

**Белорусский государственный университет**

**УТВЕРЖДАЮ**

Проректор по учебной работе и  
образовательным инновациям

О. Г. Прохоренко

«01» декабря 2022 г.

Регистрационный № УД – 11328/уч.

**РАЗРАБОТКА КРОСС-ПЛАТФОРМЕННЫХ ПРИЛОЖЕНИЙ**

**Учебная программа учреждения высшего образования  
по учебной дисциплине для специальностей:**

**1-31 03 06 Экономическая кибернетика (по направлениям)**

направление специальности:

1-31 03 06-01 Экономическая кибернетика (математические методы и  
компьютерное моделирование в экономике)

**1-31 03 07 Прикладная информатика (по направлениям)**

направление специальности:

1-31 03 07-01 Прикладная информатика (программное обеспечение  
компьютерных систем)

**1-98 01 01 Компьютерная безопасность (по направлениям)**

направление специальности:

1-98 01 01-01 Компьютерная безопасность (математические методы и  
программные системы)

Учебная программа составлена на основе образовательных стандартов высшего образования ОСВО 1-31 03 06-2021, ОСВО 1-31 03 07-2021, ОСВО 1-98 01 01-2021, типовых учебных планов №G 31-1-028/пр-тип от 30.06.2021, №G 31-1-030/пр-тип от 01.07.2021, №P 98-1-003/пр-тип от 02.07.2021, учебных планов № G 31-1-215/уч. от 22.03.2022, №G 31-1-216/уч. от 22.03.2022, №P 98-1-206/уч. от 22.03.2022; №G31-1-224 уч. ин. от 27.05.2022

### **СОСТАВИТЕЛИ:**

**Н.К. Рубашко** – старший преподаватель кафедры информационных систем управления Белорусского государственного университета

**В.В. Конах** – старший преподаватель кафедры информационных систем управления Белорусского государственного университета

### **РЕЦЕНЗЕНТЫ:**

**С.В. Абламейко** – профессор кафедры веб-технологий и компьютерного моделирования механико-математического факультета БГУ, академик НАН Беларуси, доктор технических наук, профессор

**В.А. Головко** – заведующий кафедрой интеллектуальных информационных технологий Брестского государственного технического университета, доктор технических наук, профессор

### **РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:**

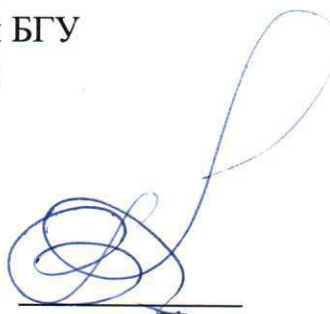
Кафедрой информационных систем управления Белорусского государственного университета

(протокол № 6 от 24.11.2022 г.)

Научно-методическим Советом БГУ

(протокол № 2 от 29.11.2022 г.)

Заведующий кафедрой



В.В. Краснопрошин

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебная дисциплина «Разработка кросс-платформенных приложений» является общепрофессиональной и направлена на формирование у студента фундамента понимания основ программирования на различных платформах. Потребность в эффективных средствах разработки программного обеспечения привела к появлению объектно-ориентированных систем программирования, предназначенных для быстрой разработки программных приложений.

Содержание дисциплины охватывает круг вопросов, связанных с разработкой программного обеспечения, способного работать под управлением различных операционных систем. В качестве среды разработки кроссплатформенных приложений рассматривается среда разработки Qt для языка программирования C++.

Основой для обучения являются компетенции, сформированные при изучении дисциплины «Основы и методология программирования».

### Цели и задачи учебной дисциплины

**Цель** преподавания учебной дисциплины «Разработка кросс-платформенных приложений» – теоретическая и практическая подготовка, обеспечивающая получение знаний по кроссплатформенному программированию, формирование практических навыков разработки кросс-платформенных решений, применения инструментальных средств разработки и отладки кроссплатформенного программного обеспечения, использования объектно-ориентированного и функционального подходов к решению практических задач.

### Задачи учебной дисциплины

1. Подготовка специалиста, уверенно владеющего разнообразными возможностями, предоставляемыми современными компьютерными технологиями и грамотно ориентирующегося в них.
2. Обучение студентов работе со средой программирования, предназначенной для разработки кроссплатформенных решений.
3. Овладение методологией разработки программ на основе императивного и объектно-ориентированного подхода.
4. Формирование навыков разработки современного востребованного кроссплатформенного программного обеспечения.

### Место учебной дисциплины в системе подготовки специалиста с высшим образованием

Учебная дисциплина относится к модулю «Программирование» государственного компонента.

Программа составлена с учётом межпредметных **связей** с учебными дисциплинами.

