасности вашего аккаунта, синхронизируйте гугл диск и ваш телефон с данным аккаунтом, скачав официальное приложение Google Classroom.
6. Оформить в электронном документе отчет по лабораторной работе с ответами на контрольные вопросы и сдать задание в google classroom.

Контрольные вопросы

1. Что такое google classroom?
2. Перечислите основные возможности google classroom?
3. Перечислите и опишите какие дополнительные сервисы google используются в google classroom?

Лабораторная работа 2

1. Учащимся необходимо перейти на сайт gmail по ссылке <https://mail.google.com/>
2. Войти в корпоративный аккаунт на домене oiate.ru введя логин и пароль.
3. Выполнить настройку интерфейса, используя клавишу шестеренки в правом верхнем углу, как показано ниже.



4. Выберите Компактный интерфейс и понравившуюся тему.
5. Выберите Вид по умолчанию и область просмотра (Справа).
6. Поставьте галочку в пункте «Объединение писем в цепочки» - В виде цепочек.
7. Сделайте скриншоты изменений и обновите вашу страницу в браузере.
8. Откройте заново значок шестеренки (Настройки) и выберите первый пункт «Все настройки»
9. Во вкладке «Общие» установите значение в строке «Количество цепочек на одной странице» - 50.
10. В строке «Отмена отправки» установите «Промежуток времени для отмены отправки» - 30 сек.
11. В строке «Стиль текста по умолчанию:» установите «Обычный» размер шрифта «Без засечек».
12. В строке «Цепочки писем» отметьте «Включить».
13. В строке «Напоминания» поставьте галочки в обоих пунктах «Напоминать...».
14. Включите «Быстрый ответ» в строке «Быстрый ответ».
15. Включите «Уведомления о новых письмах» в строке «Уведомления на рабочем столе».
16. В строке «Подпись» напишите Вашу подпись (Пример С уважением, Иванов Иван Иванович, студент гр. АЭС-С19, ИАТЭ НИЯУ МИФИ) и выставите ее по умолчанию «В новых письмах» и «В ответах и при пересылке».
17. В строке «Значки персональных писем» поставьте галочку напротив «Включить».
18. Сделайте скриншоты, сохраните изменения и перейдите во вкладку «Фильтры и заблокированные адреса».
19. Во вкладке «Фильтры и заблокированные адреса» выберите пункт «Создать фильтр».
20. Создайте фильтр от своей домашней почты, вписав ее полное название в строку «От кого». После этого нажмите «Создать» и согласитесь с выпадающим окном.
21. В следующей открывшейся вкладке поставьте галочки рядом с пунктами «Никогда не отправлять в спам» и «Применить ярлык». Нажмите рядом с последним пунктом на выпадающее меню «Выберите ярлык» -> «Создать ярлык».
22. В открывшемся окне «Новый ярлык» введите название ярлыка «Home email» и выберите под каким ярлыком вы хотите его разместить. После окончания

всех действий нажмите «Создать», а затем «Создать фильтр». Сделайте скриншот созданного фильтра и ярлыка и вставьте его в отчет.

23. Во вкладке «Чат и встречи» включите Hangouts и выберите «Показать раздел Meet в главном меню».

24. Во вкладке «Расширенные» включите «Значок непрочитанных сообщений». Сделайте скриншот, сохраните изменения.

25. Напишите сообщение вашему преподавателю с вашим ФИО, названием группы и датой рождения, сделайте скриншот и вставьте в отчет.

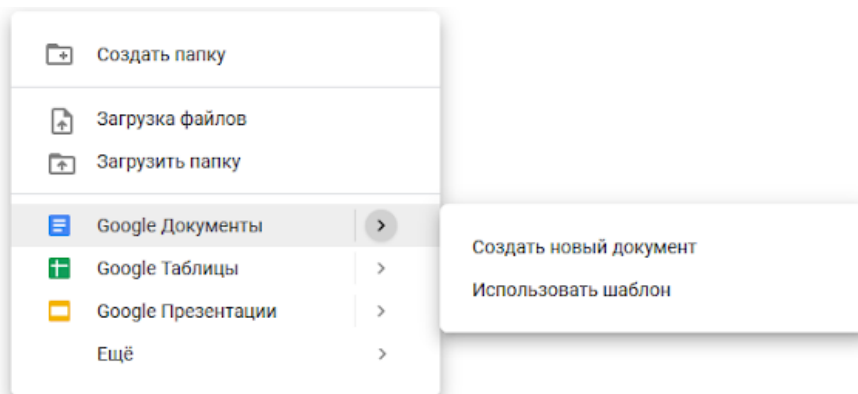
26. Сделайте массовую рассылку письма всем вашим одноклассникам, сделайте скриншоты и вставьте в отчет.

Контрольные вопросы

1. Что такое Gmail?
2. Опишите основные преимущества и особенности Gmail?
3. Какие ярлыки присутствуют в Gmail?

Лабораторная работа 3

1. Войдите в свой корпоративный аккаунт google на домене oiate.
2. Перейдите на облачное хранилище ваших данных (гугл диск) по ссылке <https://drive.google.com/>
3. Создайте папку на гугл диске под названием «Лабораторные работы».
4. В созданной папке создайте новый гугл документ, кликнув правой кнопкой мыши на пустом поле и выбрав Google Документы->Создать новый документ.



5. Откройте шаблон резюме, который должен был выслать ваш преподаватель.

6. Создайте собственное резюме, используя шаблон. По необходимости можно убирать некоторые поля и добавлять свои.

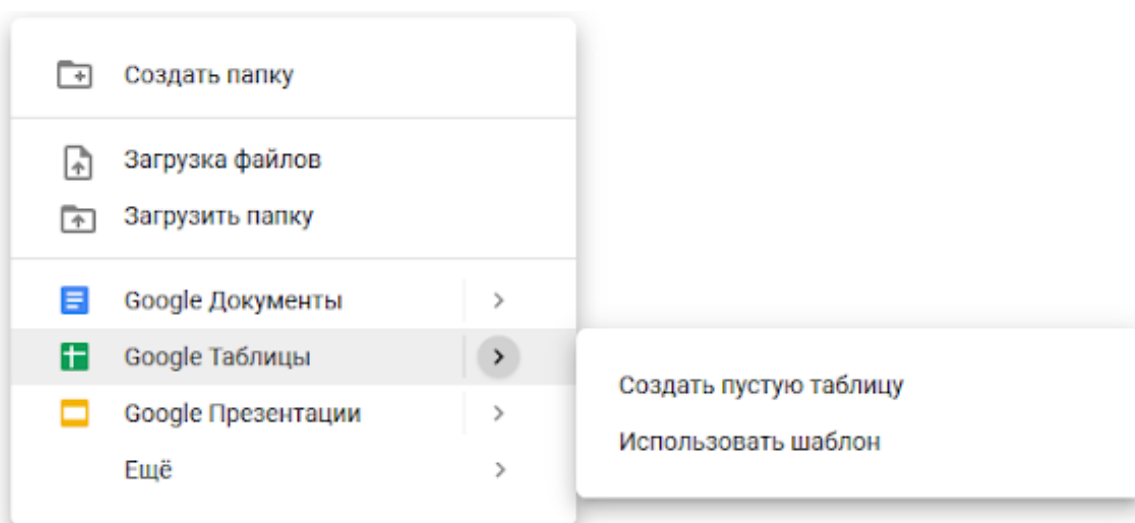
7. По окончании разработки резюме сдайте задание, выслав документ вашему преподавателю, а также отчет с ответами на контрольные вопросы.

Контрольные вопросы

1. Что такое Google документы?
2. Опишите основные преимущества и особенности Google документов?
3. В чем отличие Google документов от аналогичных офисных редакторов и программ?

Лабораторная работа 4

1. Войдите в свой корпоративный аккаунт google на домене oiate.
2. Перейдите на облачное хранилище ваших данных (гугл диск) по ссылке <https://drive.google.com/>
3. Перейдите в созданную на прошлой лабораторной работе папку «Лабораторные работы».
4. В созданной папке создайте новую таблицу, кликнув правой кнопкой мыши на пустом поле и выбрав Google Таблицы->Создать пустую таблицу.



5. Заполните ячейки случайными числами следующим образом:
 - Заполните ячейки **A1:A10** случайными числами в диапазоне **от 0 до 1**
 - Заполните ячейки **B1:B10** случайными числами в диапазоне **от 1 до 100**
 - Заполните ячейки **C1:C10** случайными числами в диапазоне **от 2 до 50**
 - Заполните ячейки **D1:D10** случайными числами в диапазоне **от -10 до 10**
 - Заполните ячейки **E1:E10** случайными числами в диапазоне **от -100 до 100**
 - Заполните ячейки **F1:F10** случайными числами в диапазоне **от 0 до 600**

Примеры формул для заполнения:

СЛЧИС() - Числа в диапазоне от 0 до 1

СЛЧИС()*100 - Числа в диапазоне от 0 до 100

СЛЧИС()*90 - Числа в диапазоне от 0 до 90

СЛЧИС()*-10 - Числа в диапазоне от -10 до 0

СЛЧИС()*(12-2)+2 Числа в диапазоне от 2 до 12

СЛЧИС()*(8-(-8))-8 - Числа в диапазоне от -8 до 8

Для указания диапазона выбора чисел используется формула: $СЛЧИС()*(b-a)+a$, где **a** и **b** - границы диапазона.

6. Отступите одну строку ниже и создайте 7 столбцов с именами: максимальное число, минимальное число, сумму, среднее значение, медиана, мода, стандартное отклонение.

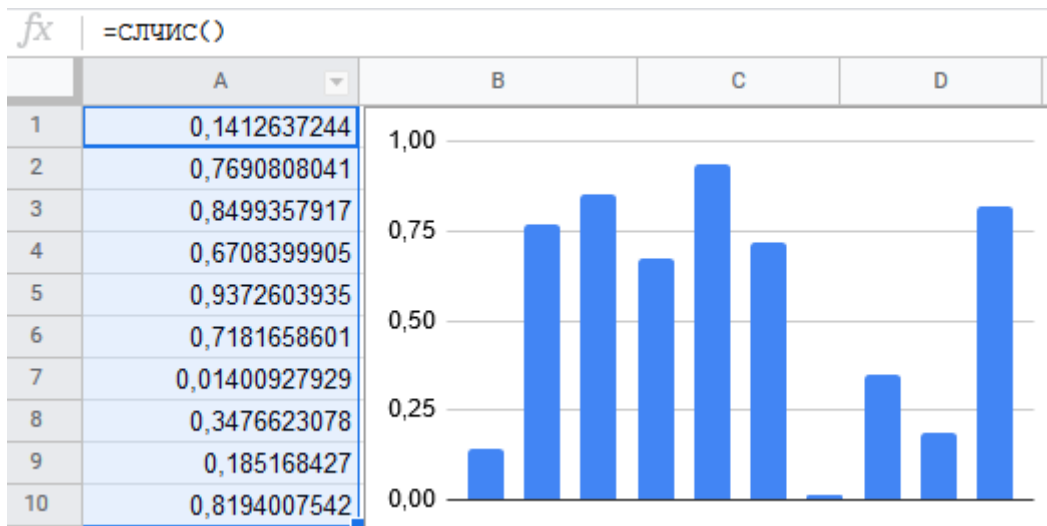
7. Для каждого столбца из исходной сгенерированной таблицы рассчитайте каждое из этих значений и запишите их в каждый новый столбец. Можно использовать следующие функции для определения: максимального (МАКС), минимального числа (МИН), суммы (СУММ), среднего значения (СРЗНАЧ), медианы (МЕДИАНА()), стандартного отклонения (СТАНДОТКЛОН()). Приведем пример расчета максимального числа.

fx		=МАКС(A1:A10)	
	A	B	
1	0,8077092265	62,16396136	
2	0,7701741022	94,44072791	
3	0,5513735201	1,068665757	
4	0,2575763143	56,68727609	
5	0,8697886601	35,69392892	
6	0,1184659254	21,60298093	
7	0,3664571588	44,73830105	
8	0,6261153581	59,5125732	
9	0,20845911	97,77002423	
10	0,4310469406	53,41644768	
11			
12	Максимальное число	Минимальное число	
13	0,8697886601	Для 1-ого столбца	
14	Для 2-ого столбца	1,068665757	
15	Для 3-его столбца	Для 3-его столбца	

8. Заполните все столбцы рассчитанными значениями.

9. Сделайте скриншот исходной таблицы и скриншот полученных (рассчитанных) значений.

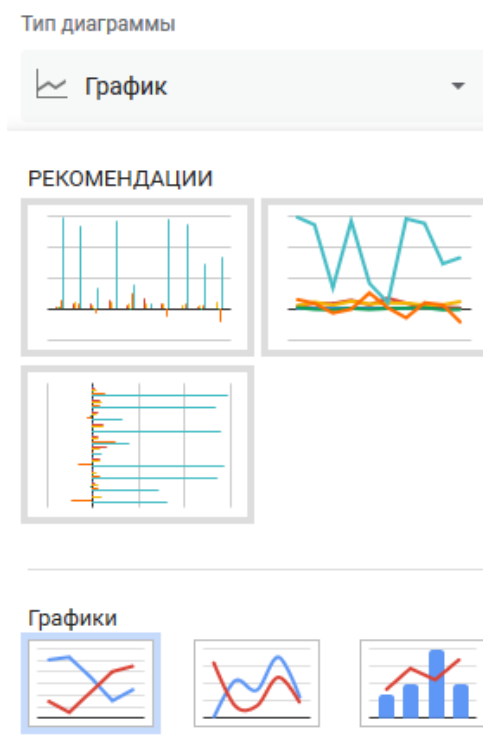
10. Для каждого столбца из исходных данных постройте гистограммы распределения. Для построения гистограммы необходимо выделить нужный столбец, открыть вкладку «Вставка» и выбрать пункт «Диаграмма». Пример гистограммы, показан на рисунке ниже.



11. Выполните пункт 10 для каждого столбца исходных данных и сделайте скриншота каждой диаграммы с соответствующим столбцом (как на рисунке выше) и вставьте в отчет.

12. Выделите все столбцы и постройте общую гистограмму. Диаграмму поместить на отдельном листе, который назвать «ГИСТОГРАММА». Сделайте скриншот, вставьте в отчет.

13. После этого снова выделите все столбцы, нажмите меню «Вставка»-«Диаграмма» и в открывшемся справа меню выберите «Тип диаграммы» - «Графики», как показано на рисунке ниже. Диаграмму поместить на отдельном листе, который назвать «ГРАФИКИ». Сделайте скриншот и вставьте в отчет.



14. Аналогично описанным выше действиям построить следующие диаграммы: диаграмма с областями, столбчатые, линейчатые, круговые, точечные.

Поместите каждую диаграмму в отдельный лист и назовите одноименным названием с диаграммой. Сделайте скриншоты каждой диаграммы и вставьте в отчет с пояснениями.

