

**Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Санкт-Петербургский политехнический университет Петра Великого»**

---

УТВЕРЖДАЮ  
Директор ИКНК  
\_\_\_\_\_ Д.П. Зегжда  
«17» июня 2024 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

**«Конструирование программного обеспечения»**

Разработчик	Высшая школа программной инженерии
Направление (специальность) подготовки	09.03.04 Программная инженерия
Наименование ООП	09.03.04_01 Технология разработки и сопровождения качественного программного продукта
Квалификация (степень) выпускника	<b>бакалавр</b>
Образовательный стандарт	<b>СУОС</b>
Форма обучения	<b>Очная</b>

СОГЛАСОВАНО  
Руководитель ОП  
\_\_\_\_\_ А.В. Петров  
«21» мая 2024 г.

Соответствует СУОС  
Утверждена протоколом заседания  
высшей школы "ВШПИ"  
от «21» мая 2024 г. № 1

РПД разработали:  
Специалист по учебно-методической работе 1 категории Т.А. Вишневская  
Директор, к.т.н., доц. П.Д. Дробинцев

## 1. Цели и планируемые результаты изучения дисциплины

### Цели освоения дисциплины

Сформировать у специалистов понимание современных методов конструирования программного обеспечения, знание метафор, позволяющих лучше понять разработку программного обеспечения, умение определять тип программного обеспечения, над которым планируется вести работы, используя при этом метрическую информацию, умение выбирать технологии и методики конструирования для данного типа программного обеспечения.

### Результаты обучения выпускника

Код	Результат обучения (компетенция) выпускника ООП
ПК-20	Способен проектировать компоненты системного программного обеспечения, прикладных приложений и систем
ИД-1 ПК-20	Проводит выбор стиля кодирования и шаблонов проектирования для каждого компонента

### Планируемые результаты изучения дисциплины

#### знания:

- Знает типичные конфигурации и принципы построения интерфейсов программных систем, общепринятые стили кодирования

#### умения:

- Умеет применять современные встроенные и скриптовые языки и использовать шаблоны проектирования

#### навыки:

- Владеет современными библиотеками, инструментами конфигурирования и средами программирования

## 2. Место дисциплины в структуре ООП

В учебном плане дисциплина «Конструирование программного обеспечения» относится к модулю «Модуль цифровых компетенций» / «Технологии и процессы разработки программного обеспечения» / «Проектирование и поддержка программных систем. Электив».

Изучение дисциплины базируется на результатах освоения следующих дисциплин:

- Архитектура компьютера
- Введение в профессиональную деятельность

### 3. Распределение трудоёмкости освоения дисциплины по видам учебной работы и формы текущего контроля и промежуточной аттестации

#### 3.1. Виды учебной работы

Виды учебной работы	Трудоёмкость по семестрам
	Очная форма
Лекционные занятия	14
Практические занятия	30
Самостоятельная работа	24
Промежуточная аттестация (зачет)	4
Общая трудоёмкость освоения дисциплины	72, ач
	2, зет

#### 3.2. Формы текущего контроля и промежуточной аттестации

Формы текущего контроля и промежуточной аттестации	Количество по семестрам
	Очная форма
Текущий контроль	
Расчетно-графические работы, шт.	1
Промежуточная аттестация	
Зачеты, шт.	1

### 4. Содержание и результаты обучения

#### 4.1 Разделы дисциплины и виды учебной работы

№ раздела	Разделы дисциплины, мероприятия текущего контроля	Очная форма		
		Лек, ач	Пр, ач	СР, ач
1.	Введение в общие вопросы конструирования программного обеспечения	1	2	1