Дисциплина «Разработка кросс-платформенных приложений» непосредственно связана с параллельно изучаемыми дисциплинами:

– «Линейная алгебра», «Дифференциальное и интегральное исчисление» модуля «Высшая математика» для специальностей 1-31 03 06 Экономическая кибернетика (по направлениям), направление специальности 1-31 03 06-01 Экономическая кибернетика (математические методы и компьютерное моделирование в экономике), 1-98 01 01 Компьютерная безопасность (по направлениям), направление специальности 1-98 01 01-01 Компьютерная безопасность (математические методы и программные системы);

– «Алгебра и теория чисел», «Математический анализ» модуля «Высшая математика» для специальности 1-31 03 07 Прикладная информатика (по направлениям), направление специальности 1-31 03 07-01 Прикладная информатика (программное обеспечение компьютерных систем);

– «Дискретная математика и математическая логика» модуля «Дискретная математика и алгоритмы» для специальностей 1-98 01 01 Компьютерная безопасность (по направлениям), направление специальности 1-98 01 01-01 Компьютерная безопасность (математические методы и программные системы) и 1-31 03 07 Прикладная информатика (по направлениям), направление специальности 1-31 03 07-01 Прикладная информатика (программное обеспечение компьютерных систем).

Методы, излагаемые в указанных дисциплинах, используются для проектирования алгоритмов и программных приложений.

Сформированные при изучении дисциплины «Разработка кросс-платформенных приложений» компетенции являются основой для дальнейшего изучения дисциплин:

– модуля «Программирование»: «Промышленное программирование», «Технологии программирования» для специальностей 1-31 03 06 Экономическая кибернетика (по направлениям), направление специальности 1-31 03 06-01 Экономическая кибернетика (математические методы и компьютерное моделирование в экономике), 1-31 03 07 Прикладная информатика (по направлениям), направление специальности 1-31 03 07-01 Прикладная информатика (программное обеспечение компьютерных систем) и 1-98 01 01 Компьютерная безопасность (по направлениям), направление специальности 1-98 01 01-01 Компьютерная безопасность (математические методы и программные системы);

– модуля «Дискретная математика и алгоритмы»: «Дискретная математика и математическая логика» и «Алгоритмы и структуры данных» для специальностей 1-31 03 06 Экономическая кибернетика (по направлениям), направление специальности 1-31 03 06-01 Экономическая кибернетика (математические методы и компьютерное моделирование в экономике), 1-31 03 07 Прикладная информатика (по направлениям), направление специальности 1-31 03 07-01 Прикладная информатика (программное обеспечение компьютерных систем) и 1-98 01 01 Компьютерная безопасность (по направлениям), направление специальности 1-98 01 01-01 Компьютерная безопасность (математические методы и программные системы);

– модуля «Информатика и компьютерные системы»: «Модели данных и СУБД» и «Операционные системы» для специальностей *1-31 03 06 Экономическая кибернетика (по направлениям), направление специальности 1-31 03 06-01 Экономическая кибернетика (математические методы и компьютерное моделирование в экономике) и 1-98 01 01 Компьютерная безопасность (по направлениям), направление специальности 1-98 01 01-01 Компьютерная безопасность (математические методы и программные системы)*;

– модуля «Информационные системы»: «Модели данных и СУБД» для специальности *1-31 03 07 Прикладная информатика (по направлениям), направление специальности: 1-31 03 07-01 Прикладная информатика (программное обеспечение компьютерных систем)*.

Знания, полученные в учебной дисциплине, используются при изучении всех дисциплин специализации, при выполнении курсовых и дипломных работ, а также используются как инструментарий для моделирования и компьютерного решения задач ряда математических дисциплин, изучаемых на старших курсах.

### **Требования к компетенциям**

Освоение учебной дисциплины «Разработка кросс-платформенных приложений» должно обеспечить формирование следующей **базовой профессиональной компетенции**:

БПК-2. Строить, анализировать и тестировать алгоритмы и программы решения типовых задач обработки информации с использованием структурного, объектно-ориентированного и иных парадигм программирования.

В результате освоения учебной дисциплины студент должен:

#### **знать:**

- основные аспекты концепции кросс-платформенного программирования;
- архитектуру и компоненты кросс-платформенных решений;
- особенности программирования для кросс-платформенных решений;
- концепцию и особенности применения функционального программирования;
- особенности программирования с использованием основных компонентов библиотеки Qt;

#### **уметь:**

- разрабатывать программное обеспечение, используя кроссплатформенный подход;
- применять функциональный подход при разработке программного обеспечения;

***владеть:***

- навыками работы с объектно-ориентированными языками программирования;
- инструментарием разработки кросс-платформенных приложений.

**Структура учебной дисциплины**

Дисциплина изучается во 2-м семестре. Всего на изучение учебной дисциплины «Разработка кросс-платформенных приложений» отведено:

– для очной формы получения высшего образования 216 часов, в том числе 102 аудиторных часа, из них: лекций – 42 часа, лабораторных занятий – 54 часа, управляемая самостоятельная работа – 6 часов.

Трудоемкость учебной дисциплины составляет 6 зачетных единиц.