15. Напишите вывод о проделанной работе и ответьте на контрольные вопросы.

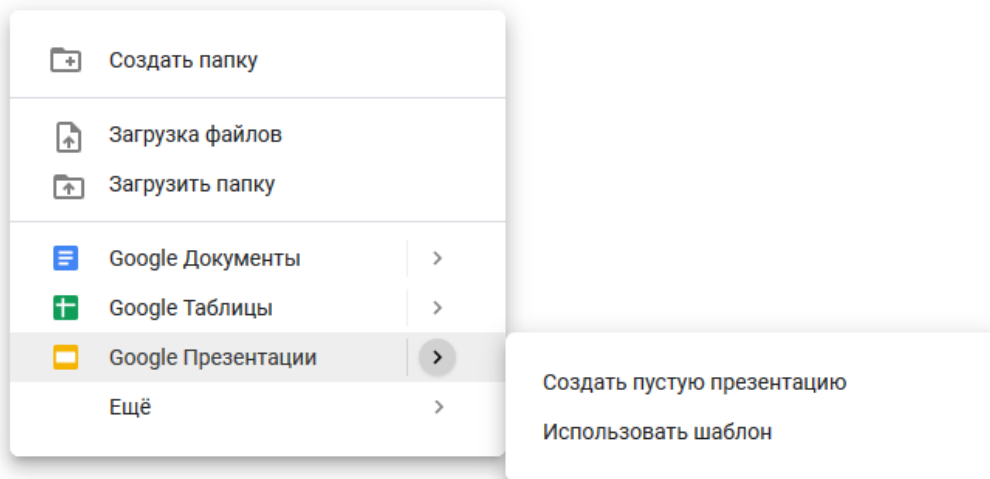
16. Оформите отчет по лабораторной работе в формате .docx. Вместе с отчетом к заданию прикрепите полученную таблицу со всеми результатами и графиками.

Контрольные вопросы

1. Что такое Google Sheets?
2. Опишите основные преимущества и особенности Google таблиц?
3. В чем отличие Google таблиц от аналогичных офисных редакторов и программ?
4. Какие диаграммы можно строить с помощью Google таблиц?

Лабораторная работа 5

1. Войдите в свой корпоративный аккаунт google на домене oiate.
2. Перейдите на облачное хранилище ваших данных (гугл диск) по ссылке <https://drive.google.com/>
3. Перейдите в ранее созданную папку «Лабораторные работы».
4. В созданной папке создайте новую таблицу, кликнув правой кнопкой мыши на пустом поле и выбрав Google Презентации->Создать пустую презентацию.



5. Создать и оформить презентацию на предложенные темы:
 1. История развития информатики.
 2. Понятие информации и информационного пространства.
 3. История развития ЭВМ.
 4. Создание, обработка и хранение информации в технических устройствах.
 5. Негативное воздействие компьютера на здоровье человека и способы защиты.

6. Спам и его разновидности. Методы борьбы со спамом.
7. Антивирусы.
8. Устройство современного ПК
9. История развития кибернетики
10. Что такое искусственный интеллект?
11. Большие данные (Big Data). Системы обработки больших данных.
12. История развития языков программирования
13. Микропроцессоры и микроконтроллеры
14. Квантовые компьютеры
15. Программное обеспечение
16. Операционные системы. Виды операционных систем.
17. Утилиты. Прикладные программы.
18. Текстовые редакторы.
19. СУБД
20. Программные средства специального назначения
21. Файловые системы
22. Компьютерный дизайн.
23. Системы автоматизированного проектирования
24. Информационный язык как средство представления информации
25. Сеть Wi-Fi и ее основные возможности.
26. Кибербезопасность.
27. Правонарушения в сфере информационных технологий.

6. В презентации обязательно должны быть картинки, заголовки, перечисления, видео.

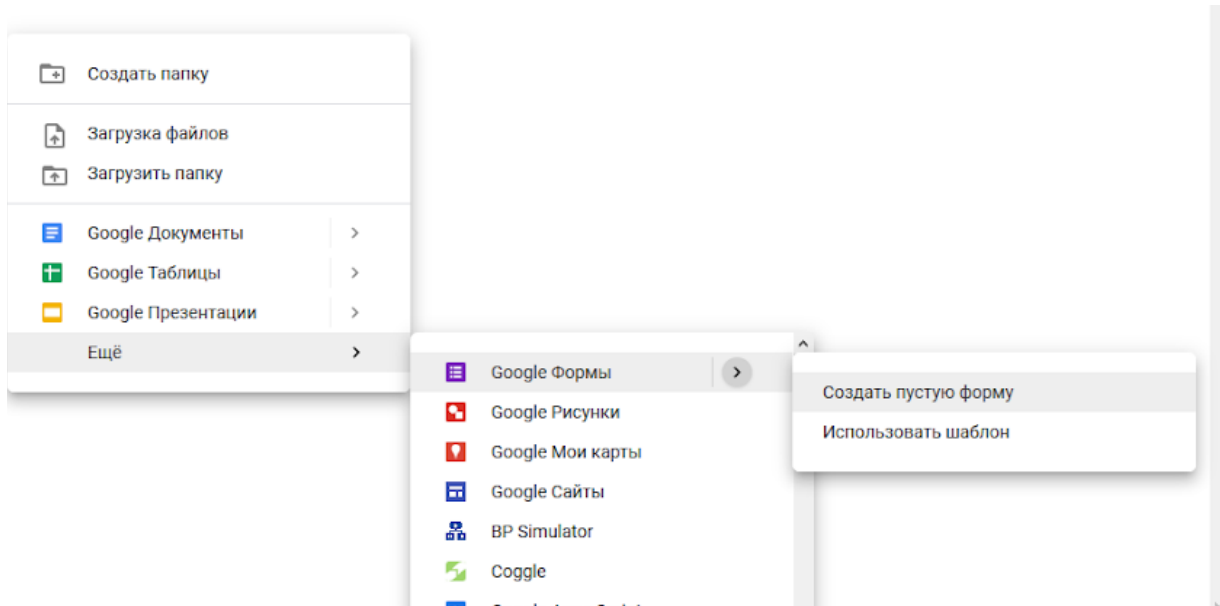
7. Выслать презентацию в онлайн-класс вместе с отчетом и ответами на контрольные вопросы и сдать работу.

Контрольные вопросы

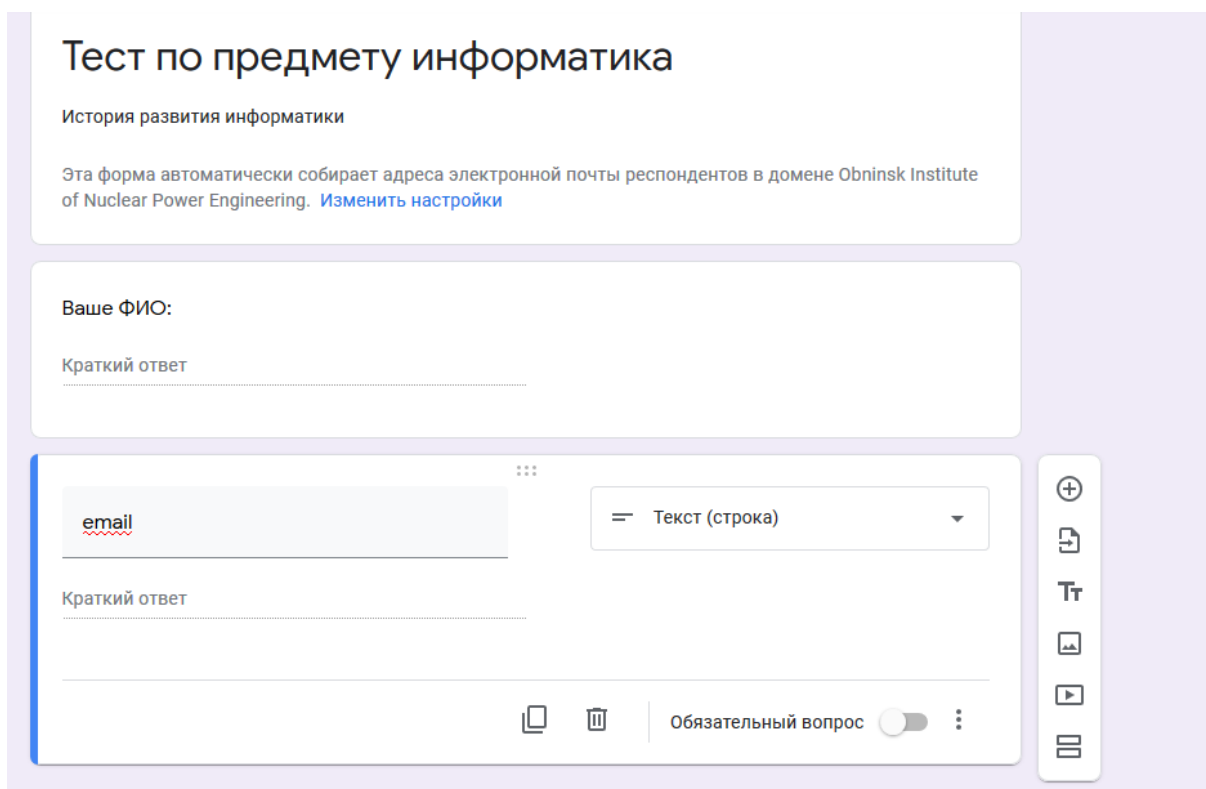
1. Что такое Google Slides?
2. Опишите основные преимущества и особенности Google Slides?
3. В чем отличие Google Slides от аналогичных офисных редакторов и программ?


Лабораторная работа 6

1. Войдите в свой корпоративный аккаунт google на домене oiate.
2. Перейдите на облачное хранилище ваших данных (гугл диск) по ссылке <https://drive.google.com/>
3. Перейдите в ранее созданную папку «Лабораторные работы».
4. В созданной папке создайте новую таблицу, кликнув правой кнопкой мыши на пустом поле и выбрав Еще->Google Презентации->Создать пустую презентацию.



5. Создать тест-опрос с вариантами ответа по тематикам из докладов-презентаций лабораторной работы 5. В тесте обязательно вначале должно быть название, строка для ввода ФИО, строка ввода email.



6. По окончании разработки теста зайдите в настройки, используя значок  в правом верхнем углу и перейдите в настройки «Презентация». Поставьте в каждом пункте галочки, как показано на рисунке ниже.

Настройки

Общие

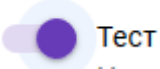
Презентация

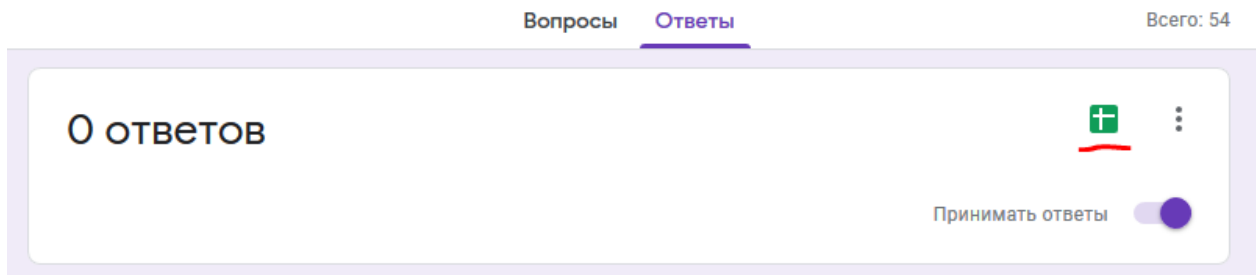
Тесты

- Показывать ход выполнения
- Перемешать вопросы
- Показывать ссылку для повторного заполнения формы

Текст подтверждения:

Ответ записан

7. Перейдите в пункт Тесты и активируйте ползунок напротив названия «Тест».  . Сохраните изменения.
8. Разошлите ссылку на тест вашим одноклассникам.
9. После того, как тест будет пройден, перейдите во вкладку ответы и нажмите на значок таблицы, чтобы сформировать таблицу с ответами и баллами.



10. Сохраните данную таблицу. Сделайте отчет по лабораторной работе.
11. Сдайте лабораторную работу в виде трех файлов: отчет в текстовом формате .docx, google форму с тестом и таблицы с ответами ваших одноклассников.

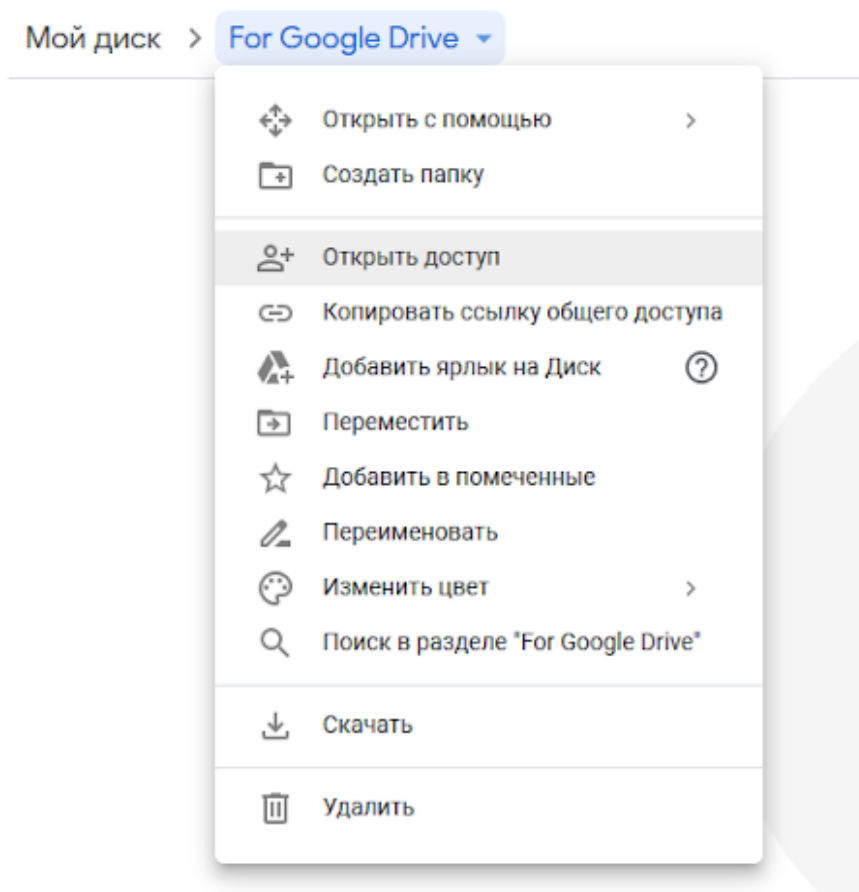
Контрольные вопросы

1. Что такое Google Формы?
2. Опишите основные преимущества и особенности Google Форм?
3. Для чего можно использовать Google Формы?

