

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

УТВЕРЖДАЮ

Декан факультета



« 03 »



Фионова  
Факультет  
Вычислительной  
техники



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**  
**Б1.В.13. КРОССПЛАТФОРМЕННОЕ ПРОГРАММИРОВАНИЕ**

Направление подготовки — 01.03.02. Прикладная математика и информатика

Направленность(профиль) — Компьютерные технологии

Квалификация выпускника — бакалавр

Форма обучения — очная

Пенза, 2019

## 1. Цели освоения дисциплины

Целью освоения дисциплины «Кроссплатформенное программирование» является формирование у обучающихся формирование систематизированных знаний в области современных технологий разработки программного обеспечения для различных архитектур и платформ.

Формируемые дисциплиной «Кроссплатформенное программирование» знания и умения готовят выпускника данной образовательной программы к выполнению следующих обобщенных трудовых функций (трудовых функций):

- Анализ требований к программному обеспечению (ПС 06.001 “Программист», ТФ D/01.6.);
- Проектирование программного обеспечения (ПС 06.001 “Программист», ТФ D/03.6.);
- Оценка архитектуры с точки зрения прослеживаемости требований:
  - согласованность с системными требованиями;
  - приспособленность стандартов и методов проектирования;
  - осуществимость функционирования и сопровождения;
  - осуществимость программных составных частей, полностью удовлетворяющих назначенным требованиям (ПС 06.003 “Архитектор программного обеспечения”, D/03.5);
- Разработка концепции системы (ПС 06.022 “Системный аналитик”, ТФ С/05.6);
- Организация оценки соответствия требованиям существующих систем и их аналогов (ПС 06.022 “Системный аналитик”, ТФ С/07.6).

## 2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина «Кроссплатформенное программирование» относится к части, формируемой участниками образовательных отношений, Блока 1 «Дисциплины (модули) ОПОП».

Для освоения дисциплины бакалавры используют знания, умения, навыки, сформированные в процессе изучения предметов «Языки и методы программирования», «Алгоритмы и алгоритмические языки», «Объектно-ориентированное программирование» соответствующего направления подготовки бакалавров.

Освоение данной дисциплины является основой для последующего прохождения практик: Учебной (проектной (проектно-технологической)) практики, Производственной (преддипломной) практики, Производственной практики (научно-исследовательской работы), выполнению и защите выпускной квалификационной работы.

## 3. Результаты освоения дисциплины «Кроссплатформенное программирование»

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО по данному направлению:

Коды компет енции	Наименование компетенции	Индикатор достижения компетенции	В результате освоения дисциплины обучающийся должен:
ПК-2	<b>Способен создавать и исследовать математические модели в естественных науках, промышленности и бизнесе, с учетом возможностей современных</b>	<i>ПК-2.1. Анализирует возможности современных информационных технологий и языков программирования и компьютерной техники</i>	Знать: знания естественных наук, математики и информатики, основные факты, концепции, принципы теорий при разработке кроссплатформенного программного обеспечения Уметь: использовать базовые знания естественных наук, математики и информатики,

	<b>информационных технологий и языков программирования и компьютерной техники</b>		основные факты, концепции, принципы теорий, связанных с прикладной математикой и информатикой при разработке кроссплатформенного программного обеспечения Владеть: практическими навыками применения базовых знаний математики и информатики при разработке кроссплатформенного программного обеспечения
<i>ПК-3</i>	<b>Способен применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии</b>	<i>ПК-3.1. Анализирует возможности современных языков программирования и методов параллельной обработки данных</i>	Знать: современные языки программирования, используемые при кроссплатформенной разработке программного обеспечения Уметь: разрабатывать программное обеспечение посредством кроссплатформенного программирования с помощью современных языков и методов параллельной обработки данных Владеть: практическими навыками разработки программных продуктов с помощью кроссплатформенного программирования

#### 4. Структура и содержание дисциплины Кроссплатформенное программирование

##### 4.1. Структура дисциплины

Общая трудоемкость дисциплины составляет 2 зачетные единицы, 72 часа.