Форма текущей аттестации – экзамен.

## СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

### Раздел 1. Основы обобщенной парадигмы программирования

#### *Тема 1.1. Шаблоны как поддержка обобщенного программирования*

Множественная реализация функций в C++. Методы решения этой проблемы. Определение шаблонов функций. Параметры шаблонов функций. Выведение типа параметров шаблона по типам аргументов при вызове функции. Переопределение шаблонов функций.

Определение шаблонов классов. Описание и конкретизация шаблона. Требования к параметрам шаблона. Явное и неявное указание параметров при конкретизации шаблонов. Специализация шаблонов. Создание объектов по шаблону.

Особенности компиляции шаблонов. Возникновение ошибок компиляции.

### Раздел 2. Расширенная парадигма объектно-ориентированного программирования

#### *Тема 2.1. Наследование и полиморфизм как базовые понятия в парадигме объектно-ориентированного программирования*

Основные принципы и правила наследования. Понятие производного класса. Базовый класс и атрибуты его доступа. Иерархия производных классов. Конструкторы производных классов. Реализация наследования в языке программирования C++. Параметризация и наследование.

Основные принципы и правила полиморфизма. Виртуальные функции. Виртуальные деструкторы. Указатели объектов производного и базовых классов. Динамическое связывание.

Понятие обрезки объектов. Обрезка векторов. Класс `std::reference_wrapper`.

#### *Тема 2.2. Абстрактные классы. Семантика перемещения*

Абстрактные классы. Виртуальные базовые классы. Множественное наследование. Полиморфизм времени компиляции (параметрический полиморфизм).

Умные указатели и семантика перемещения.

#### *Тема 2.3. Динамическая идентификация и приведение типов*

Явное приведение типов в языке C++. Сведения о типах времени выполнения. Класс `type_info`. Оператор `typeid()` в приведении типов.

Полиморфизм и приведение типов. Оператор приведения `dynamic_cast`: необходимость и использование. Динамическое приведение ссылки. Операторы приведения типов `static_cast`, `const_cast`, `reinterpret_cast`. Краткая характеристика. Особенности использования.

## Раздел 3. Паттерны проектирования

### *Тема 3.1. Введение в паттерны проектирования*

Понятие паттерна проектирования. Описание паттернов проектирования. Использование языка UML для представления диаграмм классов. Методология решения задач проектирования с помощью паттернов. Выбор подходящих паттернов. Технология использования паттерна.

### *Тема 3.2. Паттерны объектно-ориентированного проектирования*

Классификация паттернов объектно-ориентированного проектирования. Порождающие паттерны. Структурные паттерны. Паттерны поведения. Общие сведения. Области применения.

Особенности реализации паттерна Iterator.

### *Тема 3.3. Архитектура проектирования*

Общие сведения об архитектурных паттернах. Паттерн Модель-Контроллер-Представление (MVC). Особенности разработки MVC-приложения.

## Раздел 4. Концепции и современные средства кроссплатформенного программирования

### *Тема 4.1. Основы кроссплатформенного программирования*

Определение кросс-платформенности. Уровни кросс-платформенности: аппаратный, программный, компиляции, выполнения. Проблемы кроссплатформенной разработки.

Языки, реализующие кросс-платформенность на уровне компиляции: C, C++. Языки, реализующие кросс-платформенность на уровне выполнения: Java, JavaScript, PHP, C#. Кросс-платформенные среды разработки.

Подходы к кросс-платформенному программированию. Кросс-платформенный пользовательский интерфейс и проблемы его создания.

### *Тема 4.2. Qt – кросс-платформенный инструментальный разработчик на языке программирования C++*

Интегрированная среда разработки Qt Creator. Модульная архитектура и классы Qt. Особенности. Основные составляющие. Пространство имен.

Инструментарий разработки. Дизайнер, отладчик. Методы отладки.

Структура Qt-проекта. Рекомендации для проекта Qt. Глобальные определения Qt. Библиотека Qt.

Создание простого приложения с использованием Qt.

### *Тема 4.3. Философия объектной модели Qt*

Класс QObject. Понятие свойств класса QObject. Механизм сигналов и слотов для организации взаимодействия объектов в Qt.

Соединение и разъединение объектов. Организация объектных иерархий. Метаобъектная информация.



#### ***Тема 4.4. Создание графического интерфейса средствами Qt***

Базовые элементы пользовательского интерфейса, предоставляемые Qt. Виджеты – элементы управления для организации графического интерфейса. Класс QWidget. Иерархия виджетов. Политики размера. Элементы отображения. Надписи. Индикатор выполнения. Кнопки, флажки и переключатели.

Управление автоматическим размещением элементов. Менеджеры компоновки. Горизонтальное и вертикальное размещение. Класс QVBoxLayout. Вложенные размещения. Табличное размещение QGridLayout. Разделители QSplitter.

Элементы ввода. Однострочное текстовое поле. Редактор текста.