Лабораторная работа 7

1. Войдите в свой корпоративный аккаунт google на домене oiate.
2. Перейдите на облачное хранилище ваших данных (гугл диск) по ссылке <https://drive.google.com/>

3. Перейдите в созданную ранее папку «Лабораторные работы».
4. Создайте новую папку под названием «For Google Drive» и откройте ее двойным щелчком мыши.
5. Создайте новый текстовый Google Документ и назовите его «Test_project».
6. Затем перейдите обратно в Google Drive в папку «For Google Drive» и откройте доступ ко всей вашей папке. Для этого нужно нажать на стрелочку вниз возле названия папки и выбрать «Открыть доступ».



7. В открывшемся окне добавьте несколько ваших одноклассников и вышлите им уведомления на почту в виде ссылки, которую можно скопировать ниже.
8. Синхронизируйте Google Drive с вашим мобильным устройством, скачав приложение на ваш телефон и авторизовавшись в домене oiate. Сделайте скриншот и вставьте в отчет.
9. Сдайте лабораторную работу в виде текстового документа .docx с ответами на контрольные вопросы.

Контрольные вопросы

1. Что такое Google Drive?
2. Опишите основные преимущества и особенности Google Drive?
3. С какими сервисами возможна интеграция Google Drive?

Лабораторная работа 8

1. Войдите в свой корпоративный аккаунт google на домене oiate.
2. Перейдите в ваш онлайн календарь по ссылке <https://calendar.google.com>
3. В меню слева находятся все ваши календари. Прокрутите меню вниз и найдите надпись «Другие календари». Другие календари + ^
4. Нажмите на значок «+» и затем «Создать календарь».
5. Введите название и описание календаря. Назовите календарь своим именем вместе с названием группы и нажмите «Создать Календарь».

Создать календарь

Название

Иванов Иван Иванович

Описание

Расписание на 2020 год|

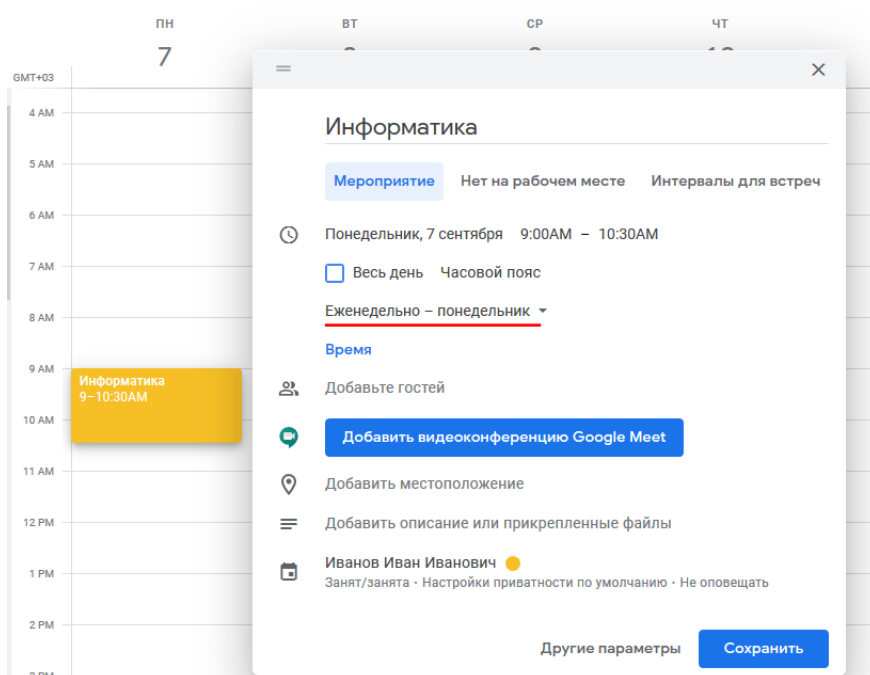
Часовой пояс

(GMT+03:00) Москва, стандартное время - Мос... ▾

6. После создания календаря он отобразится в меню слева. Перед тем как создать расписание, необходимо выполнить общую настройку сервиса.
7. В меню слева перейдите во вкладку «Общие». Найдите пункт «Мероприятия» и выставите «Продолжительность по умолчанию» - 90 минут (2 академических часа).
8. В меню «Режим просмотра» поставьте галочку в пункте «Показывать номера недель» и «Уменьшать яркость прошедших мероприятий». В этом же меню выберите «Первый день недели» - «Понедельник», а «Персонализированный режим просмотра» - «7 дней».
9. В меню «Рабочее время» поставьте галочку «Включить рабочее время». Завершите настройку общих параметров календаря.
10. Выйдите в предыдущее меню где были все ваши календари с помощью стрелочки в верхнем левом углу «Назад».
11. Уберите галочки со всех доступных календарей, кроме созданного вами на этой лабораторной работе.

- Дни рождения
- ДПО-MLBD-2020-1 Прог...
- Задачи
- Защита интеллектуаль...
- Иванов Иван Иванович

12. Заполните календарь занятиями из расписания. Для создания нового предмета в календаре необходимо выбрать с помощью мыши дату и день недели, написать название предмета в календаре а затем выбрать повторение данного предмета, согласно вашему расписанию. По окончании заполнения всех пунктов нажмите «Сохранить», как в примере ниже.



13. Полностью заполните ваше расписание, сделайте скриншоты.
14. Синхронизируйте календарь с вашим мобильным устройством (скачайте приложение на ваш телефон).
15. Сдайте лабораторную работу в виде текстового документа .docx с скриншотами на задания и ответами на контрольные вопросы.

Контрольные вопросы

1. Для чего используется Google Calendar?
2. Опишите основные преимущества и особенности Google Календарь?

Лабораторная работа 9

1. Войдите в свой корпоративный аккаунт google на домене oiate.
2. Перейдите во вкладку создания сайтов по ссылке <https://sites.google.com/new>
3. Для создания нового сайта нажмите знак «+» «Пустой файл».
4. В открывшемся окне добавьте заголовок сайта.

5. Измените тему сайта по усмотрению на вкладке «Темы» в меню справа.
6. Добавьте материал на сайт используя «Текстовое поле» в меню справа. Текст вы можете взять из ваших докладов по теме презентаций в лабораторной работе 5, например со слайда «Введение».
7. Добавьте новую страницу на слайд используя меню справа и вкладку «Страницы» нажав на знак «+» внизу. Назовите страницу «Изображения».
8. Добавьте на созданную страницу несколько изображений с комментариями из презентации лабораторной работы 5.
9. Добавьте новую страницу на слайд используя меню справа и вкладку «Страницы» нажав на знак «+» внизу. Назовите страницу «Документы».
10. Добавьте на созданную страницу вашу презентацию и резюме.
11. Выполните оформление страниц и подготовьте сайт к сдаче вашему преподавателю. Откройте доступ для преподавателя используя значок «Открыть доступ» и введя электронную почту вашего преподавателя.



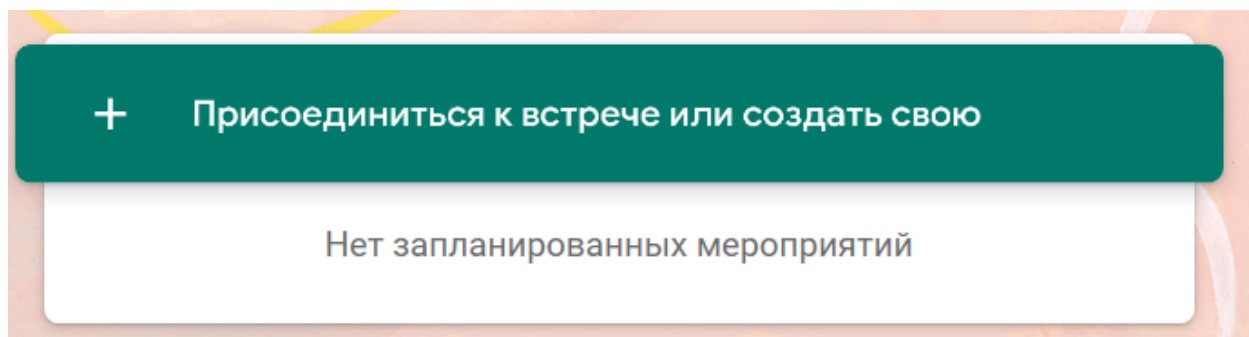
12. Сдайте лабораторную работу в виде текстового документа .docx с скриншотами на задания и ответами на контрольные вопросы и ссылкой на созданный сайт.

Контрольные вопросы

1. Что такое и для чего используется Google Sites?
2. Опишите основные преимущества и особенности Google Sites?

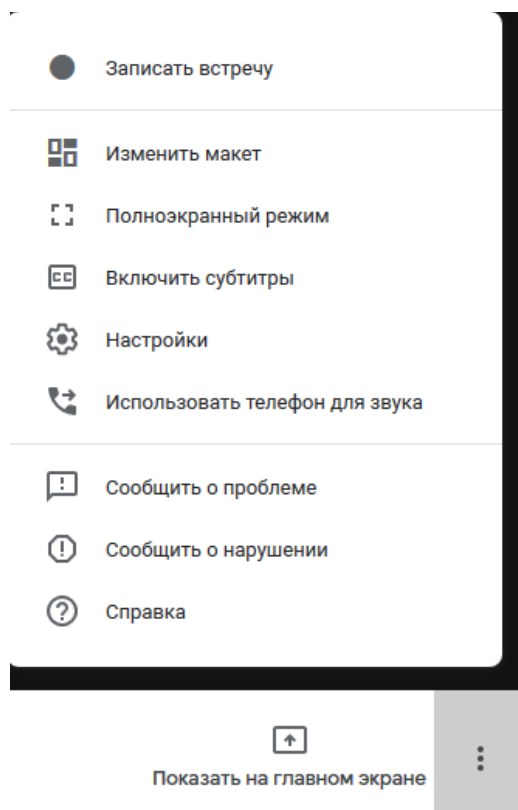
Лабораторная работа 10

1. Войдите в свой корпоративный аккаунт google на домене oiate.
2. Перейдите во вкладку создания онлайн конференций по ссылке <https://meet.google.com/>
3. Нажмите «Присоединиться к встрече или создать свою».



4. Придумайте название встречи и нажмите «Продолжить».
5. Разрешите использование камеры и микрофона, согласившись со всеми всплывающими уведомлениями.
6. Нажмите «Присоединиться».

7. Скопируйте ссылку для подключения и разошлите ее двоим одноклассникам.
8. Дождитесь пока они зайдут к вам на встречу.
9. Запустите запись видеовстречи нажав на три точки в правом нижнем углу и выбрав «Записать встречу».



10. Откройте на вашем компьютере презентацию, выполненную в рамках лабораторной работы 5.
11. Начните показ ваших материалов используя меню «Показать на главном экране»-«Весь ваш экран».
12. Откройте презентацию и пролистайте ее до конца, удостоверьтесь, что ваши слушатели ее видят.
13. Закройте презентацию.
14. Остановите запись встречи. После остановки запись будет помещена на ваш гугл диск в папку Meet Recordings. Вам необходимо будет найти ее и прикрепить к заданию в классруме вместе с отчетом по лабораторной работе в формате .docx со скриншотами, пояснениями и ответами на вопросы.

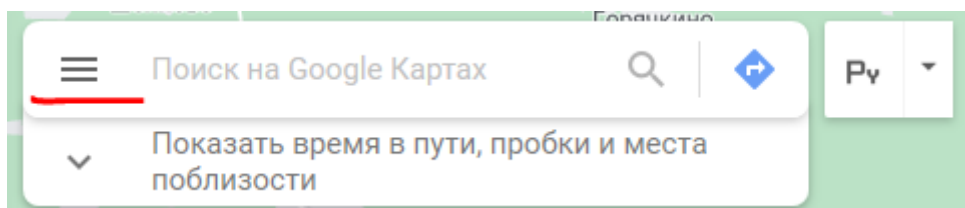
Контрольные вопросы

1. Что такое и для чего используется Google Meet?
2. Опишите основные преимущества и особенности Google Meet?

Лабораторная работа 11

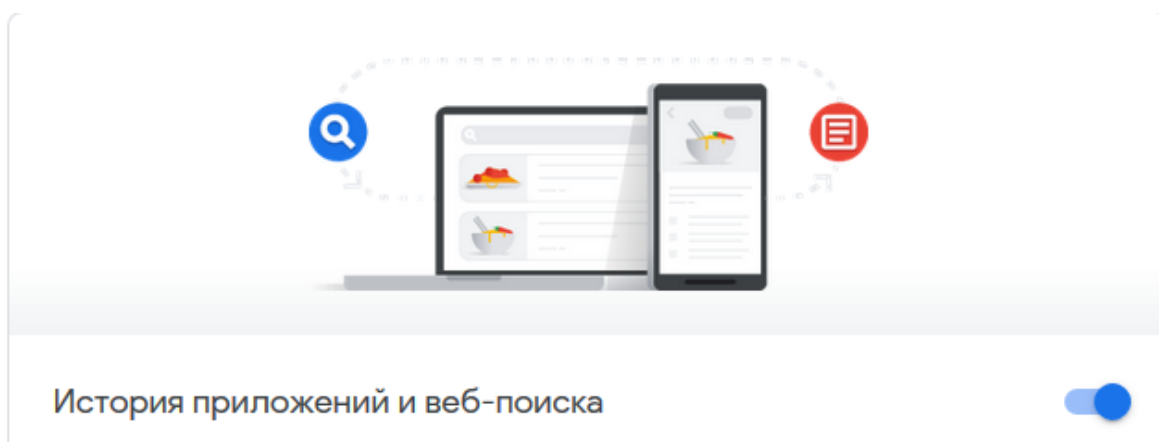
1. Войдите в свой корпоративный аккаунт google на домене oiate.
2. Перейдите во вкладку с онлайн-картами [<https://www.google.ru/maps/>]

3. Откройте полное меню, используя символ ☰ в левом верхнем углу, как показано на рисунке ниже.

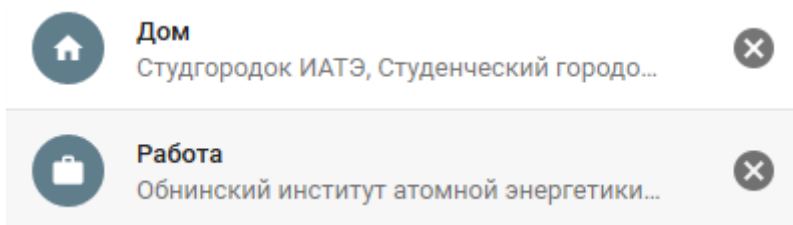


4. Выберите вкладку «Мои места» и нажмите на нее левой кнопкой мыши.

5. Добавьте домашний адрес. Обязательно необходимо открыть настройки аккаунта и включить «Историю приложений и веб-поиска» для добавления адреса.