№ п/п	Наименование разделов и тем дисциплины (модуля)	Семестр	Недели семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)								Формы текущего контроля успеваемости (по неделям семестра)			
				Контактная работа				Самостоятельная работа							
				Всего	Лекция	Лабораторные занятия	Др. виды контакт. работы	Всего	Подготовка к аудиторным занятиям	Выполнение домашнего индивидуального задания	Подготовка отчёта лабораторной работы	Опрос на лабораторном занятии	Проверка выполнения индивидуального задания	Проверка отчёта выполнения лабораторной работы	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
<b>1.</b>	<b>Раздел 1. Базовые концепции кроссплатформенного программирования</b>	<b>8</b>	<b>1-2</b>	<b>4</b>	<b>4</b>			<b>1,75</b>	<b>1,75</b>						
1.1.	Тема 1.1. Базовые концепции кроссплатформенного программирования		1-2	4	4			1,75	1,75						
<b>2.</b>	<b>Раздел 2. Основные возможности средств разработки Qt</b>	<b>8</b>	<b>1-3</b>	<b>6</b>	<b>2</b>	<b>4</b>		<b>4</b>	<b>1</b>	<b>2</b>	<b>1</b>				
2.1.	Тема 2.1. Основные возможности средств разработки Qt		3	2	2			1	1						
2.2	Лабораторная работа 1. Основы создания GUI приложений в системе Qt Creator		1-2	4		4		3		2	1	1	2	3	
<b>3</b>	<b>Раздел 3. Разработка элементов управления</b>	<b>8</b>	<b>3-5</b>	<b>8</b>	<b>4</b>	<b>4</b>		<b>6</b>	<b>2</b>	<b>3</b>	<b>1</b>				

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3.1.	Тема 3.1. Разработка элементов управления		4-5	4	4			2	2					
3.2.	Лабораторная работа 2. Создание и использование элементов управления		3-4			4		4		3	1	3	4	5
<b>4</b>	<b>Раздел 4. События и взаимодействие с пользователем</b>	<b>8</b>	<b>5-6</b>	<b>6</b>	<b>2</b>	<b>4</b>		<b>6</b>	<b>2</b>	<b>3</b>	<b>1</b>			
4.1.	Тема 4.1. События и взаимодействие с пользователем		6	2	2			2	2					
4.2.	Лабораторная работа 3. Реализация обработки событий в GUI приложениях		5-6	4		4		4		3	1	5	6	7
<b>5</b>	<b>Раздел 5. Графика и звук</b>	<b>8</b>	<b>7-8</b>	<b>8</b>	<b>4</b>	<b>4</b>		<b>6</b>	<b>2</b>	<b>3</b>	<b>1</b>			
5.1.	Тема 5.1. Графика и звук		7-8	4	4			2	2					
5.2.	Лабораторная работа 4. Реализация графического ввода-вывода в GUI приложениях		7-8	4		4		4		3	1	7	8	9
<b>6.</b>	<b>Раздел 6. Создание кроссплатформенных GUI приложений</b>	<b>8</b>	<b>9-10</b>	<b>8</b>	<b>4</b>	<b>4</b>		<b>6</b>	<b>2</b>	<b>3</b>	<b>1</b>			
6.1.	Тема 6.1. Создание кроссплатформенных GUI приложений		9-10	4	4			2	2					
6.2.	Лабораторная работа 5. Разработка GUI приложения с использованием подсистемы Graphics View		9-10	4		4		4		3	1	9	10	10
	Общая трудоемкость, в часах			42,25	20	20	2,25	29,75	10,75	14	5	Промежуточная аттестация		
												Форма	Семестр	
												Зачёт	8	
												Экзаме н		

## 4.2. Содержание дисциплины

### Раздел 1. Базовые концепции кроссплатформенного программирования

#### Тема 1.1. Базовые концепции кроссплатформенного программирования

Предмет, цели и задачи дисциплины. Основная терминология. Основные парадигмы разработки программного обеспечения. Технологии и средства разработки программного обеспечения. Встраиваемые системы.