Элементы выбора как базовые классы архитектуры «модель-представление».

#### ***Тема 4.5. Библиотека контейнеров Qt***

Контейнерные классы. Итераторы. Итераторы в стиле Java. Итераторы в стиле STL. Ключевое слово foreach.

Последовательные контейнеры: вектор QVector, массивы байтов QByteArray, битов QVector, списки QList и QListedList, стек QStack, очередь QQueue.

Ассоциативные контейнеры: словари QMap и QMapMultiMap, хэши QMapHash и QMapMultiHash, множество QMapSet.

Алгоритмы: сортировка, поиск, сравнение, заполнение значениями, копирование значений элементов, подсчет значений.

Строки. Регулярные выражения. Произвольный тип QVariant. Модель общего использования данных.

#### ***Тема 4.6. Диалоговые окна***

Правила создания диалоговых окон. Класс QDialog. Модальные диалоговые окна. Немодальные диалоговые окна. Создание собственного диалогового окна.

Стандартные диалоговые окна: выбора файлов, цвета, шрифта, окно ввода.

Диалоговые окна сообщений: информационного, предупреждающего, критического, сообщения о программе, сообщения об ошибке.

#### ***Тема 4.7. Работа с файлами, каталогами и потоками ввода/вывода***

Работа с файлами в Qt. Класс QFileDevice. Особенности использования класса QFile. Работа с каталогами. Класс QDir. Информация о файлах. Класс QFileInfo.

Потоки ввода/вывода. Назначение класса QTextStream. Понятие сериализации. Класс QDataStream как основа сериализации.

#### ***Тема 4.8. Главное окно, создание SDI- и MDI-приложений***

Класс главного окна QMainWindow. Класс действия QAction. Назначение и особенности работы. Панель инструментов. Доки. Строка состояния. Создание меню. «Анатомия» меню. Виды меню.

Модель событий в программах Qt. Цикл обработки событий. Класс QEvent. Существование двух механизмов: «слоты и сигналы» и «события». Переопределение специализированных методов обработки событий, события клавиатуры, событие обновления контекста рисования. События мыши. Общий алгоритм для программной обработки событий.

Реализация собственных классов событий. Переопределение метода event(). Сохранение работоспособности приложения. Искусственные события. Симуляция нажатия кнопки.

SDI- приложение, MDI-приложение. Создание SDI- и MDI-приложений. Быстрая разработка прототипов в Qt Designer.

#### ***Тема 4.9. Введение в компьютерную графику***

Характеристика основных классов компьютерной графики. Классы геометрии. Точка. Двумерный размер. Прямоугольник. Прямая линия. Многоугольник.

Работа с цветом. Класс QColor. Цветовая модель RGB. Цветовая модель HSV. Цветовая модель CMYK. Палитра.

Контекст рисования. Класс QPainter. Перо и кисть как средства рисования. Градиенты. Техника сглаживания. Рисование основных фигур: точек, линий, сплошных прямоугольников. Подходы к рисованию заполненных фигур. Запись команд рисования.

Трансформация систем координат. Перемещение. Масштабирование. Поворот. Скос. Графическая траектория. Отсечения. Режим совмещения. Графические эффекты.

#### ***Тема 4.10. Растровые изображения. Анимация***

Форматы графических файлов: BMP, GIF, PNG, JPEG, XPM. Контекстно-независимое представление. Класс QImage как контекст рисования. Контекстно-зависимое представление. Класс QPixmap. Класс QBitmap. Создание нестандартного окна виджета. Особенности работы со шрифтами. Отображение строки.

Анимация как событие таймера. Класс QTimer. Создание анимации с использованием таймера. Класс QMovie. SVG-графика. Анимационный движок и машина состояний. Смягчающие линии.

**УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ**  
 Дневная форма получения образования с применением электронных средств  
 обучения (ДО)

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСР	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
	<b>Разработка кросс-платформенных приложений</b>	<b>42</b>			<b>54</b>		<b>6</b>	
<b>1.</b>	<b>Основы обобщенной парадигмы программирования</b>	<b>4</b>			<b>2</b>		<b>2</b>	
1.1.	Шаблоны как поддержка обобщенного программирования	4			2		2	Электронный тест Контрольная работа по разделу 1
<b>2.</b>	<b>Расширенная парадигма объектно-ориентированного программирования</b>	<b>6</b>			<b>12</b>			
2.1.	Наследование и полиморфизм как базовые понятия в парадигме объектно-ориентированного программирования	2			4			Отчет по лабораторной работе
2.2.	Абстрактные классы. Семантика перемещения	2			4			Отчет по лабораторной работе
2.3.	Динамическая идентификация и приведение типов	2			4			Экспресс-опрос Контрольная работа по темам раздела 2
<b>3.</b>	<b>Паттерны проектирования</b>	<b>10</b>			<b>8</b>			
3.1.	Введение в паттерны проектирования	2						Тест
3.2.	Паттерны объектно-ориентированного проектирования	4			6			Отчет по лабораторной работе
3.3.	Архитектура проектирования	4			2			Отчет по лабораторной работе Коллоквиум по материалам раздела 3
<b>4.</b>	<b>Концепции и современные средства кросс-платформенного программирования</b>	<b>22</b>			<b>32</b>		<b>4</b>	
4.1.	Основы кросс-платформенного программирования	2						
4.2.	Qt–кросс-платформенный инструментарий разработки на языке программирования C++	1					2	Собеседование