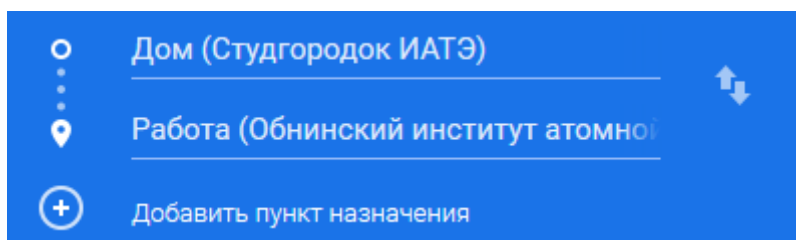


6. Добавьте рабочий (учебный) адрес.



7. Выберите домашний адрес и нажмите на него левой кнопкой мыши.

8. После этого нажмите на «Проложить маршрут» и проложите ваш маршрут к вашей работе (учебе), например, как на скриншоте ниже.



9. Отправьте ваш маршрут на телефон в письме на почту с помощью одноименной функции «Отправить маршрут на телефон».

10. Откройте маршрут с телефона, сделайте скриншот и вставьте его в ваш отчет.

11. Оформите отчет в формате .docx со скриншотами и описаниями всех действий и сдайте вашему преподавателю в классруме.

Контрольные вопросы

1. Что такое и для чего используется Google Maps?
2. Опишите основные преимущества и возможности Google Maps?

Лабораторная работа 12

1. Войдите в свой корпоративный аккаунт google на домене oiate.
2. Запустите онлайн сервис Google Colaboratory с помощью ссылки <https://colab.research.google.com>
3. Создайте новый блокнот (Jupyter Notebook) нажав на надпись «Создать блокнот».
4. Поменяйте название блокнота. Вместо «Untitled» напишите «Лабораторная работа 12 ФИО студента».
5. Решите следующие задачи с помощью языка программирования Python. С помощью текстовых полей на языке Markdown пишите условие и номер задачи, а затем решение в строке для написания кода.

- Напишите программу, которая считывает четыре числа и выводит их произведение. Каждое число записано в отдельной строке.
- Даны два целых числа. Выведите значение наибольшего из них используя оператор if.
- Даны три целых числа. Выведите значение наименьшего из них.
- Даны четыре целых чисел. Определите, сколько среди них совпадающих. Программа должна вывести одно из чисел: 4 (если все совпадают), 3 (если три совпадают) 2 (если два совпадает) или 0 (если все числа различны).
- Дано одно натуральное число. Выведите на экран число десятков в его десятичной записи.
- Дано положительное действительное число X. Выведите его первую цифру после десятичной точки.
- Дано трехзначное число. Найдите сумму его цифр.
- Даны два целых числа A и B (при этом $A \leq B$). Выведите все числа от A до B включительно.
- Пицца в магазине стоит с рублей и d копеек. Нужно определить, сколько рублей и копеек нужно заплатить за n пицц. Программа получает на вход три числа: c, d, n, и должна вывести два числа: стоимость всей покупки в рублях и копейках.
- По данному натуральном n вычислите сумму факториалов $1!+2!+3!+\dots+n!$. В решении этой задачи можно использовать только один цикл.

- Есть список $c = [3, 4, 2, 10, 3, 1, 5, 8, 13, 21, 34, 55, 96]$. Выведите все элементы, которые меньше 10.
- Даны списки:
 $a = [5, 6, 2, 1, 5, 2, 13, 21, 34, 55, 36]$;
 $b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 36, 11, 12, 13]$.
 Нужно вернуть список, который состоит из элементов, общих для этих двух списков.
- Отсортируйте словарь $d = \{1: 2, 3: 4, 4: 3, 2: 1, 0: 0\}$ по значению в порядке возрастания и убывания.
- Написать функцию факториал(), которая принимает один параметр — число, и возвращает значение — факториал этого числа.
- Написать функцию max_list(), которая принимает список [1,10,55,104,123,568] и выводит 3 самых максимальных элемента в одну строку.
- Дана последовательность чисел в виде списка [10 10 4 2 5 5 10 4 1 10]. Для каждого числа выведите слово ДА (в отдельной строке), если это число ранее встречалось в последовательности или НЕТ, если не встречалось.
- Написать функцию, которая выводит периметр, площадь и диагональ квадрата, после того как пользователь вводит сторону.
- Вычислить гипотенузу прямоугольного треугольника. Катеты вводит пользователь.
- Дан список [10, 20, 54, 2, 32, 89, 90]. Поменяйте местами минимальный и максимальный элемент этого списка.

6. Оформите отчет в формате .ipynb с решениями всех задач и ответами на контрольные вопросы, и сдайте вашему преподавателю в классруме.

Контрольные вопросы

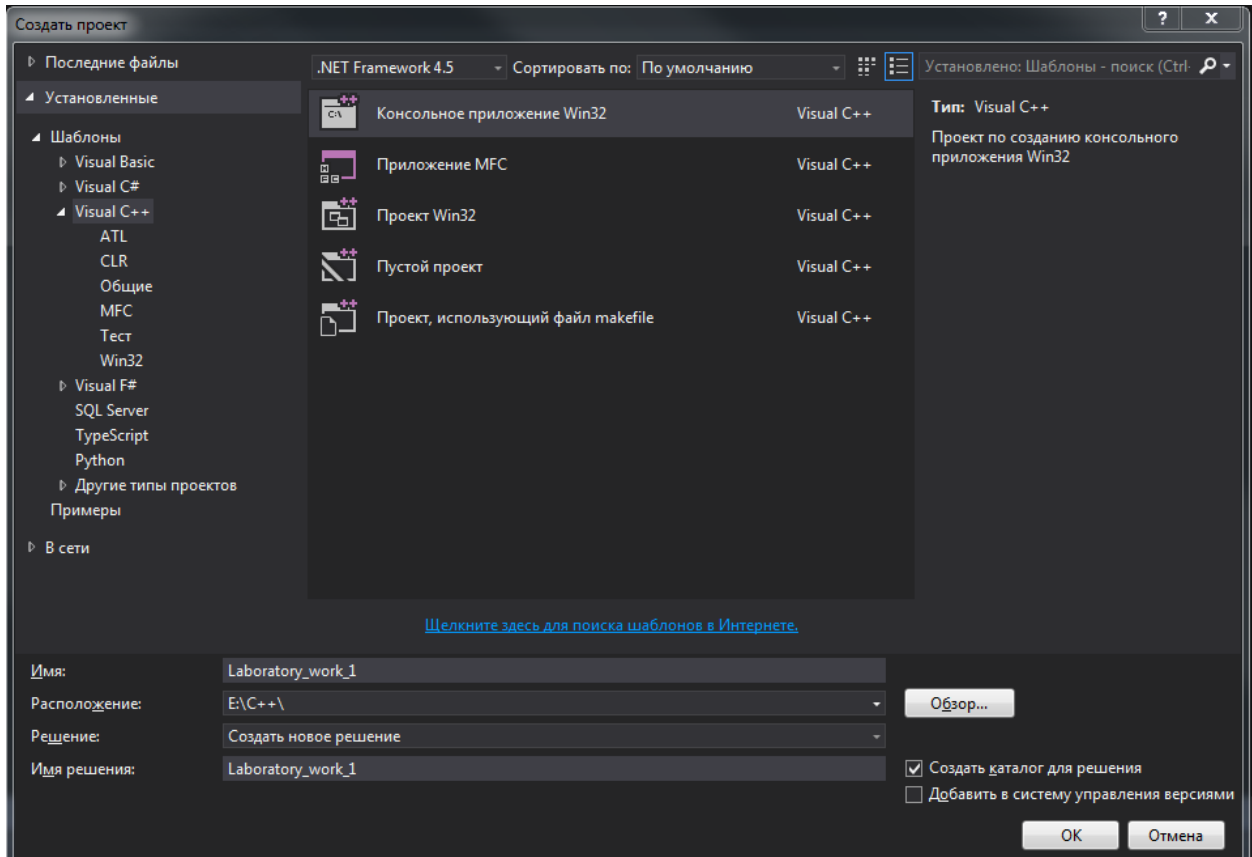
1. Что такое и для чего используется Google Maps?
2. Опишите основные преимущества и возможности Google Maps?

Второй семестр

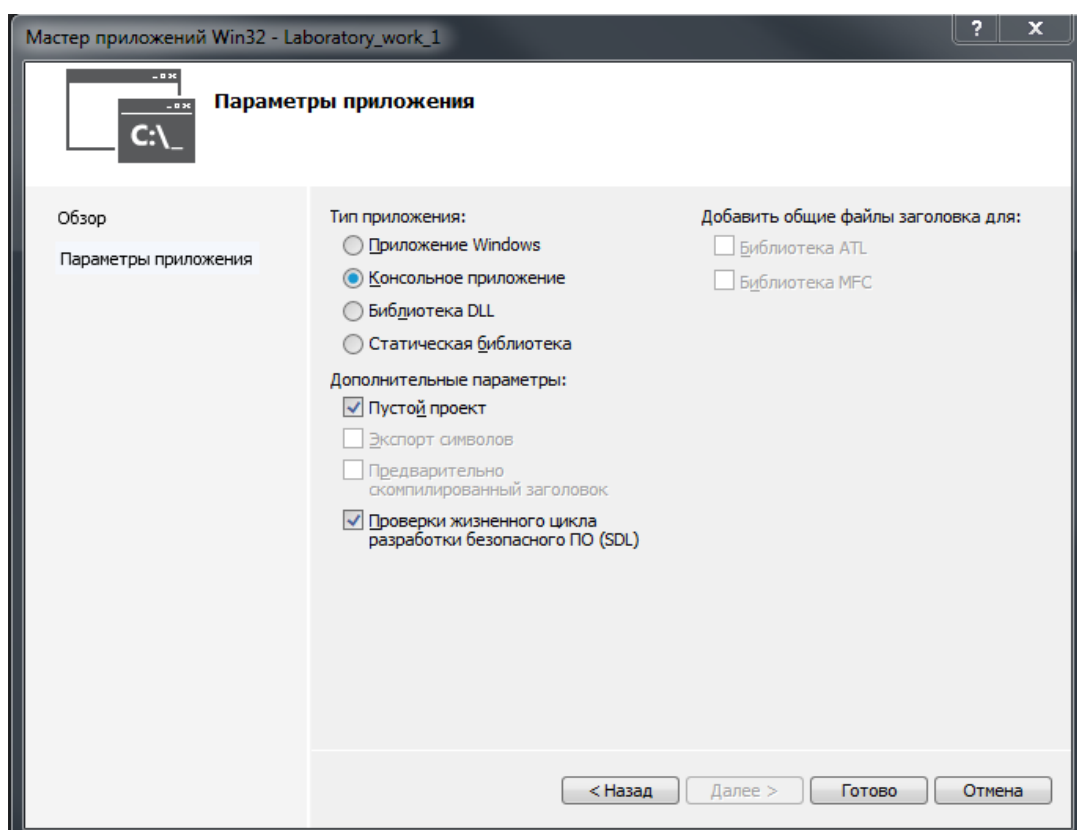
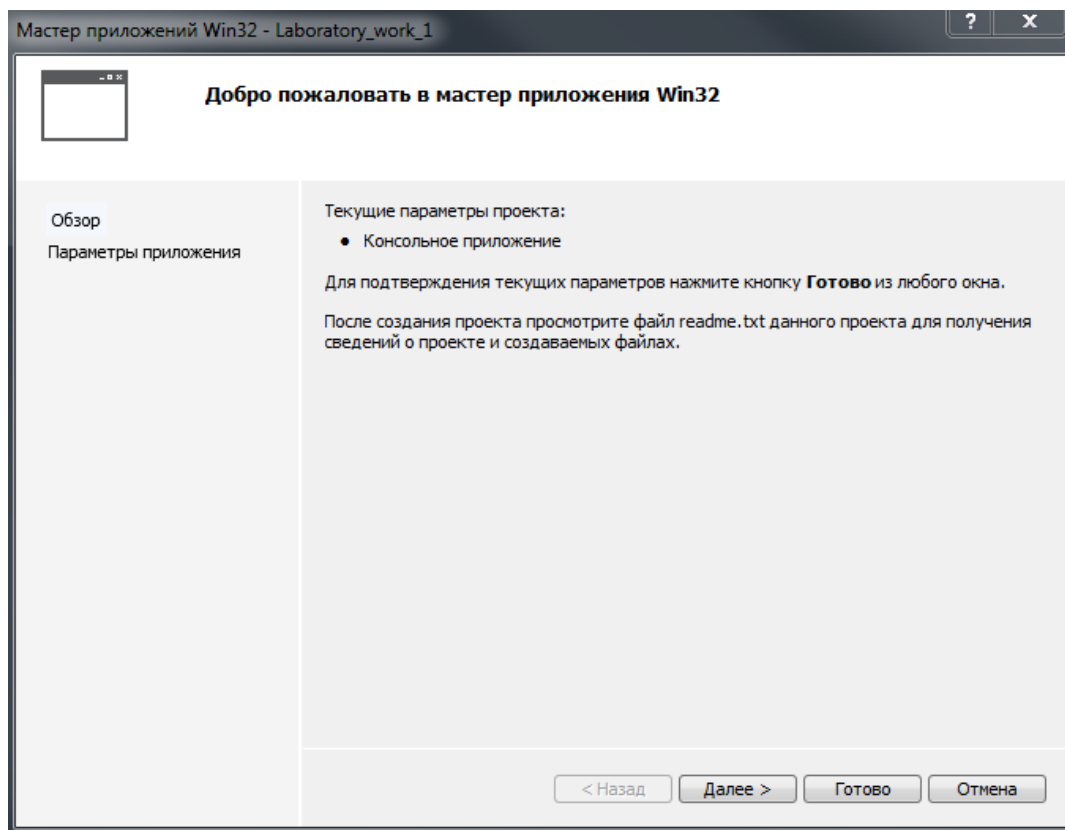
Лабораторная работа 1

1. Запустите интегрированную среду разработки Microsoft Visual Studio.
2. В открывшемся окне на начальной странице создайте новый проект с помощью меню «ФАЙЛ»-«Создать»-«Проект».
3. В открывшемся меню выберите «Visual C++».
4. Затем выберите тип шаблона «Консольное приложение Win32».

5. Введите название проекта «Laboratory_work_1» в поле «Имя». Введенное название автоматически копируется в поле Имя решения. В поле Расположение укажите путь к каталогу, в котором будет сохранен проект (создайте директорию, как в примере: E:\C++\), и установите флажок возле строки «Создать каталог для решения». Нажмите «ОК».



6. В открывшемся окне «Мастер приложений Win32» нажмите на «Далее». В новом окне выберите «Тип приложения»-«Консольное приложение». Снимите флажок с «Предварительное скомпилированный заголовок» в дополнительных параметрах и установите флажок «Пустой проект».