2.	Метафоры, характеризующие разработку ПО: написание кода, выращивание системы, медленное приращение системы, построение ПО.	1	2	1
3.	Применение методов разработки ПО. Комбинирование метафор.	1	2	1
4.	Понятие предварительных условий, важность выполнения предварительных условий.	1	3	1
5.	Определение типа ПО. Критерии выбора. Связь типа ПО с понятиями последовательного и итеративного проекта.	1	2	1
6.	Предварительные условия, связанные с определением проблемы, предварительные условия, связанные с выработкой требований.	1	2	1
7.	Предварительные условия, связанные с разработкой архитектуры	1	2	1
8.	Оценка времени затрачиваемого на выполнение предварительных условий. Метрики.	1	2	1
9.	Выбор языка программирования	1	2	1
10.	Конвенции программирования	1	2	1
11.	Выбор технологии конструирования ПО.	1	2	1
12.	Выбор основных методик конструирования.	0	2	0
13.	Проблемы, связанные с проектированием ПО.	1	2	0
14.	Основные концепции проектирования. Компоненты проектирования. Эвристические принципы	1	1	0
15.	Методики проектирования	1	1	0
16.	Популярные методологии	0	1	13
<b>Итого по видам учебной работы:</b>		14	30	24
Зачеты, ач				0
<b>Часы на контроль, ач</b>				0
<b>Промежуточная аттестация (зачет)</b>		4		
<b>Общая трудоёмкость освоения: ач / зет</b>		72 / 2		

## 4.2. Содержание разделов и результаты изучения дисциплины

Раздел дисциплины	Содержание
<b>1. Введение в общие вопросы конструирования программного обеспечения</b>	<p>Понятие проблемы в контексте конструирования ПО, требований и их выработки, плана конструирования, архитектуры ПО, высокоуровневого проектирования, детального проектирования. Знания на уровне понятий, определений, описаний, формулировок. История развития подходов к конструированию ПО, актуальность дисциплины при постоянном увеличении сложности и объема кода в современных проектах; понятия требований, планов конструирования, архитектуры ПО, типов проектирования.</p>
<b>2. Метафоры, характеризующие разработку ПО: написание кода, выращивание системы, медленное приращение системы, построение ПО.</b>	<p>Основные сведения о методах повышения производительности ПО. Краткая характеристика достоинств, недостатков и ограничений каждого метода. Знания на уровне понятий, определений, описаний, формулировок. Терминология, используемая при описании метафор разработки ПО. Умения в решении задач. Выбирать метафору разработки ПО в зависимости от целей, рисков проекта и других факторов, влияющих на выбор.</p>
<b>3. Применение методов разработки ПО. Комбинирование метафор.</b>	<p>Связь этапа выбора метафор и аналитических инструментов. Знания на уровне понятий, определений, описаний, формулировок. Понятие интеллектуального инструментария. Умения в решении задач. Умение использовать аналитический инструментарий для выбора метафор разработки ПО.</p>
<b>4. Понятие предварительных условий, важность выполнения предварительных условий.</b>	<p>Актуальность выполнения предварительных условий при разработке современного промышленного ПО. Причины неполной подготовки в большом количестве проектов. Связь выполнения предварительных условий и этапа нахождения дефектов. Стоимость дефектов. Знание понятий, определений, описаний, формулировок. Понятие предварительных условий. Понятие дефекта. Умения в решении задач. Умение обосновать важность выполнения предварительных требований при разработке ПО. Умение оценить объем предварительных требований для конкретных проектов.</p>