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСР	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
4.3.	Философия объектной модели Qt	1			2			Дискуссия
4.4.	Создание графического интерфейса средствами Qt	4			6			Отчет по лабораторной работе
4.5.	Библиотека контейнеров Qt	2			2		2	Электронный тест
4.6.	Диалоговые окна	2			4			Отчет по лабораторной работе
4.7.	Работа с файлами, каталогами и потоками ввода/вывода	2			2			Отчет по лабораторной работе
4.8.	Главное окно, создание SDI- и MDI-приложений	4			6			Отчет по лабораторной работе
4.9.	Введение в компьютерную графику	2			4			Отчет по лабораторной работе
4.10.	Растровые изображения. Анимация	2			6			Отчет по лабораторной работе Контрольная работа по темам раздела 4

## ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

### Перечень основной литературы

1. Фримен, Э. Head First. Паттерны проектирования / Фримен Э., Робсон Э., Сьерра К., Бейтс Б. – СПб.: Питер. 2022. – 640 с.
2. Гамма, Э. Паттерны объектно-ориентированного проектирования / Гамма Э., Хелм Р., Джонсон Р., Влиссидес Д. – СПб.: Питер. Библиотека программиста, 2022. – 448 с.
3. Шлее, Макс. Qt 5.10. Профессиональное программирование на C++ / Макс Шлее. – Санкт-Петербург: БХВ-Петербург, 2022. –1072 с.
4. Прохоренок, Н. А. Qt 6. Разработка оконных приложений на C++ / Николай Прохоренок. – Санкт-Петербург: БХВ-Петербург, 2022. –512 с.

### Перечень дополнительной литературы

1. Тепляков, С. Паттерны проектирования на платформе .NET / Тепляков С. – СПб.: Питер, 2015. – 320 с.
2. Алексеев, Е. Программирование на языке C++ в среде Qt Creator / Е.Р. Алексеев, Г.Г. Злобин, Д.А. Костюк, О.В. Чеснокова, А.С. Чмыхало – М.: ALT Linux, 2015. – 448 с.
3. Иванова, Г. Создание пользовательских интерфейсов в программах на C++ с использованием библиотеки Qt / Иванова Г.С. – Учебное пособие. – М. – 2011. 52 с.
4. Чакон, С. Git для профессионального программиста / Чакон С., Штрауб Б.. – СПб.: Питер. Библиотека программиста, 2022. – 496 с.
5. Хориков, В. Принципы юнит-тестирования / Хориков В. – СПб.: Питер. Для профессионалов, 2021. – 320 с.

### Электронные ресурсы

1. Погружение в паттерны проектирования / Швец А. – Электронная книга о принципах и паттернах проектирования [Электронный ресурс]. – Режим доступа: <https://refactoring.guru/ru/design-patterns/book>
2. Все о кроссплатформенном программировании / Справочная документация по Qt (Выпуск OpenSource) [Электронный ресурс]. – Режим доступа: <http://doc.crossplatform.ru/qt/4.3.2/index.html>
3. Руководство по Qt для начинающих / Князев А.А.[Электронный ресурс]. – Режим доступа: <http://knzsoft.ru/qt-bgr/>
4. Образовательный портал БГУ [Электронный ресурс]. – Режим доступа: <https://edufpmi.bsu.by/course/view.php?id=838>

5. Образовательная платформа InsightRunner [Электронный ресурс]. – Режим доступа: <https://acm.bsu.by>

### **Перечень рекомендуемых средств диагностики и методика формирования итоговой отметки**

Для диагностики компетенций в рамках учебной дисциплины рекомендуется использовать следующие формы:

1. Устная форма: собеседование, экспресс-опрос, дискуссия.
2. Письменная форма: контрольные работы, коллоквиум, тесты.
3. Устно-письменная форма: отчеты по лабораторным заданиям с их устной защитой.
4. Техническая форма: электронные тесты.

В качестве рекомендуемых технических средств диагностики используется обучение, организованное на платформе Moodle (<https://edufpmi.bsu.by>).

Формой текущей аттестации по дисциплине учебным планом предусмотрен экзамен.

При формировании итоговой отметки используется рейтинговая система оценки знаний студента, дающая возможность проследить и оценить динамику процесса достижения целей обучения. Рейтинговая система предусматривает использование весовых коэффициентов для текущего контроля знаний и текущей аттестации студентов по дисциплине.