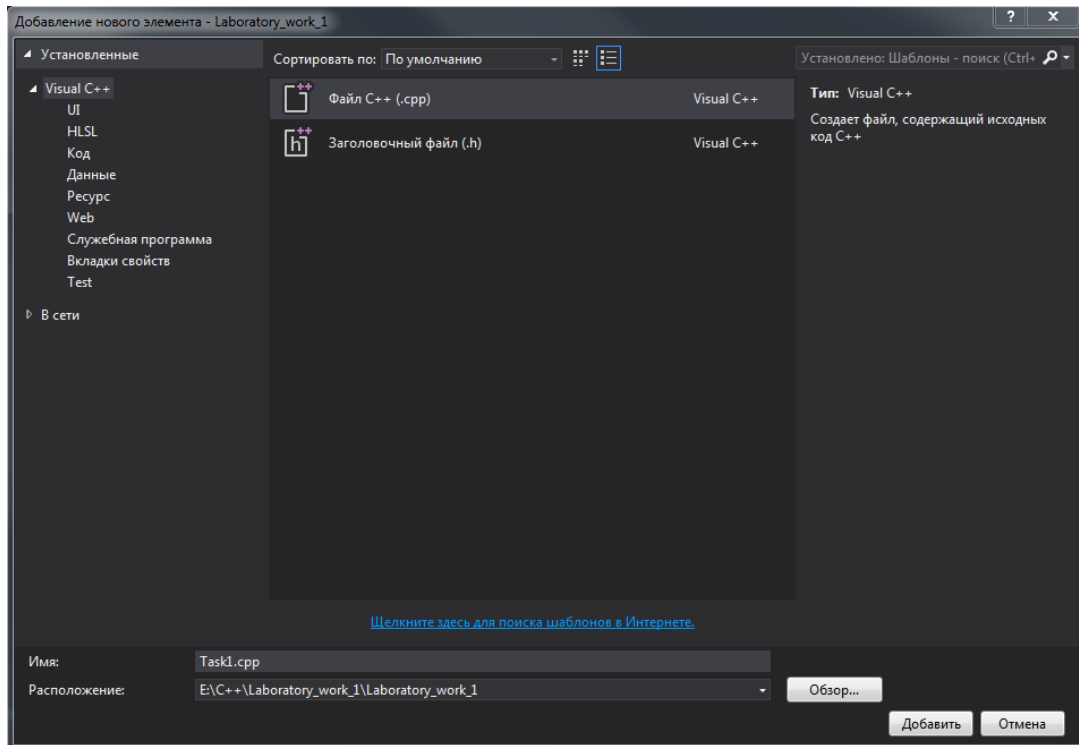


7. Нажмите на кнопку «Готово» и создайте проект.

Примечание: создать пустой проект можно быстрее. Для этого в окне «Создать проект» вместо пункта «Консольное приложение Win32» следует выделить пункт 1. «Пустой проект». В этом случае проект будет создан сразу после нажатия кнопки «ОК».

8. После создания проект добавьте основной файл. Для этого необходимо выбрать, щелкнув правой кнопкой мыши, папку «Файлы исходного кода» в окне «Обозреватель решений» и из контекстного меню выберете пункт «Добавить»-«Создать элемент».

9. В открывшемся окне «Добавление нового элемента» выберите «Файл C++ (.cpp)» и внизу введите имя файла «Task 1.cpp». Затем нажмите кнопку «Добавить».



10. После создания файла, он откроется на отдельной вкладке, добавится в папку проекта и отобразится в окне «Обозреватель решений».

11. Создайте первую программу выводящую надпись “Hello, world!” в окно консоли.

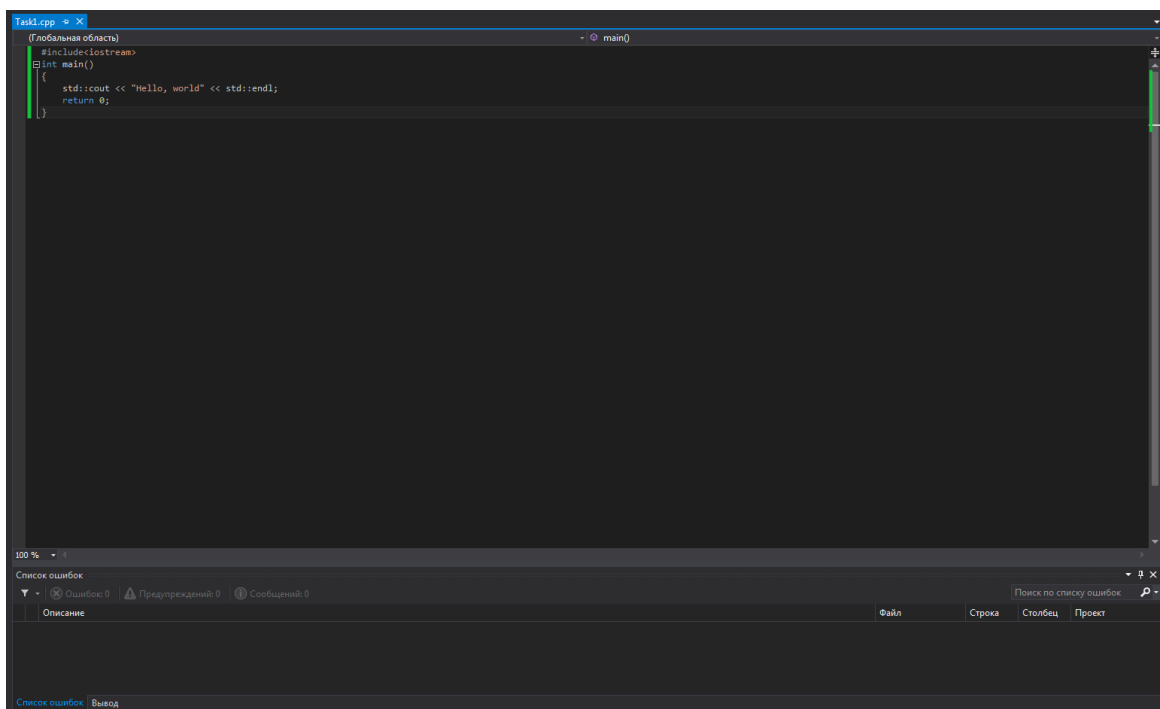
```
Task1.cpp [X]
(Глобальная область)
#include<iostream>
int main()
{
    std::cout << "Hello, world" << std::endl;
    return 0;
}
```

В первой строке программы с помощью директивы include подключается файл iostream, в котором объявлен объект cout, предназначенный для вывода данных в окно консоли. Далее создается функция main(), внутри которой расположены все остальные инструкции, ограниченные фигурными скобками. Именно функция с названием main() будет автоматически вызываться при запуске программы. Перед названием функции указывается тип возвращаемого значения. Ключевое слово int означает, что функция возвращает целое число. Возвращаемое

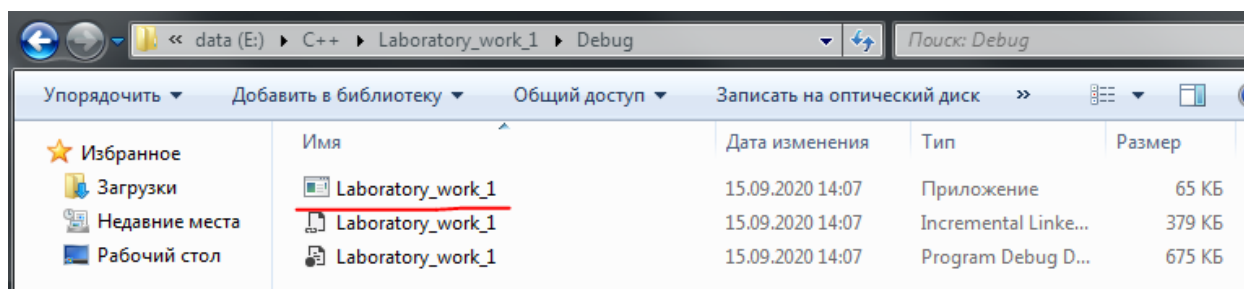
значение указывается после ключевого слова `return` в самом конце функции `main()`. Число `0` в данном случае означает нормальное завершение программы. Если указано другое число, то это свидетельствует о некорректном завершении программы. Согласно стандарту, внутри функции `main()` ключевое слово `return` можно не указывать. В этом случае компилятор должен самостоятельно вставить инструкцию, возвращающую значение `0`.

Вывод строки "Hello, world!" осуществляется с помощью объекта `cout`. Чтобы исключить конфликт имен все стандартные идентификаторы в языке C++ определены в пространстве имен `std`. Поэтому перед именем объекта `cout` необходимо указать, название пространства имен. Между названием пространства имен и названием объекта указываются два двоеточия (`std::cout`). И, наконец, константа `endl` (сокращение от "endline") вставляет символ перевода строки.

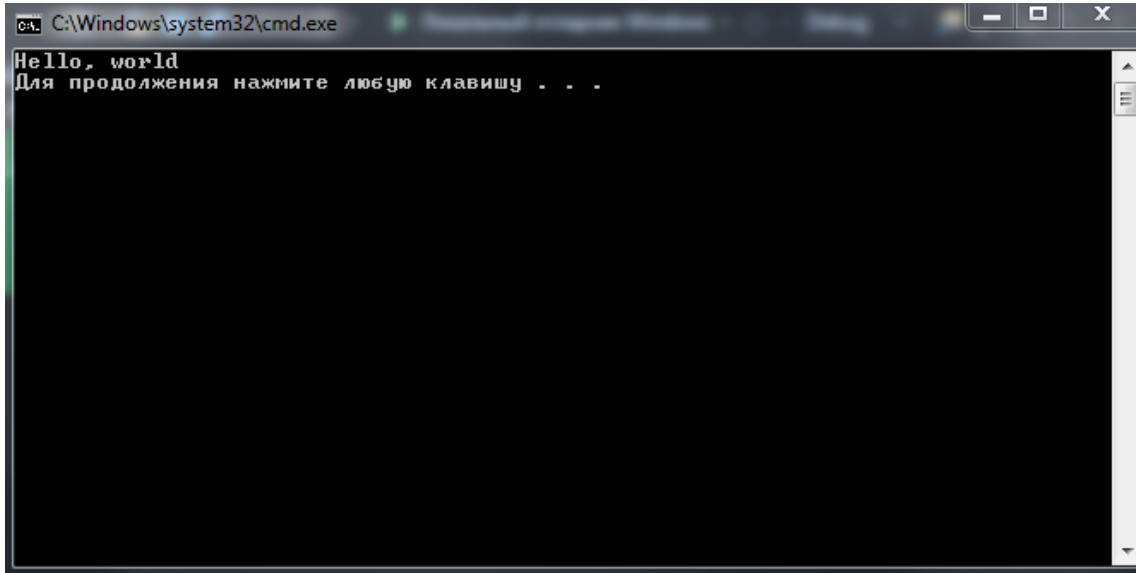
12. Скомпилируйте и запустите программу открыв вкладку «ОТЛАДКА» в меню сверху главного окна и выбрав «Начать отладку» или просто нажав на клавишу `F5`. Запустится Локальный отладчик Windows и если все сделано верно, то вы получите снизу в консоли результат об успешном выполнении вашей программы. Откройте вкладку «Список ошибок» и проверьте, что они отсутствуют.



13. Затем перейдите в папку с вашим проектом на жестком диске ПК и найдите директорию «Debug». Если все было сделано верно, то внутри папки Debug вы увидите созданный `.exe` файл «Laboratory_work_1.exe».



14. Вернитесь обратно в среду разработки Microsoft Visual Studio и запустите программу без отладки. Для этого перейдите во вкладку «ОТЛАДКА» и выберите пункт «Запуск без отладки», либо используйте комбинацию «Ctrl+F5». Запустится программа в консоли. Если все сделано верно, то результат будет как на следующем скриншоте. Нажатие любой клавиши приведет к закрытию окна консоли.

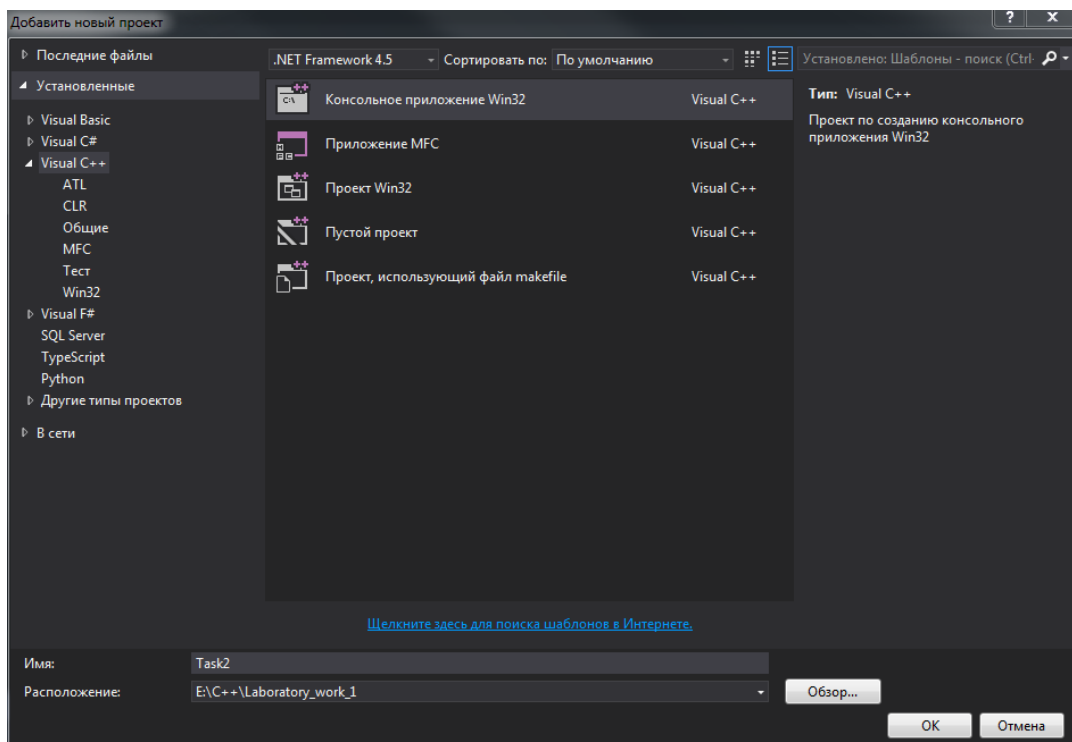


```
C:\Windows\system32\cmd.exe
Hello, world
Для продолжения нажмите любую клавишу . . .
```