### Раздел 2. Основные возможности средств разработки Qt

#### Тема 2.1. Основные возможности средств разработки Qt

Создание диалоговых окон: быстрое проектирование диалоговых окон, динамические диалоговые окна, встроенные классы виджетов и диалоговых окон. Создание главных окон: создание меню и панели инструментов, создание и настройка строки состояния, применение диалоговых окон, работа с несколькими документами. Реализация функциональности приложений: центральный виджет, хранение данных в объектах типа «элемент», загрузка и сохранение. Создание пользовательских виджетов, настройка виджетов Qt.

**Лабораторная работа 1.** Основы создания GUI приложений в системе Qt Creator

### Раздел 3. Разработка элементов управления

#### Тема 3.1. Разработка элементов управления

Класс *QWidget*. Стек виджетов. Рамки. Виджет видовой прокрутки. Управление автоматическим размещением элементов. Элементы отображения: надписи, индикатор прогресса, электронный индикатор. Кнопки, флажки и переключатели, группировки кнопок. Элементы настройки. Элементы ввода. Элементы выбора.

**Лабораторная работа 2.** Создание и использование элементов управления

### Раздел 4. События и взаимодействие с пользователем

#### Тема 4.1. События и взаимодействие с пользователем

Переопределение специализированных методов обработки событий. События клавиатуры. Событие обновления контекста рисования. События мыши. Событие таймера. Событие перетаскивания объектов. Реализация собственных классов событий. Переопределение метода *event()*. Фильтры событий. Искусственное создание событий.

**Лабораторная работа 3.** Реализация обработки событий в GUI приложениях

### Раздел 5. Графика и звук

#### Тема 5.1. Графика и звук

Классы геометрии: точка, прямоугольник, прямая линия, полигон. Растровые изображения. Работа со шрифтом. Графическое представление: сцена, представление, элемент. Работа с OpenGL. Вывод на печать. Элементы со стилем: встроенные стили, методы рисования простых элементов управления, использование каскадных стилей документа. Звук: воспроизведение звука, проверка возможности воспроизведения

**Лабораторная работа 4.** Реализация графического ввода-вывода в GUI приложениях

### Раздел 6. Создание кроссплатформенных GUI приложений

#### Тема 6.1. Создание кроссплатформенных GUI приложений

Управление сеансом. Организация буфера обмена. Реализация drag. Реализация drop. Подготовка приложения к интернационализации. Создание меню. Правила создания диалоговых окон. Стандартные диалоговые окна. Диалоговые окна сообщений. Реализация всплывающих подсказок. Область уведомлений. Виджет экрана. Класс сервиса рабочего стола.

## Лабораторная работа 5. Разработка GUI приложения с использованием подсистемы Graphics View

### 5. Образовательные технологии

В ходе освоения дисциплины «Кроссплатформенное программирование» при проведении аудиторных занятий используется образовательная технология, предусматривающая такие методы и формы изучения материала как лекция, лабораторное занятие, включающие в том числе активные и интерактивные формы занятий:

- проведение интерактивной лекции (Тема 2.1. «Основные возможности средств разработки Qt»);
- проведение работы в парах (Лабораторная работа 5. «Разработка GUI приложения с использованием подсистемы Graphics View»).

Занятия, проводимые в интерактивной форме составляют 25 % от общего количества аудиторных занятий.

Самостоятельная работа студентов подразумевает работу под руководством преподавателя (консультации, помощь в написании и отладки программ и др.) и индивидуальную работу студента, выполняемую в том числе в компьютерном классе с выходом в Интернет на факультете вычислительной техники университета.

При реализации образовательных технологий используются следующие виды самостоятельной работы:

- работа с конспектом лекции (обработка текста);
- подготовка к лабораторной работе;
- выполнение индивидуального задания по лабораторным работам;
- поиск информации в сети «Интернет» и литературе;
- подготовка к сдаче зачёта.