<p><b>5. Определение типа ПО.</b>  <b>Критерии выбора. Связь типа ПО с понятиями последовательного и итеративного проекта.</b></p>	<p>Типы ПО: Бизнес-системы, системы целевого назначения, встроенные системы критические к качеству. Типичные приложения, модели жизненного цикла, планирование и управление, выработка требований, проектирование, конструирование, тестирование и гарантии качества, внедрения для каждого типа ПО. Описание итеративного и последовательного проекта.</p> <p>Знание понятий, определений, описаний, формулировок. Понятие бизнес-системы, системы целевого назначения, встроенной системы критической к качеству. Знание критериев каждого типа ПО. Описание итеративного и последовательного проекта. Умения в решении задач. Определить тип ПО, который планируется разработать. Выбрать итеративный или последовательный процесс для задачи. Обосновать решение.</p>
<p><b>6. Предварительные условия, связанные с определением проблемы, предварительные условия, связанные с выработкой требований.</b></p>	<p>Формулирование сути проблемы. Понятие видения проекта. Понятие официальных требований, работа с требованиями. Запрос на изменение. Контрольный список вопросов при внесении изменений в требования к системе.</p> <p>Знание понятий, определений, описаний, формулировок. Понятие требований, понятие видения проекта, понятие запроса на изменение. Умения в решении задач. Принимать или отвергать запрос на изменения в соответствии с контрольным списком вопросов.</p>
<p><b>7. Предварительные условия, связанные с разработкой архитектуры</b></p>	<p>Типичные компоненты архитектуры: организация программы, основные классы, организация данных, бизнес-правила, пользовательский интерфейс, управление ресурсами, безопасность, производительность, масштабируемость, взаимодействие с другими системами, интернационализация, ввод-вывод, обработка ошибок, отказоустойчивость. Возможность реализации архитектуры. Избыточная функциональность, повторное использование, стратегия изменений. Общее качество архитектуры.</p> <p>Знание понятий, определений, описаний, формулировок. Понятие компонентов архитектуры. Знание из чего складываются компоненты архитектуры. Умения в решении задач. Строить и критически оценивать архитектуру системы.</p>
<p><b>8. Оценка времени затрачиваемого на выполнение предварительных условий. Метрики.</b></p>	<p>Методы оценки сроков. Эвристические методы.</p> <p>Знание понятий, определений, описаний, формулировок. Понятие метрика. Умения в решении задач. Оценивать время, затрачиваемое на выполнение предварительных условий проекта.</p>

<b>9. Выбор языка программирования</b>	<p>Сравнение функциональности операторов высокоуровневых языков. Описание языков: Assembler, C, C++, C#, Cobol, Fortran, Java, JavaScript, Perl, PHP, Python, SQL, Visual Basic.</p> <p>Знание понятий, определений, описаний, формулировок. Понятие функциональности оператора. Знание сильных и слабых сторон рассмотренных языков программирования. Умения в решении задач. Умение выбирать язык программирования для решения конкретной задачи.</p>
<b>10. Конвенции программирования</b>	<p>Стили кодирования. Общепринятые стандарты кодирования. Автоматические средства контроля выполнения стандарта кодирования.</p> <p>Знание понятий, определений, описаний, формулировок. Понятие стиля кодирования, стандарта кодирования. Умения в решении задач. Умение использовать программу StyleCop для автоматического отслеживания следования стандарту кодирования.</p>
<b>11. Выбор технологии конструирования ПО.</b>	<p>Понятие архитектурных шаблонов. Понятие framework и его связь с архитектурными шаблонами.</p> <p>Знание понятий, определений, описаний, формулировок. Понятие архитектурного шаблона, понятие framework. Знание распространенных архитектурных шаблонов. Знание распространенных framework. Умения в решении задач. Умение выбрать framework для конкретной задачи</p>
<b>12. Выбор основных методик конструирования.</b>	<p>Краткое описание процесса создания ПО. Методологии scrum, agile, xp, waterfall. Инструментарий для реализации данных процессов.</p> <p>Знание понятий, определений, описаний, формулировок. Понятие процесса создания ПО. Знание основных особенностей процессов разработки ПО. Умения в решении задач. Умение выбрать процесс для реализации задачи разработки ПО.</p>
<b>13. Проблемы, связанные с проектированием ПО.</b>	<p>Проблемы, которые можно решить только полным или частичным решением. Уровни детализации. Ограничение возможностей.</p> <p>Проектирование как эвристический процесс. Проектирование как постепенный процесс.</p> <p>Знание понятий, определений, описаний, формулировок. Понятие проектирования и уровня детализации. Умения в решении задач. Умение итеративно проектировать ПО.</p>