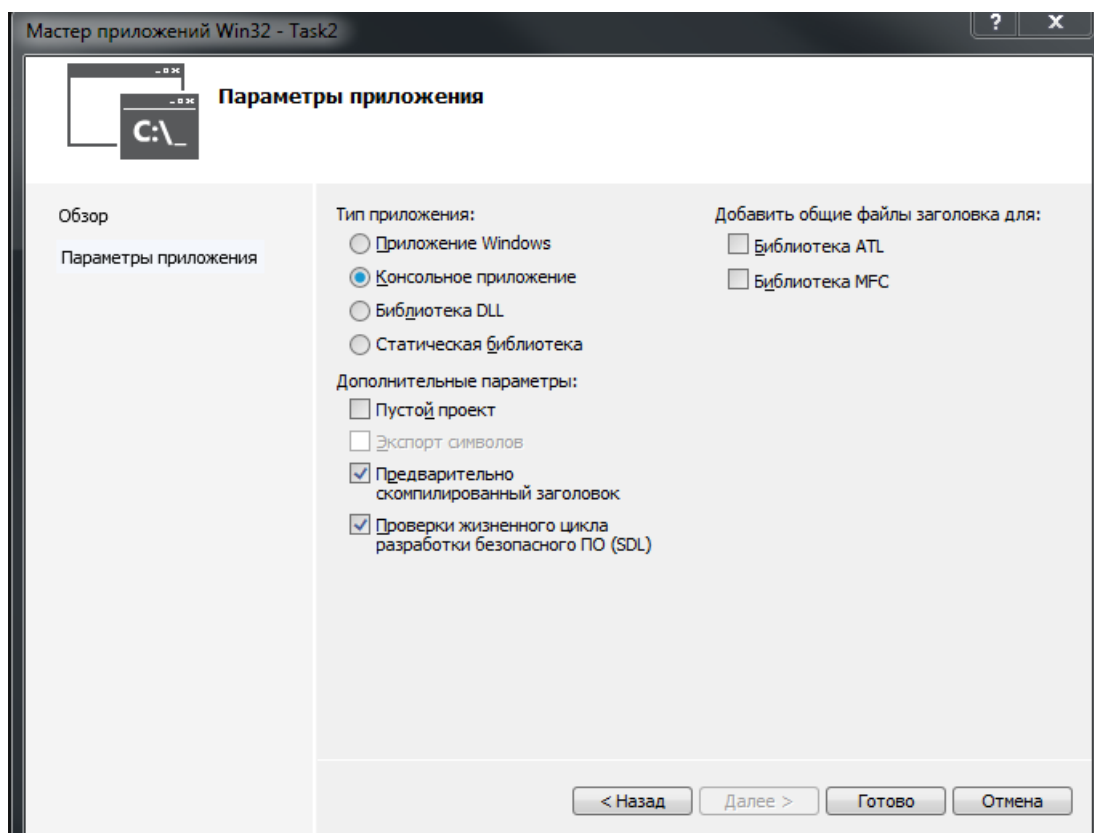
Примечание: Запустить программу можно также в режиме отладки. Для этого из меню «Отладка» выберите пункт «Начать отладку» или нажмите клавишу «F5». В этом случае в консоли откроется и окно и сразу же закроется, а в окне «Вывод» отобразится отладочная информация.

15. Создайте консольное приложение Win32 в вашем проекте (решении) «Laboratory_work_1». Для этого щелкните правой кнопкой мыши по решению «Laboratory_work_1» в Обозревателе решений и выберите команды «Добавить»-«Создать проект».

16. В открывшемся окне выделите пункт «Консольное приложение Win32». В поле Имя введите название проекта «Task2» и нажмите на кнопку ОК.



17. В открывшемся окне «Мастер приложений Win32». Нажмите на кнопку Далее. В группе переключателей «Тип приложения» выберите «Консольное приложение». Установите флажок «Предварительно скомпилированный заголовок» и нажмите на кнопку «Готово».



18. В результате будет создан новый проект, файлы которого можно увидеть в Обозревателе решений. Рассмотрим, для чего предназначены файлы данного консольного приложения.

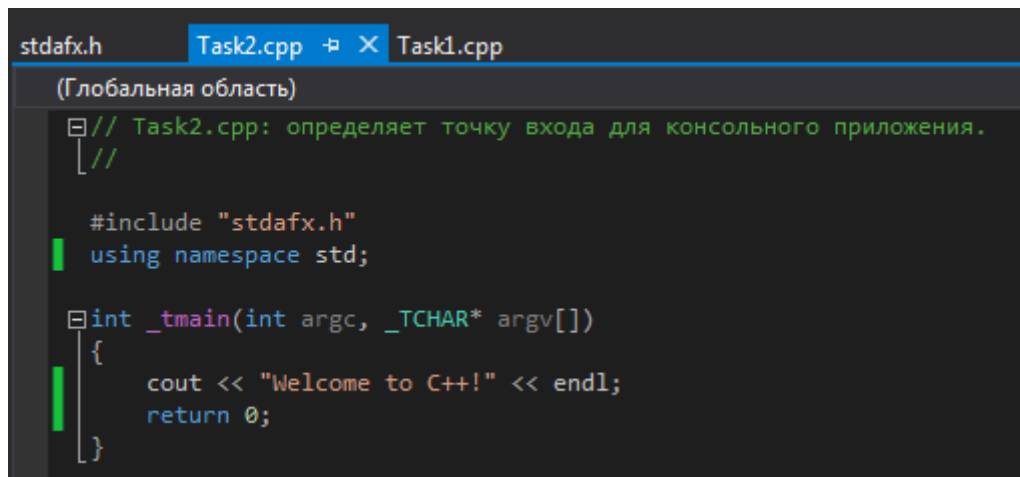
- Задание 2.cpp – основной файл проекта;

- `stdafx.h` – файл для подключения заголовочных файлов, используемых в проекте. Этот файл необходимо подключать во всех файлах проекта,
- `stdafx.cpp` – файлы `stdafx.h` и `stdafx.cpp` используются для построения файла предкомпилированного заголовка (`Task2.pch`) и файла предкомпилированных типов (`stdafx.obj`)
- `targetver.h` – в этом файле подключается файл `SDKDDKVer.h`, который обеспечивает определение последней доступной платформы Windows.

Файлы `stdafx.cpp` и `targetver.h` изменять не нужно. Необходимо в конце файла `stdafx.h` вставить строку, позволяющую использовать в программе объект `cout`, предназначенный для вывода данных в окно консоли: `#include<iostream>`

Директива `pragma once` указывает, что содержимое файла может быть включено только один раз. Директивы `include` подключают файлы, в которых содержатся объявления идентификаторов. Такие файлы называются заголовочными файлами. Они содержат только объявление идентификаторов без описания их реализации. Обратите внимание на то, что в директиве `include` заголовочные файлы подключаются разными способами. Поиск файлов, указанных внутри угловых скобок, производится в системных каталогах, а поиск файлов, указанных внутри кавычек, производится в каталоге проекта. Таким образом, названия файлов стандартной библиотеки необходимо указывать внутри угловых скобок, а пользовательских файлов внутри кавычек. Три первых заголовочных файла содержат расширение `h`, а в последней директиве расширение файла не указано. Наличие расширения файла принято в стандартной библиотеке языка `C`. В стандартной библиотеке языка `C++` расширение файла принято не указывать. Так как язык `C++` наследует все библиотеки языка `C`, то файлы можно подключать как в стиле языка `C`, так и в стиле языка `C++`. Например, файл `string.h` из стандартной библиотеки языка `C` доступен в языке `C++` под названием `string`, а файл `math.h` под названием `cmath`. Отличие между этими способами подключения заключается в импортировании идентификаторов. В языке `C` при подключении файла (например, `math.h`) все идентификаторы импортируются в глобальное пространство имен, а в языке `C++` при подключении файла (например, `math`) идентификаторы определяются в пространстве имен под названием `std`. Поэтому перед идентификатором необходимо указать название пространства имен (например, `std::cout`). Использование пространств имен позволяет избежать конфликта имен в программе. Чтобы посмотреть содержимое заголовочного файла (например, `stdio.h`) щелкните правой кнопкой мыши на строке `"#include<stdio.h>"` и из контекстного меню выберите пункт Открыть документ `<stdio.h>`. В результате заголовочный файл `stdio.h` будет открыт на отдельной вкладке. Следует заметить, что файл `stdio.h` содержит объявления идентификаторов, предназначенных для операций ввода/вывода в языке `C`. Так как для вывода мы используем объект `cout`, объявленный в файле `iostream`, то подключение файла `stdio.h` можно вообще удалить из файла `stdafx.h`. Производить операции ввода/вывода мы будем только в стиле языка `C++`

19. Откройте файл `Task2.cpp` из Обозревателя решений.
20. Добавьте в этот файл следующие строки

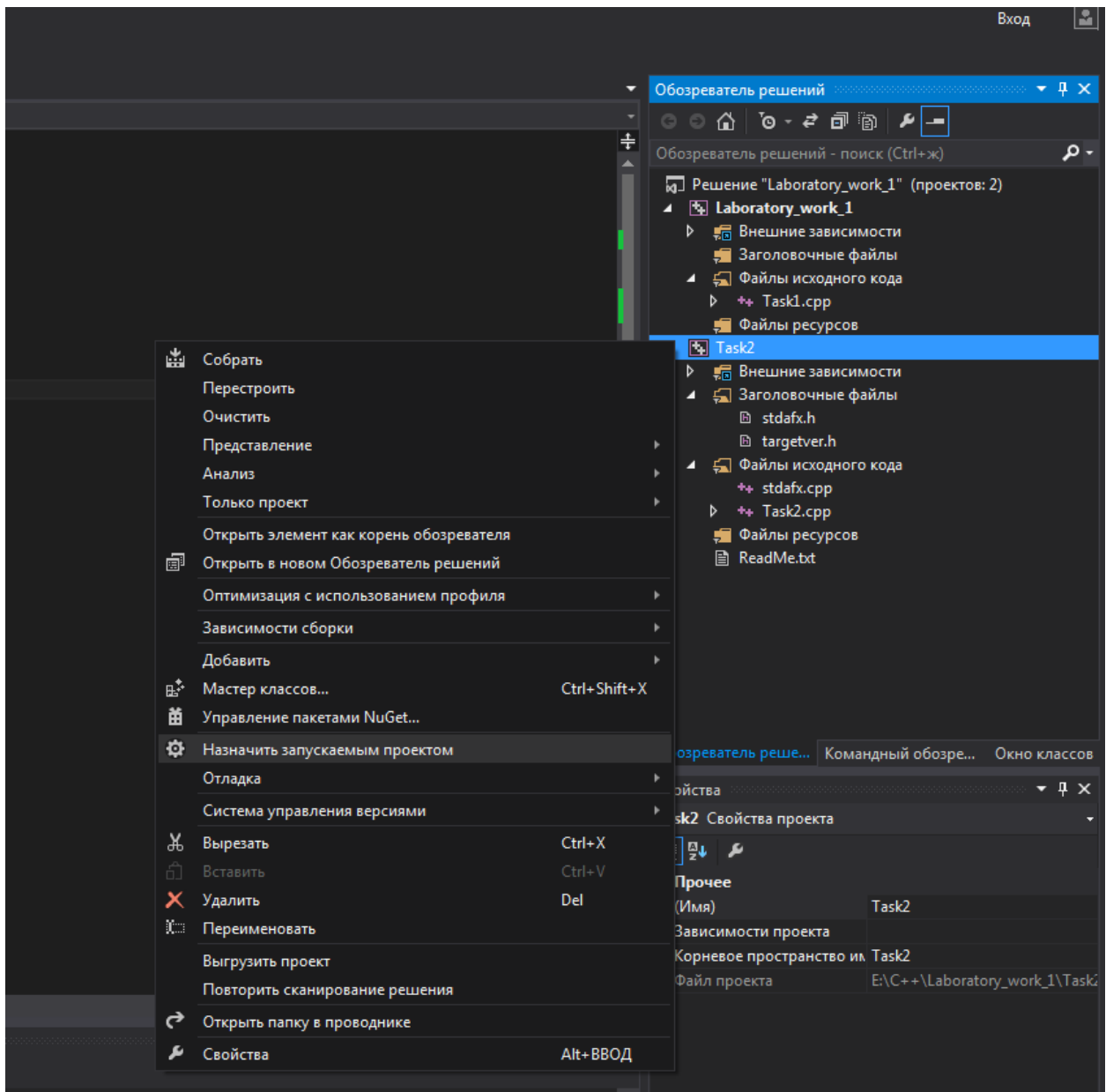


```
stdafx.h Task2.cpp Task1.cpp
(Глобальная область)
// Task2.cpp: определяет точку входа для консольного приложения.
//
#include "stdafx.h"
using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    cout << "Welcome to C++!" << endl;
    return 0;
}
```

21. Запустите созданный скрипт сначала в режиме отладки и проверьте содержимое папки Debug и наличие ошибок в консоли, сделав соответствующие скриншоты для отчета.

Примечание: при запуске будет выполняться предыдущий проект. Для того, чтобы в рамках одного решения запускать разные проекты надо выполнить определенные действия. Перед запуском нового проекта надо в **Обзревателе решений** щелкнуть правой кнопкой по названию нового проекта и выбрать команду «Назначить запускаемым проектом».



22. Запустите созданный скрипт **без отладки в консоли**, сделайте скриншот и вставьте в отчет.

23. Подготовьте отчет о проделанной работе в формате .docx с пояснениями и скриншотами выполнения лабораторной работы. Подготовьте проект со всеми созданными файлами, заархивируйте его и отправьте преподавателю вместе с отчетом.

Контрольные вопросы

1. Что такое Microsoft Visual Studio?
2. Что такое Решение и как оно связано с проектом.
3. Что такое пустой проект.
4. Чем отличается создание пустого проекта от консольного приложения.

Лабораторная работа 2

1. Запустите интегрированную среду разработки Microsoft Visual Studio.
2. Создайте новый проект «Консольное приложение Win32» с помощью шагов из первой лабораторной работы и назовите «Laboratory_work_2».
3. Решите следующие задачи. Для каждой задачи создавайте новый файл .cpp в «Файлах исходного кода» (например, для Задачи 1 – «Task1», для Задачи 2 – «Task2.cpp» и т.д.). Сохраняйте ваши решения, делайте скриншоты и вставляйте в отчет, прикладывая файлы .cpp к нему в классруме.