В целях реализации индивидуального подхода к обучению студентов, осуществляющих учебный процесс по собственной траектории в рамках индивидуального рабочего плана, изучение данной дисциплины базируется на следующих возможностях: обеспечение внеаудиторной работы со студентами в том числе в электронной образовательной среде с использованием соответствующего программного оборудования, дистанционных форм обучения. возможностей интернет-ресурсов, индивидуальных консультаций и т.д.

В качестве других видов контактной работы запланированы консультации при подготовке и проведении текущей и промежуточной аттестации.

### 6. Учебно-методическое обеспечение самостоятельной работы студентов.

#### Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

##### 6.1. План самостоятельной работы студентов

№ нед.	Тема	Вид самостоятельной работы	Задание	Рекомендуемая литература	Количество часов
1	2	3	4	5	6
1-2	1.1	<i>Подготовка к аудиторному занятию</i>	Рассмотреть основные парадигмы разработки программного обеспечения, технологии и средства разработки программного обеспечения. Дополнительно ознакомится с историческими аспектами	1,2,3	1,75

			развития кроссплатформенного программирования.		
	<b>2.2</b>	<i>Подготовка к лабораторной работе. Выполнение индивидуального задания Оформление отчета по лабораторной работе</i>	Изучить среду разработки приложений Qt Creator Разработать программу согласно индивидуальному заданию Подготовить отчёт по лабораторной работе	<b>1,2,3</b>	<b>2</b>  <b>1</b>
<b>3-4</b>	<b>2.1</b>	<i>Подготовка к аудиторному занятию</i>	Изучить способы создание диалоговых окон, главных окон, пользовательских виджетов	<b>1,2,3</b>	<b>1</b>
	<b>3.2</b>	<i>Подготовка к лабораторной работе. Выполнение индивидуального задания Оформление отчета по лабораторной работе</i>	Изучить примитивы создания элементов управления в GUI приложениях Разработать программу согласно индивидуальному заданию Подготовить отчёт по лабораторной работе	<b>1,2,3</b>	<b>3</b>  <b>1</b>
<b>4-5</b>	<b>3.1</b>	<i>Подготовка к аудиторному занятию</i>	Изучить способы создания элементов отображения, элементов настройки, элементов ввода, элементов вывода. Рассмотреть возможность управления автоматическим размещением элементов управления.	<b>1,2,3</b>	<b>2</b>
<b>5-6</b>	<b>4.2</b>	<i>Подготовка к лабораторной работе. Выполнение индивидуального задания Оформление отчета по лабораторной работе</i>	Изучить примитивы обработки событий от устройств. Разработать программу согласно индивидуальному заданию Подготовить отчёт по лабораторной работе	<b>1,2,3</b>	<b>3</b>  <b>1</b>
	<b>4.1</b>	<i>Подготовка к аудиторному занятию</i>	Изучить методы обработки событий клавиатуры, обновления контекста, мыши, таймера, перетаскивания объектов.	<b>1,2,3</b>	<b>2</b>
<b>7-8</b>	<b>5.1</b>	<i>Подготовка к аудиторному занятию</i>	Изучить работу со шрифтами, с элементами со стилями, со звуком.	<b>1,2,3</b>	<b>2</b>
	<b>5.2</b>	<i>Подготовка к лабораторной работе. Выполнение индивидуального задания Оформление отчета по лабораторной работе</i>	Изучить примитивы организации графического ввода/вывода в GUI-приложениях. Разработать программу согласно индивидуальному заданию Подготовить отчёт по лабораторной работе	<b>1,2,3</b>	<b>3</b>  <b>1</b>
<b>9-10</b>	<b>6.1</b>	<i>Подготовка к аудиторному занятию</i>	Изучить организация буфера обмена, реализацию drag и drop, правила создания диалоговых окон, реализацию всплывающих подсказок	<b>1,2,3</b>	<b>2</b>