<b>14. Основные концепции проектирования. Компоненты проектирования. Эвристические принципы</b>	<p>Управление сложностью. Важность управления сложностью. Методы борьбы со сложностью. Желательные характеристики проекта. Уровни проектирования.</p> <p>Отображение объектов реального мира на абстракции.</p> <p>Инкапсуляция реализации. Практическое применение наследования. Определение областей вероятных изменений.</p> <p>Слабое сопряжение модулей. Популярные шаблоны проектирования. Применение эвристических принципов.</p>
<b>15. Методики проектирования</b>	<p>Итеративный процесс проектирования. Философия разделяй и властвуй. Нисходящий и восходящий подходы к проектированию.</p> <p>Экспериментальное прототипирование. Совместное проектирование. Регистрация процесса проектирования.</p> <p>Знание понятий, определений, описаний, формулировок. Понятие итеративного процесса проектирования. Понятие нисходящего и восходящего подхода к проектированию. Умения в решении задач.</p> <p>Умение использовать средства javadoc, wiki. Умение строить UML диаграммы. Умение использовать карточки CRC (class, responsibility, collaborator)</p>
<b>16. Популярные методологии</b>	<p>Крупномасштабное предварительное проектирование, Достаточное предварительное проектирование, Небольшое предварительное проектирование.</p> <p>Знание понятий, определений, описаний, формулировок. Понятие предварительного проектирования. Умения в решении задач.</p> <p>Умение выбрать степень детализации предварительного проектирования для выбранной задачи.</p>

## 5. Образовательные технологии

В преподавании курса «Конструирование программного обеспечения» используются традиционные технологии - лекции, - лабораторный практикум. Вместе с тем, нетрадиционным является: - лекционный материал подается с применением презентаций с использованием мультимедийного проектора; предполагается интерактивное обсуждение материала курса; - предусматривается опережающая самостоятельная работа студентов; - курс предусматривает выполнение и защиту курсовой работы.

## 6. Лабораторный практикум

Не предусмотрено

## 7. Практические занятия

№ раздела	Наименование практических занятий (семинаров)	Трудоемкость, ач
		Очная форма
1.	Выбор метафоры для разработки ПО в соответствии с заданием. Определение типа ПО. Обоснование выбора.	16
2.	Формирование архитектуры системы на уровне программной системы и ее подсистем.	14
Итого часов		30

## 8. Организация и учебно-методическое обеспечение самостоятельной работы

Самостоятельная работа студентов предполагает работу с лекционным материалом, с учебной литературой; опережающую самостоятельную работу; подготовку к лабораторным работам; подготовку к лабораторным работам; подготовку к экзамену. Курс предусматривает творческую самостоятельную работу студентов: выполнение курсовой работы; анализ научных публикаций по заданной теме.

Методы контроля самостоятельной работы студентов: контрольные работы, самоконтроль, контроль преподавателя.

## Примерное распределение времени самостоятельной работы студентов

Вид самостоятельной работы	Примерная трудоемкость, ач
	Очная форма
<b>Текущая СР</b>	
работа с лекционным материалом, с учебной литературой	4
опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	0
самостоятельное изучение разделов дисциплины	3
выполнение домашних заданий, домашних контрольных работ	0
подготовка к лабораторным работам, к практическим и семинарским занятиям	0
подготовка к контрольным работам, коллоквиумам	5
<b>Итого текущей СР:</b>	12
<b>Творческая проблемно-ориентированная СР</b>	
выполнение расчётно-графических работ	0
выполнение курсового проекта или курсовой работы	0
поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме	0
работа над междисциплинарным проектом	0
исследовательская работа, участие в конференциях, семинарах, олимпиадах	0
анализ данных по заданной теме, выполнение расчётов, составление схем и моделей на основе собранных данных	12
<b>Итого творческой СР:</b>	12
<b>Общая трудоемкость СР:</b>	24