Задачи:

1. N студентов взяли в библиотеке K книжек и решили разделить их поровну. Определите, сколько книжек достанется каждому студенту. На вход в программу подается два целых положительных числа N и K, каждое из которых не превышает 10000. Выведите одно целое число. Для проверки работоспособности программы используйте N=7, K = 26.
2. N студентов взяли K книг в библиотеке и решили разделить их поровну. Определите, сколько книг останется после того, как все студенты возьмут себе равное количество книг. На вход в программу подается два целых положительных числа N и K, каждое из которых не превышает 10000. Выведите одно целое число. Для проверки работоспособности программы используйте N=5, K = 18.
3. Дано натуральное число, выведите его последнюю цифру.
4. Дано трехзначное число. Выведите его первую цифру (число десятков). Проверьте правильность работы программы с помощью числа 584.
5. Дано целое неотрицательное число N, определите число десятков в нем (предпоследнюю цифру числа). Если предпоследней цифры нет, то можно считать, что число десятков равно нулю. На вход в программу подается целое положительное число N ($0 \leq N \leq 1000000$). Проверьте правильность работы программы с помощью числа 35.
6. Дано целое трехзначное число. Найдите сумму его цифр. Проверьте правильность работы программы с помощью числа: 346
7. Программа получает на вход целое натуральное число. Выведите следующее за ним четное целое число. Проверьте работоспособность программы с произвольными натуральными числами.
8. Товар в магазине стоит A рублей и B копеек. Определите, сколько рублей и копеек нужно заплатить за N товаров в магазине. Программа должна получать на вход три числа: A, B, N - целые, положительные. Программа должна вывести два числа через пробел: стоимость покупки в рублях и копейках. Проверьте работоспособность программы при A = 10, B = 50, N = 7.
9. Улитка ползет по вертикальному шесту высотой H метров, поднимаясь за день на A метров, а за ночь спускаясь на B метров. На какой день улитка доползет до вершины шеста? Программа получает на вход целые неотрицательные числа H, A, B, причем $H > B$. Числа не превосходят 100.

Программа должна вывести одно натуральное число. Гарантируется, что $A > B$.

Контрольные вопросы

1. Для чего надо подключать директиву `iostream`?
2. Какой оператор выводит данные на экран?
3. Для чего предназначен `endl`?

Лабораторная работа 3

1. Запустите интегрированную среду разработки Microsoft Visual Studio.
2. Создайте новый проект «Консольное приложение Win32» с помощью шагов из первой лабораторной работы и назовите «Laboratory_work_3».
3. Решите следующие задачи. Для каждой задачи создавайте новый файл `.cpp` в «Файлах исходного кода» (например, для Задачи 1 – «Task1», для Задачи 2 – «Task2.cpp» и т.д.). Сохраняйте ваши решения, делайте скриншоты и вставляйте в отчет, прикладывая файлы `.cpp` к нему в классруме.
4. Даны два целых числа. Выведите значение наименьшего из них. Если числа равны, выведите любое из них. Проверьте работоспособность программы сначала с числами 8 и 9, а затем с числами 5 и 5.
5. Даны четыре целых числа. Выведите значение наибольшего из них. Если числа равны, выведите любое из них. Проверьте работоспособность программы сначала с числами 10, 25, 15, 8, а затем с числами 4,4,4,4.
6. Даны два целых числа. Программа должна вывести единицу, если первое число больше второго, двойку, если второе больше первого, или ноль, если они равны. Проверьте работоспособность программы несколько раз с разными числами: 30 и 15, 55 и 78, 60 и 60.
7. Даны четыре целых числа. Определите, сколько среди них совпадающих. Программа должна вывести одно из чисел: 4 (если все совпадают), 3 (если три совпадают), 2 (если два совпадают) или 0 (если все числа различны). Проверьте работоспособность программы несколько раз с различными наборами чисел: 1,2,3,4; 1,1,1,1; 2,2,3,4; 3,3,3,7
8. Дано три числа 5,4,2, упорядочите их в порядке возрастания с помощью условного оператора.
9. Напишите программу, которая будет выдавать пользователю является ли год високосным. Год является високосным если он кратен 4 и не кратен 100 или кратен 400. Выведите «Високосный» если год является таковым и «Не високосный», если не является.

Контрольные вопросы

1. Для чего надо подключать директиву `iostream`?
2. Какой оператор выводит данные на экран?
3. Для чего предназначен `endl`?

Лабораторная работа 4

1. Запустите интегрированную среду разработки Microsoft Visual Studio.
2. Создайте новый проект «Консольное приложение Win32» с помощью шагов из первой лабораторной работы и назовите «Laboratory_work_4».
3. Решите следующие задачи. Для каждой задачи создавайте новый файл .cpp в «Файлах исходного кода» (например, для Задачи 1 – «Task1», для Задачи 2 – «Task2.cpp» и т.д.). Сохраняйте ваши решения, делайте скриншоты и вставляйте в отчет, прикладывая файлы .cpp к нему в классруме.
4. Напишите программу, которая получает на вход натуральное целое число N и выводит в качестве выходных данных квадраты всех чисел, не превышающих введенное число в порядке возрастания. Протестируйте программу на нескольких вариантах.
5. Напишите программу, которая получает целое положительное число не менее 2 и выводит в качестве выходных данных его наименьший натуральный делитель, но не число 1. Протестируйте программу на нескольких вариантах.
6. Напишите программу, которая получает на вход натуральное число N и выводит в качестве выходных данных слово ДА, если число N является степенью двойки или слово НЕТ в противном случае. Протестируйте программу на нескольких вариантах.
7. Напишите программу, которая получает на вход последовательность целых неотрицательных чисел, с условием, что каждое число записано в отдельной строке. Последовательность должна быть закончена числом 0, во время считывания которого программа должна завершить свою работу и вывести количество элементов последовательности (не считая последнего числа 0). Числа после 0 не учитываются. Протестируйте программу на нескольких вариантах, в том числе и с числами после 0.
8. Напишите программу, которая получает на вход последовательность целых чисел, оканчивающейся числом 0 и выводит в качестве выходных данных сумму всех элементов последовательности, до числа 0. Протестируйте программу на нескольких вариантах, в том числе и с числами после 0.
9. Напишите программу, которая получает на вход последовательность целых чисел, оканчивающейся числом 0 и выводит в качестве выходных данных значение наибольшего элемента последовательности, до числа 0. Протестируйте программу на нескольких вариантах, в том числе и с числами после 0.
10. Напишите программу, которая получает на вход последовательность целых чисел, оканчивающейся числом 0 и выводит в качестве выходных данных значение второго по величине элемента в этой последовательности до числа 0. Протестируйте программу на нескольких вариантах, в том числе и с числами после 0.
11. Напишите программу, которая получает на вход натуральное число N и выводит в качестве выходных данных N -е число Фибоначчи $F(N)$. Формула последовательности Фибоначчи: $F(0) = 0$, $F(1) = 1$, ..., $F(n) = F(n-1) + F(n-2)$. Протестируйте работоспособность программы на нескольких вариантах чисел.

Контрольные вопросы

1. Что такое и для чего используется цикл `while`?
2. Какие существуют альтернативы циклу `while`?

Лабораторная работа 5

1. Запустите интегрированную среду разработки Microsoft Visual Studio.
2. Создайте новый проект «Консольное приложение Win32» с помощью шагов из первой лабораторной работы и назовите «Laboratory_work_5».
3. Решите следующие задачи. Для каждой задачи создавайте новый файл `.cpp` в «Файлах исходного кода» (например, для Задачи 1 – «Task1», для Задачи 2 – «Task2.cpp» и т.д.). Сохраняйте ваши решения, делайте скриншоты и вставляйте в отчет, прикладывая файлы `.cpp` к нему в классруме.
4. Напишите программу, которая получает на вход положительное действительное число, а в качестве выходных данных выводит его дробную часть. Протестируйте программу на нескольких вариантах действительных чисел.
5. Напишите программу, которая получает на вход положительное действительное число, а в качестве выходных данных выводит первую цифру его дробной части. Протестируйте программу на нескольких вариантах действительных чисел.
6. Напишите программу, которая получает на вход три положительных действительных числа, которые означают длины сторон треугольника. В качестве выходных данных программа должна выводить площадь треугольника. Протестируйте программу на нескольких вариантах.
7. Напишите программу, которая получает на вход три целых положительных числа, которые означают процентную ставку (P), размер вклада в рублях (R) и копейках (C). В качестве выходных данных программа должна выводить размер вклада через год в рублях и копейках, т.е. 2 числа через пробел. Используйте правило: процентная ставка по вкладу прибавляется к сумме вклада в конце года. Протестируйте программу на нескольких вариантах.
8. Напишите программу, которая получает на вход последовательность целых чисел, оканчивающейся числом 0 и выводит в качестве выходных данных среднее арифметическое значение всех элементов последовательности, до числа 0. Протестируйте программу на нескольких вариантах, в том числе и с числами после 0.
9. Напишите программу, которая получает на вход последовательность целых чисел, оканчивающейся числом 0 и выводит в качестве выходных данных стандартное отклонение для данной последовательности, до числа 0. Протестируйте программу на нескольких вариантах, в том числе и с числами после 0. Формула стандартного отклонения:

$$\sigma = \sqrt{\frac{(x_1 - s)^2 + (x_2 - s)^2 + \dots + (x_n - s)^2}{n - 1}}$$

10. Напишите программу, которая получает на вход три действительных числа a, b, c – коэффициенты квадратного уравнения $ax^2 + bx + c = 0$, при этом a не равно 0. В качестве выходных данных программа должна выводить решение квадратного уравнения. Используйте правило: если уравнение имеет два корня, выведите два корня в порядке возрастания, если один корень — выведите одно число, если нет корней — не выводите ничего. Протестируйте программу на нескольких вариантах.

11. Напишите программу, которая получает на вход шесть действительных чисел a, b, c, d, e, f – коэффициенты системы линейных уравнений.

$$\begin{cases} ax + by = e, \\ cx + dy = f. \end{cases}$$

. Известно, что система имеет ровно одно решение. Программа должна выводить в качестве выходных данных два числа x и y , являющиеся решением этой системы. Протестируйте программу на нескольких вариантах.

Контрольные вопросы

1. Что такое действительные числа?
2. Какие существуют еще числа, кроме действительных? Опишите подробно каждую категорию чисел?

Лабораторная работа 6

1. Запустите интегрированную среду разработки Microsoft Visual Studio.
2. Создайте новый проект «Консольное приложение Win32» с помощью шагов из первой лабораторной работы и назовите «Laboratory_work_6».
3. Решите следующие задачи. Для каждой задачи создавайте новый файл .cpp в «Файлах исходного кода» (например, для Задачи 1 – «Task1», для Задачи 2 – «Task2.cpp» и т.д.). Сохраняйте ваши решения, делайте скриншоты и вставляйте в отчет, прикладывая файлы .cpp к нему в классруме.
4. Напишите программу, которая получает на вход 2 строки ввода, в первой строке вводится количество элементов в массиве, а во второй сами элементы массива. В качестве выходных данных программа должна выводить все элементы массива с четными индексами. Протестируйте программу на нескольких вариантах.
5. Напишите программу, которая получает на вход 2 строки ввода, в первой строке вводится количество элементов в массиве, а во второй сами элементы массива. В качестве выходных данных программа должна выводить все четные элементы массива. Протестируйте программу на нескольких вариантах.
6. Напишите программу, которая получает на вход 2 строки ввода, в первой строке вводится количество элементов в массиве, а во второй сами

элементы массива. В качестве выходных данных программа должна выводить количество элементов в введенном массиве. Протестируйте программу на нескольких вариантах.

7. Напишите программу, которая получает на вход 2 строки ввода, в первой строке вводится количество элементов в массиве, а во второй сами элементы массива. В качестве выходных данных программа должна выводить все элементы массива, которые больше предыдущего элемента. Протестируйте программу на нескольких вариантах.

8. Напишите программу, которая получает на вход 2 строки ввода, в первой строке вводится количество элементов в массиве, а во второй сами элементы массива, отличные от 0. В качестве выходных данных программа должна выводить элементы массива по следующим правилам: если в массиве есть два соседних элемента одинакового знака, то на выходе будут только эти элементы; если соседних элементов с одинаковым знаком нет, то программа ничего не выводит; если пар соседей с одним знаком несколько, то выводится только первая пара. Протестируйте программу на нескольких вариантах действительных чисел.

9. Напишите программу, которая получает на вход 2 строки ввода, в первой строке вводится количество элементов в массиве, а во второй сами элементы массива с хотя бы одним положительным элементом. В качестве выходных данных программа должна выводить значение наименьшего из всех положительных элементов в массиве. Протестируйте программу на нескольких вариантах.

10. Напишите программу, которая получает на вход 2 строки ввода, в первой строке вводится количество элементов в массиве, а во второй сами элементы массива с хотя бы одним положительным элементом. В качестве выходных данных программа должна выводить значение наименьшего из всех положительных элементов в массиве. Протестируйте программу на нескольких вариантах.

11. Напишите программу, которая получает на вход 2 строки ввода, в первой строке вводится количество элементов в массиве, а во второй сами элементы массива. В качестве выходных данных программа должна выводить значение наименьшего нечетного элемента, а если в массиве нет нечетных элементов, то на выходе число 0. Протестируйте программу на нескольких вариантах.

12. Напишите программу, которая получает на вход 2 строки ввода, в первой строке вводится количество элементов в массиве, а во второй сами элементы массива, упорядоченный по возрастанию, например (1 2 2 3 3 3 4 4 4 4). В качестве выходных данных программа должна выводить количество уникальных элементов в массиве. Протестируйте программу на нескольких вариантах.

Контрольные вопросы

1. Для чего используется цикл for?
2. Какие существуют еще числа, кроме действительных? Опишите подробно каждую категорию чисел?

Лабораторная работа 7

1. Запустите интегрированную среду разработки Microsoft Visual Studio.
2. Создайте новый проект «Консольное приложение Win32» с помощью шагов из первой лабораторной работы и назовите «Laboratory_work_7».
3. Решите следующие задачи. Для каждой задачи создавайте новый файл .cpp в «Файлах исходного кода» (например, для Задачи 1 – «Task1», для Задачи 2 – «Task2.cpp» и т.д.). Сохраняйте ваши решения, делайте скриншоты и вставляйте в отчет, прикладывая файлы .cpp к нему в классруме.
4. Напишите программу, которая получает на вход в первой строке 2 числа n и m – размеры двумерного массива, а затем n строк по m чисел в каждой. В качестве выходных данных программа должна выводить индексы (n и m) первого вхождения максимального элемента, т.е. индекс максимального элемента в массиве, при первой его встрече. Протестируйте программу на нескольких вариантах.
5. Дано число n , не превышающее 100. Создайте массив размером $n \times n$ и заполните его по следующему правилу. На главной диагонали должны быть записаны числа 0. На двух диагоналях, прилегающих к главной, числа 1. На следующих двух диагоналях числа 2, и т.д. Выведите полученный массив на экран, разделяя элементы массива пробелами.
6. Напишите программу, которая получает на вход в первой строке размеры массива n и m , затем в n строках элементы массива, а затем числа i и j (не превышающие n и m). В качестве выходных данных программа должна поменять в массиве столбцы с номерами i и j и вывести полученный массив на экран. Протестируйте программу на нескольких вариантах.
7. Напишите программу, которая получает на вход в первой строке 2 числа n и m – размеры двумерного массива, а затем n строк по m чисел в каждой. В качестве выходных данных программа должна выводить слов ДА, если массив является симметричным относительно главной диагонали и слово НЕТ в противном случае. Протестируйте программу на нескольких вариантах.
8. Напишите программу, которая получает на вход в первой строке 1 число n - количество строк и столбцов квадратного двумерного массива, затем n строк по n элементов, а затем в отдельной строке число k . В качестве выходных данных программа должна выводить элементы k -ой по счету диагонали ниже главной диагонали (если $k = 1$), второй диагонали (если $k = 2$) и т.д.
9. Напишите программу, которая получает на вход в первой строке 2 числа n и m – размеры двумерного массива, а затем n строк по m чисел в каждой. В качестве выходных данных программа должна выводить индексы (n и m) первого вхождения максимального элемента, т.е. индекс максимального элемента

в массиве, при первой его встрече. Протестируйте программу на нескольких вариантах.