<b>6.2</b>	<i>Подготовка к лабораторной работе. Выполнение индивидуального задания Оформление отчета по лабораторной работе</i>	Изучить примитивы создания диалоговых окон, реализации всплывающих подсказок. Разработать программу согласно индивидуальному заданию Подготовить отчёт по лабораторной работе	<b>1,2,3</b>	<b>3</b>  <b>1</b>
------------	--	---	--------------	--------------------------

## **6.2. Методические указания по организации самостоятельной работы студентов**

Планируются следующие виды самостоятельной работы:

- подготовка к лабораторным работам занятиям,
- выполнение индивидуального задания,
- оформление отчётов по лабораторным работам,
- подготовка к зачёту,
- работа с конспектом лекций и изучение литературы при подготовке к зачёту.

Подготовка к лабораторным занятиям проводится посредством изучения курса лекций, дополнительной литературы, Интернет-ресурсов.

Задание к выполнению каждой лабораторной работы состоит с общей части, которая сформулирована в разделе "Задание к выполнению" и уточнения варианта, который приведён в разделе "Варианты заданий". Студент должен заранее ознакомиться со своим заданием и, если у него возникают какие-либо вопросы относительно задания, поставить эти вопросы преподавателю до начала работы.

Выполнение начинается с разработки алгоритма решения задачи. На этом этапе должны быть детально проанализированы условия задания. Хотя представление схемы алгоритма не является обязательным элементом проектирования программы, но схема является удобным инструментом для осмысления задачи и оптимизации решения. При разработке алгоритма следует уделять внимание его упрощению, минимизации объёма вычислений, удалению лишних операций и т.п.

Написание текста программы начинается с определения переменных, которые необходимы для функционирования алгоритма. Большая часть переменных может быть определена ещё на этапе проектирования схемы алгоритма.

Для каждой переменной необходимо определить её тип и тщательно проверить, удовлетворяет ли диапазон значений выбранного типа тем значениям, которые может реально принимать переменная. Для массивов и символьных строк следует убедиться, что их размерность соответствует возможным размерам агрегаций данных.

Если схема алгоритма сделана достаточно тщательно, написание кодовой части программы сводится к записи каждого элемента схемы оператором языка программирования. Если в составе оператора встречается обращение к функции, следует проверить соответствие состава, последовательности и типов параметров, и возвращаемого значения спецификациям функции. Одновременно необходимо убедиться, что был включён файл-заголовок с описанием этой функции в программу.

Подготовленный текст программы следует набрать в текстовом редакторе и сохранить его в файле, который затем необходимо скомпилировать, чтобы получить исполняемый файл.

Отчёт к лабораторной работе должен содержать:

- Тему работы
- Цель работы
- Задание для выполнения, включая индивидуальное задание
- Описание алгоритма программы
- Описание переменных и структур данных, которые применяются в программе
- Описание ключевых программных решений, принятых при реализации алгоритма в тексте программы

- Текст программы
- Результат работы программы
- Выводы

Подготовка к зачёту проводится посредством изучения курса лекций, изучения дополнительной литературы, Интернет-ресурсов.

### 6.3. Материалы для проведения текущего и промежуточного контроля знаний студентов

#### *Контроль освоения компетенций*

№ п/п	Вид контроля	Контролируемые темы (разделы)	Компетенции, компоненты которых контролируются
1	Текущий: проверка выполнения индивидуального задания на лабораторной работе	Разделы 1 – 6	ПК-2,3
2	Текущий: собеседование при защите лабораторных работ	Разделы 1 – 6	ПК-2,3
3	Промежуточный: зачёт по результатам выполнения лабораторных работ	Разделы 1 – 6	ПК-2,3

*Материалы для проведения текущего контроля знаний и промежуточной аттестации составляют отдельный документ – Фонд оценочных средств по дисциплине «Современное программирование».*

*Демонстрационные варианты оценочных средств для каждого вида контроля доступны в ЭИОС (<http://moodle.pnzgu.ru>) в разделе Оценочные средства по дисциплине в курсе «Современное программирование».*