Примерные весовые коэффициенты, определяющие вклад текущего контроля знаний и текущей аттестации в итоговую отметку:

- контрольные работы – 50 %;
- отчеты по лабораторным заданиям с их устной защитой – 30 %;
- выполнение тестов – 10%;
- коллоквиум – 10 %.

Итоговая отметка по дисциплине рассчитывается на основе отметки текущей успеваемости и экзаменационной отметки с учетом их весовых коэффициентов. Вес отметки по текущей успеваемости составляет 40 %, экзаменационной отметки – 60 %.

Точки контроля по текущей успеваемости формируются из расчета общего количества часов (зачетных единиц), выделенных на изучение дисциплины.

### **Примерный перечень заданий для управляемой самостоятельной работы студентов**

В качестве заданий для управляемой самостоятельной работы могут быть выданы задания для самостоятельного решения задач по следующим темам:

1. **Тема 1.1. Шаблоны как поддержка обобщенного программирования.** Шаблоны функций. Переопределение шаблонов функций. Шаблоны классов. Описание и конкретизация шаблона. Специализация шаблонов. (2 ч.)

**Форма контроля** – электронный тест на Образовательном портале БГУ.

2. **Тема 4.2.** *Qt – кроссплатформенный инструментарий разработки на языке программирования C++.* Среда разработки. Установка, настройка, создание проекта. (2 ч.)

**Форма контроля** – собеседование.

3. **Тема 4.5.** *Библиотека контейнеров Qt.* Алгоритмы: сортировка, поиск, сравнение, заполнение значениями, копирование значений элементов, подсчет значений. (2 ч.)

**Форма контроля** – электронный тест на Образовательном портале БГУ.

Варианты заданий выкладываются на портал <https://edufpmi.bsu.by/course/view.php?id=838>.

### **Примерная тематика лабораторных занятий**

Лабораторная работа 1. Наследование, полиморфизм, абстрактные классы.

Лабораторная работа 2. Применение паттерна Factory Method.

Лабораторная работа 3. Применение паттерна Composite.

Лабораторная работа 4. Структуры. Шаблон проектирования Iterator.

Лабораторная работа 5. Создание графического интерфейса средствами Qt.

Лабораторная работа 6. Контейнеры Qt и их визуализация с помощью QListWidget, QTableWidgetItem.

Лабораторная работа 7. Программирование игр (C++, Qt).

Лабораторная работа 8. Применение архитектуры MVC при разработке приложений.

Лабораторная работа 9. Разработка SDI-приложения.

Лабораторная работа 10. Рисование и анимация.

### **Рекомендуемая тематика контрольных работ и коллоквиума**

1) Контрольная работа №1 «Решение задач с использованием библиотеки STL».

2) Контрольная работа №2 «Разработка пользовательских классов с реализацией наследования и полиморфизма».

3) Коллоквиум «Паттерны проектирования».

4) Контрольная работа №3 «Разработка оконных приложений на базе Qt Creator».

### **Описание инновационных подходов и методов к преподаванию учебной дисциплины**

При организации образовательного процесса используются следующие методы:

– **метод учебной дискуссии**, который предполагает участие студентов в целенаправленном обмене мнениями, идеями для предъявления и/или согласования существующих позиций по определенной проблеме.

Использование метода обеспечивает появление нового уровня понимания изучаемой темы, применение знаний (теорий, концепций) при решении проблем, определение способов их решения;

– *метод группового обучения*, который представляет собой форму организации учебно-познавательной деятельности обучающихся, предполагающую функционирование разных типов малых групп, работающих как над общими, так и специфическими учебными заданиями.

В качестве технических средств для организации работы в рамках учебной дисциплины рекомендуется использовать Образовательный портал БГУ (<https://edufpmi.bsu.by>) – инструмент с эффективной функциональностью контроля, тренинга и самостоятельной работы;

– *практико-ориентированный подход*, который предполагает освоение содержания образования через решения практических задач; приобретение навыков эффективного выполнения разных видов профессиональной деятельности; ориентацию на генерирование идей, реализацию групповых студенческих проектов; использование процедур, способов оценивания, фиксирующих профессиональные компетенции.

### **Методические рекомендации по организации самостоятельной работы обучающихся**

Для организации самостоятельной работы студентов по учебной дисциплине следует использовать современные информационные ресурсы: разместить на образовательном портале комплекс учебных и учебно-методических материалов (учебно-программные материалы, учебное издание для теоретического изучения дисциплины, методические указания к лабораторным занятиям, материалы текущего контроля и текущей аттестации, позволяющие определить соответствие учебной деятельности обучающихся требованиям образовательных стандартов высшего образования и учебно-программной документации, в том числе вопросы для подготовки к зачету и экзамену, задания, тесты, вопросы для самоконтроля и др., список рекомендуемой литературы, информационных ресурсов и др.).