## 9. Учебно-методическое обеспечение дисциплины

### 9.1. Адрес сайта курса

<https://dl.spbstu.ru/course/view.php?id=201>

## 9.2. Рекомендуемая литература

### Основная литература

№	Автор, название, место издания, издательство, год (годы) издания	Год изд.	Источник
1	Дробинцев П.Д. и др. Подход к тестированию параллельных систем на основе UCM-спецификаций // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. 2014. № 3 (198) URL: <a href="http://elibrary.spbstu.ru/dl/2/4685.pdf">http://elibrary.spbstu.ru/dl/2/4685.pdf</a>	2014	ЭБ СПбПУ

### Ресурсы Интернета

1. ресурсы интернета: <https://tproger.ru>

## 9.3. Технические средства обеспечения дисциплины

Свободно распространяемая версия Twiki и JavaDoc (или аналога для используемого в проекте языка программирования) для ведения документации. IDE для разработки программного продукта.

## 10. Материально-техническое обеспечение дисциплины

В аудиториях, где проводятся занятия, необходимо наличие досок и средств письма на них. Желательно наличие флип-чартов (2-3 штуки). Мебель должна позволять перестановку для организации работы студентов в группах

## 11. Критерии оценивания и оценочные средства

### 11.1. Критерии оценивания

Для дисциплины «Конструирование программного обеспечения» формой аттестации является зачёт. Дисциплина реализуется с применением системы индивидуальных достижений.

#### Текущий контроль успеваемости

Максимальное значение персонального суммарного результата обучения (ПСРО) по приведенной шкале - 100 баллов

Максимальное количество баллов приведенной шкалы по результатам прохождения двух точек контроля - 80 баллов.

Подробное описание правил проведения текущего контроля с указанием баллов по каждому контрольному мероприятию и критериев выставления оценки размещается в СДО в навигационном курсе дисциплины.

#### Промежуточная аттестация по дисциплине

Максимальное количество баллов по результатам проведения аттестационного испытания в период промежуточной аттестации – 20 баллов приведенной шкалы.

Промежуточная аттестация по дисциплине проводится в соответствии с расписанием.

Зачет получают студенты, успешно ответившие на вопросы к зачету, продемонстрировав знание основных понятий теории верификации и возможностей проведения верификации с использованием формальных методов и подходов.

Результаты промежуточной аттестации, определяются на основе баллов, набранных в рамках применения, СИД

Баллы по приведенной шкале в рамках применения СИД (ПСРО+ ПА)	Оценка по результатам промежуточной аттестации
	Экзамен/диф.зачет/зачет
0 - 60 баллов	Неудовлетворительно/не зачтено
61 - 75 баллов	Удовлетворительно/зачтено
76 - 89 баллов	Хорошо/зачтено
90 и более	Отлично/зачтено

## **11.2. Оценочные средства**

Оценочные средства по дисциплине представлены в фонде оценочных средств, который является неотъемлемой частью основной образовательной программы и размещается в электронной информационно-образовательной среде СПбПУ на портале etk.spbstu.ru

## **12. Методические рекомендации по организации изучения дисциплины**

Успешное изучение курса требует посещения лекций, практических занятий, выполнения всех учебных заданий преподавателя, ознакомления с основной и дополнительной литературой.

Во время лекции студент должен вести краткий конспект. Если какие-то материалы конспекта вызывают затруднения, необходимо постараться найти ответы на затруднительные вопросы, используя рекомендуемую литературу. Если студенту самостоятельно не удалось разобраться в материале, необходимо сформулировать вопросы и обратиться за помощью к преподавателю на консультации или ближайшей лекции. Обучающемуся необходимо регулярно отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.

При подготовке к экзамену в дополнение к изучению конспектов лекций и учебных пособий, необходимо пользоваться учебной литературой, рекомендованной в настоящей программе. При подготовке к экзамену нужно изучить теорию: определения всех понятий и подходы к оцениванию до состояния понимания материала.

## **13. Адаптация рабочей программы для лиц с ОВЗ**

Адаптированная программа разрабатывается при наличии заявления со стороны обучающегося (родителей, законных представителей) и медицинских показаний (рекомендациями психолого-медико-педагогической комиссии). Для инвалидов адаптированная образовательная программа разрабатывается в соответствии с индивидуальной программой реабилитации.