#### **Перечень примерных вопросов к зачёту**

1. Основные парадигмы разработки программного обеспечения
2. Технологии и средства разработки программного обеспечения
3. Создание диалоговых окон
4. Создание главных окон
5. Создание пользовательских виджетов
6. Создание виджета видовой прокрутки
7. Управление автоматическим размещением элементов
8. Создание элементов отображения
9. Создание элементов настройки
10. Создание элементов ввода
11. Создание элементов выбора
12. Переопределение специализированных методов обработки событий
13. Обработка событий клавиатуры
14. Обработка событий обновления контекста рисования
15. Обработка событий мыши
16. Вывод объектов на печать
17. Реализация drag, drop
18. Реализация диалоговых окон сообщений
19. Реализация всплывающих подсказок
20. Работа с виджетом экрана

## 7. Учебно-методическое и материально-техническое обеспечение дисциплины «Кроссплатформенное программирование»

а) учебная литература:

1. Иванова Г.С. Технология программирования: учебник. – М.: Изд-во МГТУ им. Баумана, 2006. (17 экземпляров) [http://kleopatra.pnzgu.ru/cgi-bin/irbis64r\\_91/cgiirbis\\_64.exe?P21DBN=KATL&I21DBN=KATL\\_PRINT&S21FMT=fullw\\_print&C21COM=F&Z21MFN=7433](http://kleopatra.pnzgu.ru/cgi-bin/irbis64r_91/cgiirbis_64.exe?P21DBN=KATL&I21DBN=KATL_PRINT&S21FMT=fullw_print&C21COM=F&Z21MFN=7433)
2. Программирование на языке С++ в среде Qt Creator / Е. Алексеев, Г. Злобин, Д. Костюк, О. Чеснокова, А. Чмыхало: курс лекций. НОУ Интуит <http://www.intuit.ru/studies/courses/3479/721/info> (неограниченно)
3. Павловская Т.А. С/С++. Программирование на языке высокого уровня: Учебник для бакалавров и магистров. – СПб.: Питер, 2012. (5 экземпляров) [http://kleopatra.pnzgu.ru/cgi-bin/irbis64r\\_91/cgiirbis\\_64.exe?P21DBN=KATL&I21DBN=KATL\\_PRINT&S21FMT=fullw\\_print&C21COM=F&Z21MFN=14891](http://kleopatra.pnzgu.ru/cgi-bin/irbis64r_91/cgiirbis_64.exe?P21DBN=KATL&I21DBN=KATL_PRINT&S21FMT=fullw_print&C21COM=F&Z21MFN=14891)

б) Интернет-ресурсы <http://crossplatform.ru>,  
<http://www.programmersforum.ru/forumdisplay.php?f=40>.

в) Программное обеспечение:

Все работы выполняются на персональных компьютерах под управлением ОС Windows, Ubuntu в среде Qt Creator на языке С++.

г) Другое материально-техническое обеспечение

При проведении лабораторных занятий используются рабочие места в компьютерном классе, оборудованном локальной сетью и выходом в Internet, имеющиеся в библиотеке учебники. Все лабораторные работы выполняются на персональных компьютерах.

Рабочая программа дисциплины «Кроссплатформенное программирование» составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 01.03.02 «Прикладная математика и информатика», утвержденного приказом Министерства образования и науки Российской Федерации от «10» января 2019 г. № 9.

Программу составили:

1 Абрамов И.А., доцент кафедры КТ

**Настоящая программа не может быть воспроизведена ни в какой форме без предварительного письменного разрешения кафедры-разработчика программы.**

Программа одобрена на заседании кафедры «Компьютерные технологии»


Протокол № 12 от «26» июня 2019 года

Зав. кафедрой «Компьютерные технологии» \_\_\_\_\_  В. И. Горбаченко

Программа одобрена методической комиссией факультета вычислительной техники

Протокол № 10 от «03» июня 2019 года

Председатель методической комиссии  
факультета вычислительной техники

\_\_\_\_\_   
(подпись)

Т. В. Глотова  
(Ф.И.О.)