При составлении заданий УСП по учебной дисциплине необходимо предусмотреть возрастание их сложности: от заданий, формирующих достаточные знания по изученному учебному материалу на уровне узнавания, к заданиям, формирующим компетенции на уровне воспроизведения, и далее к заданиям, формирующим компетенции на уровне применения полученных знаний.

Таким образом, задания УСП по учебной дисциплине рекомендуется делить на три модуля:

I. Задания, формирующие достаточные знания по изученному учебному материалу **на уровне узнавания:**



*Задание 1.* Приведите пример явного и неявного указания параметров при конкретизации шаблонов. В чем особенность специализации шаблонов для типа `char*`?

*Задание 2.* Что такое паттерны проектирования. Приведите примеры.

*Задание 3.* Назначение модели событий в программах Qt. Как работает цикл обработки событий? Охарактеризуйте класс `QEvent`.

### II. Задания, формирующие компетенции на уровне воспроизведения:

*Задание 1.* Решить задачу нахождения суммы элементов массива с использованием шаблонов.

*Задание 2.* Реализовать класс контейнерного типа. Структуры реализуются на базе динамического массива. К структуре данных применить паттерн `Iterator`, реализовав его как через методы (в стиле `Java`), так и через операции (в стиле `C++`).

*Задание 3.* Сформировать два списка. Для хранения использовать `QList`, для визуализации – `QListWidget`. Организовать обмен данными между списками.

### III. Задания, формирующие компетенции на уровне применения полученных знаний:

*Задание 1.* Реализовать класс `Stack / Deck / Queue / Vector / List` с использованием шаблонов.

*Задание 2.* Реализовать базовый класс: спортивный рекорд и производные классы: рекорды по бегу, тяжёлой атлетике, плаванию. Преобразовать решение с использованием паттерна `Factory Method`:

- 1) с помощью обобщенного конструктора;
- 2) с помощью классической реализации паттерна.

*Задание 3.* Написать программу для построения графика функции вида  $y=f(x)$ .

### Примерный перечень вопросов к экзамену

1. Многократная реализация функций в `C++`. Методы решения этой проблемы. Определение шаблона. Особенности компиляции шаблонов. Возникновение ошибок компиляции.

2. Описание и конкретизация шаблона. Требования к параметрам шаблона. Явное и неявное указание параметров при конкретизации шаблонов. Специализация шаблонов.

3. Семантика перемещения. Перемещающий конструктор и оператор присваивания.

4. Понятие обрезки объектов. Обрезка векторов. Класс `std::reference_wrapper`.

5. Явное приведение типов в языке `C++`. Операторы приведения. Сведения о типах времени выполнения. Класс `typeid`.

6. Оператор `typeid()` в приведении типов.

7. Оператор приведения `dynamic_cast`: необходимость и использование. Динамическое приведение ссылки.

8. Операторы приведения типов `static_cast`, `const_cast`, `reinterpret_cast`. Краткая характеристика. Особенности использования.
9. Понятие паттерна проектирования. Описание паттернов. Результаты применения.
10. Классификация паттернов.
11. Паттерн `Factory_Method`. Особенности реализации.
12. Паттерн `Composite` и его реализация.
13. Паттерн `Iterator`: назначение, структура, реализация с помощью методов и операторов.
14. Паттерн `Observer`: назначение, структура, особенности реализации.
15. Паттерн `Visitor`: назначение, структура, особенности реализации.
16. Архитектурный паттерн `MVC`. Структура взаимодействия.
17. Понятие кроссплатформенности. Кроссплатформенные языки программирования. Сценарии и интерпретируемые языки.
18. Подходы к кроссплатформенному программированию. Кроссплатформенные среды исполнения.
19. Видеоигры. Понятие кроссплатформенного пользовательского интерфейса.
20. Проблемы кроссплатформенной разработки.
21. `Qt` – фреймворк для разработки кроссплатформенных приложений. Особенности. Основные составляющие. Пространство имен.
22. Модуль `QtCore`. Особенности класса `QCoreApplication`.
23. Модуль `QtGui`. Классы `QGuiApplication` и `QApplication`.
24. Модуль `QtWidgets`. Основные классы.
25. Структура проекта `Qt`. Характеристика проектного файла. Компиляция проекта.
26. Философия объектной модели `Qt`. Класс `QObject`. Понятие свойств класса `QObject`.
27. Механизм сигналов и слотов. Реализация. Соединение объектов. Метод `connect()`. Разъединение объектов.
28. Организация объектных иерархий.
29. Создание графического интерфейса средствами `Qt`. Класс `QWidget`. Основные типы виджетов.
30. Иерархия виджетов. Политики размера.
31. Понятие компоновки. Способы размещения виджетов.
32. Менеджеры компоновки.
33. Основные классы компоновки. Краткая характеристика. Взаимодействие виджетов.
34. Элементы управления в `Qt`. Взаимодействие с пользователем.
35. Элементы выбора. Основные классы.
36. Последовательные контейнеры `Qt`.
37. Ассоциативные контейнеры `Qt`.

