

# МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учебно-методическое объединение по образованию  
в области информатики и радиоэлектроники

## УТВЕРЖДАЮ

Первый заместитель Министра образования  
Республики Беларусь

\_\_\_\_\_ И.А. Старовойтова

\_\_\_\_\_

Регистрационный № ТД-\_\_\_\_\_/тип.

## ОСНОВЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ

Типовая учебная программа по учебной дисциплине  
для специальности

**1-40 01 01 Программное обеспечение информационных технологий**

### СОГЛАСОВАНО

Начальник Управления  
электроники и приборостроения,  
электротехнической и оптико-  
механической промышленности  
Министерства промышленности  
Республики Беларусь

\_\_\_\_\_ А.С. Турцевич

\_\_\_\_\_

### СОГЛАСОВАНО

Председатель Учебно-методического  
объединения по образованию  
в области информатики  
и радиоэлектроники

\_\_\_\_\_ В.А. Богуш

\_\_\_\_\_

### СОГЛАСОВАНО

Начальник Главного управления  
профессионального образования  
Министерства образования  
Республики Беларусь

\_\_\_\_\_ С.А. Касперович

\_\_\_\_\_

### СОГЛАСОВАНО

Проректор по научно-методической  
работе Государственного учреждения  
образования «Республиканский  
институт высшей школы»

\_\_\_\_\_ И.В. Титович

\_\_\_\_\_

Эксперт-нормоконтролер

\_\_\_\_\_

\_\_\_\_\_

Минск 2021

**СОСТАВИТЕЛИ:**

А.И. Парамонов, доцент кафедры программного обеспечения информационных технологий учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат технических наук, доцент;  
Н.В. Лапицкая, заведующий кафедрой программного обеспечения информационных технологий учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат технических наук, доцент.

**РЕЦЕНЗЕНТЫ:**

Кафедра технологий программирования Белорусского государственного университета (протокол № 13 от 15.04.2021 г.);  
Кафедра программной инженерии учреждения образования «Белорусский государственный технологический университет» (протокол № 8 от 05.05.2021 г.).

**РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ В КАЧЕСТВЕ ТИПОВОЙ:**

Кафедрой программного обеспечения информационных технологий учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № 15 от 17.05.2021 г.);  
Научно-методическим советом учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № 11 от 18.06.2021 г.);  
Научно-методическим советом по разработке программного обеспечения и информационно-коммуникационным технологиям Учебно-методического объединения по образованию в области информатики и радиоэлектроники (протокол № 6 от 18.05.2021 г.).

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

### ХАРАКТЕРИСТИКА УЧЕБНОЙ ДИСЦИПЛИНЫ

Типовая учебная программа по учебной дисциплине «Основы программной инженерии» разработана для студентов учреждений высшего образования, обучающихся по специальности 1-40 01 01 «Программное обеспечение информационных технологий» в соответствии с требованиями образовательного стандарта высшего образования первой ступени и типового учебного плана вышеуказанной специальности.

Настоящая программа устанавливает минимальные требования к знаниям и умениям студентов специальности 1-40 01 01 «Программное обеспечение информационных технологий» и определяет содержание и виды учебных занятий по учебной дисциплине «Основы программной инженерии». Построение содержания учебной дисциплины отвечает требованиям отечественных профессиональных стандартов в области информационных технологий и международного профессионального стандарта Guide to the Software Engineering Body of Knowledge ISO/IEC TR 19759 IEEE.

Подготовка современного специалиста в области программных и информационных технологий требует владения возможностями, предоставляемыми компьютерными технологиями. Учебная дисциплина «Основы программной инженерии» является одной из базовых учебных дисциплин фундаментальной подготовки в области программной инженерии, усвоение которой необходимо для профессионального становления студентов как инженеров-программистов. Учебная дисциплина актуализирует базовые знания по специальности, воспитывает у будущих специалистов стремление к расширению профессионального кругозора, формирует потребность к разносторонней оценке применяемых решений, показывает важность анализа проблемы и предметной области при создании программных продуктов.