10. Напишите программу, которая получает на вход в первой строке 2 числа n и m – размеры двумерного массива (кинотеатр с n рядов и m мест, не превосходящие 20), а затем n строк по m чисел в каждой, состоящие из 0 и 1 (0 – место свободно, 1 – место занято), а в последней строке ввода содержится значение k – количество билетов на соседние места в одном ряду. В качестве выходных данных программа должна выводить номер ряда, в котором есть k подряд идущих свободных мест. Если таких рядов не имеется, то необходимо вывести номер наименьшего подходящего ряда. Если подходящего ряда нет, на выходе должно быть число 0. Протестируйте программу на нескольких вариантах.

11. Напишите программу, которая получает на входе в первой строке значения n и m – размер двумерного прямоугольного массива, а в следующих n строках записываются по m значений данного массива. В качестве выходных данных программа должна повернуть массив на 90 градусов по часовой стрелке и выдать новый массив $m \times n$ на экран. Протестируйте программу на нескольких вариантах.

Контрольные вопросы

1. Что такое массив?
2. Какие бывают виды массивов?

Лабораторная работа 8

1. Запустите интегрированную среду разработки Microsoft Visual Studio.
2. Создайте новый проект «Консольное приложение Win32» с помощью шагов из первой лабораторной работы и назовите «Laboratory_work_8».
3. Решите следующие задачи. Для каждой задачи создавайте новый файл .cpp в «Файлах исходного кода» (например, для Задачи 1 – «Task1», для Задачи 2 – «Task2.cpp» и т.д.). Сохраняйте ваши решения, делайте скриншоты и вставляйте в отчет, прикладывая файлы .cpp к нему в классруме.
4. Напишите программу в которой будет функция $\text{func1}(a,b)$, вычисляющая минимум двух чисел a и b . На входе программа должна считывать два числа и выполняться функция func1 . Протестируйте программу на нескольких вариантах.
5. Напишите программу в которой будет функция $\text{func2}(a,b,c,d)$, вычисляющая минимум четырех чисел. На входе программа должна считывать четыре числа и выполняться функция func2 . Протестируйте программу на нескольких вариантах.
6. Напишите программу в которой будет функция $\text{func3}(a,b,c,d)$, вычисляющая расстояние между точками (a,b) и (c,d) . На входе программа должна считывать четыре действительных числа и выполняться функция $\text{func3}()$. Протестируйте программу на нескольких вариантах.

7. Напишите программу в которой будет рекурсивная функция `func4(b,n)`, вычисляющая b^n . На входе программа должна считывать действительное положительное число b и целое число n и выполняться функция `func4()` с решением задачи. Протестируйте программу на нескольких вариантах.

8. Напишите программу в которой будет функция `func5(n)`, вычисляющая наименьший делитель натурального числа $n > 1$. На входе программа должна считывать натуральное число n и выполняться функция `func5()` с ответом на задачу. Обработайте условие: Если у числа n нет делителя, меньшего n , то число n — простое и ответом будет само число n . Протестируйте программу на нескольких вариантах.

9. Напишите программу в которой будет функция `func6(n)`, вычисляющая сумму всех чисел последовательности n , оканчивающейся на 0, без использования цикла. На входе программа должна считывать последовательность целых чисел, оканчивающихся числом 0 и выполняться функция `func6()` с ответом на задачу. Протестируйте программу на нескольких вариантах.

10. Напишите программу в которой будет функция `func7(n)`, которая по данному целому положительному n возвращает n -е число Фибоначчи. Вместо цикла в функции должна быть рекурсия. На входе программа должна считывать целое число и выполняться функция `func7()` с ответом на задачу. Первое и второе числа Фибоначчи равны 1, а каждое следующее равно сумме двух предыдущих. Протестируйте программу на нескольких вариантах.

Контрольные вопросы

1. Что такое функция?
2. Что такое рекурсия?

Лабораторная работа 9

1. Запустите интегрированную среду разработки Microsoft Visual Studio.
2. Создайте новый проект «Консольное приложение Win32» с помощью шагов из первой лабораторной работы и назовите «Laboratory_work_9».
3. Решите следующие задачи. Для каждой задачи создавайте новый файл `.cpp` в «Файлах исходного кода» (например, для Задачи 1 – «Task1», для Задачи 2 – «Task2.cpp» и т.д.). Сохраняйте ваши решения, делайте скриншоты и вставляйте в отчет, прикладывая файлы `.cpp` к нему в классруме.
4. Напишите программу, которая будет считывать введённый единственный символ и определять является ли он цифрой. Программа должна выводить ДА, если символ является цифрой и НЕТ в противном случае. Протестируйте программу на нескольких вариантах.
5. Напишите программу, которая будет переводить введённый единственный символ или фразу в верхний регистр. Программа должна работать как на русском, так и на английском языке ввода. Протестируйте программу на нескольких вариантах.

6. Напишите программу, которая будет переводить введенный единственный символ или фразу в обратный регистр, т.е. если вводится заглавная буква, то она переводится в нижний регистр, если строчная – верхний. Программа должна работать как на русском, так и на английском языке ввода. Протестируйте программу на нескольких вариантах.

7. Напишите программу, которая будет считать количество слов в произвольной строке, например The world is change. Программа должна работать как на русском, так и на английском языке ввода и выводить на экран количество слов в строке исключая пробелы. Протестируйте программу на нескольких вариантах. В этой задаче может быть полезен метод find с двумя параметрами. Первый из них - искомая подстрока, второй - позиция, начиная с которой нужно искать первое вхождение.

8. Напишите программу, которая будет определять, является ли строка палиндромом (читается одинаково, как справа-налево, так и слева-направо). На вход программе подается 1 строка без пробелов (слово), например deed, level, nun и т.д., а на выходе на экране высвечивается ДА, если строка является таковой, и НЕТ в противном случае. Программа должна работать как на русском, так и на английском языке ввода. Протестируйте программу на нескольких вариантах.

9. Напишите программу, которая будет определять самое длинное слово в строке и выводить его на экран. На вход программе подается 1 строка, в которой слова отделены пробелом, а на выходе выводится самое длинное слово. Программа должна работать как на русском, так и на английском языке ввода. Протестируйте программу на нескольких вариантах.

10. Напишите программу, которая будет определять корректные записи IP-адресов. На вход программе подается 1 строка из произвольных символов, а на выходе выдается слово ДА, если IP-адрес записан корректно и НЕТ в противном случае. Корректный IP-адрес состоит из четырех чисел, принимающих значения от 0 до 255, разделенных точками, например 127.0.0.0, 192.168.0.1, 255.0.255.255. Протестируйте программу на нескольких вариантах.

Контрольные вопросы

1. Что такое символ?
2. Что такое строка?
3. Какие есть методы работы с символами и строками?

Лабораторная работа 10

1. Запустите интегрированную среду разработки Microsoft Visual Studio.
2. Создайте новый проект «Консольное приложение Win32» с помощью шагов из первой лабораторной работы и назовите «Laboratory_work_10».
3. Решите следующие задачи. Для каждой задачи создавайте новый файл .cpp в «Файлах исходного кода» (например, для Задачи 1 – «Task1», для Задачи 2 –

«Task2.cpp» и т.д.). Сохраняйте ваши решения, делайте скриншоты и вставляйте в отчет, прикладывая файлы .cpp к нему в классруме.

4. Напишите программу, которая будет считывать в первой строке значение N – количество элементов, а во второй строке N чисел в списке через пробел, и выводить количество уникальных чисел в полученном списке. Протестируйте программу на нескольких вариантах.

5. Напишите программу, которая будет считывать в первой строке значение N – количество элементов, а во второй строке N чисел в списке через пробел, и выводить для каждого числа в отдельной строке слово ДА, если это число ранее встречалось в списке и НЕТ в противном случае. Протестируйте программу на нескольких вариантах.

6. Напишите программу, которая будет считывать в первой строке значение N – количество элементов в первом списке, во второй строке N чисел в первом списке через пробел, затем число M – количество элементов во втором списке, а в последней строке M чисел второго списка через пробел, и выводить количество чисел, которые содержатся как в первом, так и во втором списке одновременно. Протестируйте программу на нескольких вариантах.

7. Напишите программу, которая будет считывать в первой строке значение N – количество элементов в первом списке, во второй строке N чисел в первом списке через пробел, затем число M – количество элементов во втором списке, а в последней строке M чисел второго списка через пробел, и выводить все числа, которые содержатся как в первом, так и во втором списке одновременно в порядке их возрастания. Протестируйте программу на нескольких вариантах.

Контрольные вопросы

1. Что такое словари и для чего они используются?
2. Что такое множества и для чего они используются?
3. Какие есть методы работы со словарями и множествами?

Лабораторная работа 11

1. Запустите интегрированную среду разработки Microsoft Visual Studio.
2. Создайте новый проект «Консольное приложение Win32» с помощью шагов из первой лабораторной работы и назовите «Laboratory_work_11».
3. Решите следующие задачи. Для каждой задачи создавайте новый файл .cpp в «Файлах исходного кода» (например, для Задачи 1 – «Task1», для Задачи 2 – «Task2.cpp» и т.д.). Сохраняйте ваши решения, делайте скриншоты и вставляйте в отчет, прикладывая файлы .cpp к нему в классруме.
4. Напишите программу, которая будет считывать в первой строке значение N – количество элементов, а во второй строке N чисел в списке через пробел, и выводить все числа в порядке возрастания. Протестируйте программу на нескольких вариантах.

5. Напишите программу, которая будет считывать в первой строке значение S – размер ноги покупателя обувного магазина, во второй строке N – количество пар обуви в магазине и затем список из N -элементов с размерами каждой пары обуви. Размер не должен превосходить 50, как и количество пар обуви в магазине. На выходе из программы должно выводиться единственное число – максимальное количество пар обуви, которое может надеть покупатель на свои ноги, при условии, что одну пару обуви можно надеть на другую, если она хотя бы на четыре размера больше. Протестируйте программу на нескольких вариантах.

6. Напишите программу, которая будет считывать в первой строке значение N – количество участников олимпиады, а в следующих N строках по 2 числа в каждой – уникальный идентификационный номер участника и количество баллов за олимпиаду (например 91 80, участник под номером 91 набрал 80 баллов). Программа должна выводить исходный список (номера-баллы участников) в порядке убывания баллов участников. Если у некоторых участников одинаковые баллы, то их между собой нужно упорядочить в порядке возрастания идентификационного номера. Протестируйте программу на нескольких вариантах.

7. Напишите программу, которая получает на вход набор точек на плоскости. Сначала задано количество точек n , затем идет последовательность из n строк, каждая из которых содержит два числа: координаты точки. Величина n не превосходит 100, все исходные координаты – целые числа, не превосходящие 10^3 . Программа должна вывести все исходные точки в порядке возрастания их расстояний от начала координат. Протестируйте программу на нескольких вариантах.

8. Напишите программу, которая принимает на вход в первой строке число N – количество учеников ВУЗа, а в следующих N -строках данные об учениках и их оценки: Фамилия, Имя и 3 числа (оценки по математике, информатике и физике). Данные в строке разделены одним пробелом. Оценки принимают значение от 1 до 5. Пример одной строки данных: Ivanov Ivan 5 4 5. Программа должна выводить фамилии и имена учащихся в порядке убывания их среднего балла по трем предметам. Протестируйте программу на нескольких вариантах.

Контрольные вопросы

1. Какие вы знаете структуры данных в C++?
2. Какие есть методы для работы со списками, словарями и множествами?

Лабораторная работа 12

1. Запустите интегрированную среду разработки Microsoft Visual Studio.
2. Создайте новый проект «Консольное приложение Win32» с помощью шагов из первой лабораторной работы и назовите «Laboratory_work_12».

3. Решите следующие задачи. Для каждой задачи создавайте новый файл .cpp в «Файлах исходного кода» (например, для Задачи 1 – «Task1», для Задачи 2 – «Task2.cpp» и т.д.). Сохраняйте ваши решения, делайте скриншоты и вставляйте в отчет, прикладывая файлы .cpp к нему в классруме.

4. Создайте класс Car (машины).
5. Создайте атрибуты класса: name = "c200", make = "mercedez", model = "2008".
6. Создайте два метода (функций) класса start и stop: 1) start (Завести двигатель) 2) stop (Отключить двигатель). Методы должны выводить на экран информацию в скобках (завести или отключить двигатель).
7. Создайте объект класса Car под названием car_a и объект класса Car под названием car_b.
8. Вызовите созданный метод класса (start), используя созданный на прошлом шаге объект класса. То же самое сделайте с методом (stop)
9. Создайте класс Car с одним атрибутом car_count = 0 и одним методом start с несколькими атрибутами (name, make, model), которые будут меняться, согласно экземпляру класса.
10. Создайте объект нового класса (car_a) со значениями атрибутов name, make, model соответственно: "Corrola", "Toyota", 2015.
11. По очереди вызовите метод start и метод car_count от этого объекта класса.
12. Создайте еще один объект класса Car (car_b) со значениями атрибутов name, make, model соответственно: "City", "Honda", 2013.
13. По очереди вызовите метод start и метод car_count от этого объекта класса.

Контрольные вопросы

1. Что такое ООП?
2. Что такое класс в ООП?
3. Что такое методы класса?
4. Что такое функции класса?

Критерии и шкала оценивания лабораторных работ

Оценка	Критерии оценки
Отлично 5 баллов	От 90 до 100% заданий из лабораторной работы сделаны верно, подготовлен отчет и другие необходимые материалы по лабораторной работе.
Хорошо 4 балла	От 75 до 90% и более заданий из лабораторной работы сделаны верно, подготовлен отчет и другие необходимые материалы по лабораторной работе.
Удовлетворительно 3 балла	От 60 до 75% процентов заданий из лабораторной работы сделаны верно, подготовлен отчет и другие необходимые материалы по лабораторной работе.
Неудовлетворительно с 0 до 2 баллов	Сделано верно менее 60% заданий из лабораторной работы, не подготовлен отчет и другие материалы по работе.