38. Особенности работы с итераторами в Qt.
39. Особенности реализации алгоритмов в Qt. Примеры алгоритмов.
40. Работа с текстовыми строками в Qt. Класс QString. Примеры операций над строками.
41. Работа с файлами в Qt. Класс QFileDevice.
42. Особенности использования класса QFile.
43. Работа с классом QDir.
44. Поток ввода-вывода. Назначение класса QTextStream.
45. Понятие сериализации. Класс QDataStream как основа сериализации.
46. Модальные и немодальные диалоговые окна. Особенности реализации и работы.
47. Диалоговые окна Qt. Правила создания. Класс QDialog.
48. Стандартное диалоговое окно выбора файлов.
49. Стандартные окна выбора цвета и шрифта.
50. Диалоговое окно ввода.
51. Диалоговые окна сообщений. Виды.
52. Модель событий в программах Qt. Цикл обработки событий. Класс QEvent.
53. События обработки клавиатуры.
54. События обработки мыши.
55. События обработки таймера и рисования.
56. Общий алгоритм для программной обработки событий.
57. Класс главного окна QMainWindow. Характеристика.
58. Создание меню. Виды меню.
59. Класс действия QAction. Назначение и особенности работы.
60. Характеристика SDI- и MDI-приложений.
61. Характеристика основных классов компьютерной графики.
62. Класс QColor. Основные цветовые модели.
63. Контекст рисования. Класс QPainter.
64. Перо и кисть как средства рисования.
65. Рисование основных фигур.
66. Подходы к рисованию заполненных фигур.
67. Особенности работы со шрифтами.
68. Основные форматы растровых изображений.
69. Контекстно-независимое представление. Класс QImage.
70. Контекстно-зависимое представление. Класс QPixmap.
71. Анимация как событие таймера.
72. Назначение класса QMovie. Особенности работы.
73. Анимационный движок и машина состояний.

### Примерная тематика заданий к экзамену

Решить задачу с визуализацией решения в оконном приложении и с использованием классов:

1. Два выпуклых многоугольника заданы на плоскости перечислением координат вершин в порядке обхода границы. Определить площади многоугольников и проверить, лежит ли один из них строго внутри другого. Если нельзя решить задачу, то выдать сообщение.

2. Определить радиус и центр окружности, проходящей по крайней мере через три различные точки заданного множества точек на плоскости и содержащей внутри наибольшее количество точек этого множества. Если нельзя решить задачу, то выдать сообщение.

3. Задано множество точек на плоскости. Найти все четверки точек, являющихся, вершинами квадратов. Найти квадрат, внутри которого лежит наибольшее количество точек множества. Если нельзя решить задачу, то выдать сообщение.

4. Найти три треугольника с вершинами в заданном множестве точек на плоскости так, чтобы второй треугольник лежал строго внутри первого, а третий внутри второго. Если нельзя решить задачу, то выдать сообщение.

5. Нарисовать три пересекающиеся окружности, закрасив каждую окружность выбранным цветом. Область пересечения должна быть цветом, являющимся миксом выбранных цветов. Цвет выбирать через стандартный диалог. Выполнять масштабирование и восстановление рисунка при изменении размеров экрана и при его сворачивании.

6. Нарисовать 4 непересекающихся прямоугольника. Цвет каждого выбирать через стандартный диалог. С изменением цвета - перерисовка. Выполнять масштабирование и восстановление рисунка при изменении размеров экрана и при его сворачивании.

**ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО  
ПО ИЗУЧАЕМОЙ УЧЕБНОЙ ДИСЦИПЛИНЕ  
С ДРУГИМИ ДИСЦИПЛИНАМИ СПЕЦИАЛЬНОСТИ**

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Машинно-ориентированное программирование	Информационных систем управления	Нет	Изменений не требуется (протокол № 6 от 24.11.2022 г.)
Операционные системы	Информационных систем управления	Нет	Изменений не требуется (протокол № 6 от 24.11.2022 г.)
Промышленное программирование	Информационных систем управления	Нет	Изменений не требуется (протокол № 6 от 24.11.2022 г.)
Технологии программирования	Информационных систем управления	Нет	Изменений не требуется (протокол № 6 от 24.11.2022 г.)

**ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ  
ПО ИЗУЧАЕМОЙ УЧЕБНОЙ ДИСЦИПЛИНЕ**

на \_\_\_\_ / \_\_\_\_ учебный год

№№ ПП	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры информационных систем управления (протокол № \_\_\_\_ от \_\_\_\_\_ 200\_ г.)

Заведующий кафедрой

\_\_\_\_\_ (степень, звание)      \_\_\_\_\_ (подпись)      \_\_\_\_\_ (И.О.Фамилия)

УТВЕРЖДАЮ  
Декан факультета

\_\_\_\_\_ (степень, звание)      \_\_\_\_\_ (подпись)      \_\_\_\_\_ (И.О.Фамилия)