### ЦЕЛЬ, ЗАДАЧИ, РОЛЬ УЧЕБНОЙ ДИСЦИПЛИНЫ

Цель учебной дисциплины: формирование студентами систематизированного представления о современных подходах к конструированию программного обеспечения, методах программной инженерии и ее стандартах, о процессах создания и эволюции сложных, многоверсионных, тиражируемых программных продуктов высокого качества, необходимого для практического использования на последующих этапах обучения и в профессиональной сфере деятельности будущего специалиста.

Задачи учебной дисциплины:

изучение современных методов создания качественного программного обеспечения, удовлетворяющего заданным требованиям;

освоение методов программной инженерии на всех этапах жизненного цикла программного обеспечения;

приобретение навыков работы с инструментальными средствами при создании программных продуктов;  
 формирование навыков разработки проектной документации.

Базовыми учебными дисциплинами по курсу «Основы программной инженерии» являются: «Информатика» (в объеме уровня общего среднего образования), «Математика» (в объеме уровня общего среднего образования), «Основы алгоритмизации и программирования» (разделы, изучаемые студентами в первом семестре). В свою очередь, учебная дисциплина «Основы программной инженерии» является базой для таких учебных дисциплин, как «Конструирование программного обеспечения», «Объектно-ориентированные технологии программирования и стандарты проектирования», «Разработка и анализ требований», «Стандартизация программного обеспечения» (компонент учреждения высшего образования), а также при выполнении курсовых и дипломных работ обучающихся, связанных с созданием программных продуктов.

### ТРЕБОВАНИЯ К УРОВНЮ ОСВОЕНИЯ СОДЕРЖАНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

В результате изучения учебной дисциплины «Основы программной инженерии» формируются компетенции:

*базовые профессиональные:*

анализировать подходы и стандарты, используемые в регламентированных процессах создания сложных, тиражируемых программных продуктов, соответствующих формальным требованиям заказчика.

В результате изучения учебной дисциплины обучающийся должен:

*знать:*

области SEEK, связанные с задачами, методами и стандартами программной инженерии;

современные модели, ключевые концепции и технологии разработки программных систем;

подходы к инженерному проектированию в конкретных предметных областях;

*уметь:*

формулировать и согласовывать с заказчиком требования к программному продукту;

использовать инструментальные средства для планирования и разработки программного продукта;

оформлять документацию для представления проекта;

*владеть:*

навыками разработки проектной (программной) документации в соответствии с ГОСТ ЕСПД;

навыками оценки времени и трудоемкости реализации требований к программному продукту;

навыками персональной и командной разработки;  
нормами оформления исходного программного кода в соответствии с установленными требованиями;  
умениями самостоятельного анализа новых тенденций в программной инженерии.

Программа рассчитана на 108 учебных часов, из них – 64 аудиторных.

Примерное распределение аудиторных часов по видам занятий:  
лекций – 30 часов, практических занятий – 34 часа.

Программа разработана без учета часов, отводимых на проведение текущей аттестации, определенной типовым учебным планом.

## ПРИМЕРНЫЙ ТЕМАТИЧЕСКИЙ ПЛАН

Наименование раздела, темы	Всего аудиторных, часы	Лекции, часы	Практические занятия, часы
Введение. Общие сведения о программной инженерии	2	2	-
<b>Раздел 1. Базовые понятия программной инженерии</b>	<b>14</b>	<b>8</b>	<b>6</b>
Тема 1. История развития программной инженерии как отрасли	2	2	-
Тема 2. Программы, программное обеспечение, программные продукты	2	2	-
Тема 3. IT-проекты. Технологии программирования	8	2	6
Тема 4. Стандарты программной инженерии. Проект SWEBOOK	2	2	-
<b>Раздел 2. Организация технологического процесса разработки программного продукта</b>	<b>24</b>	<b>12</b>	<b>12</b>
Тема 5. Этапы создания программного продукта. Жизненный цикл программного продукта	2	2	-
Тема 6. Методологии разработки программного продукта	4	2	2
Тема 7. Организация и планирование проекта	8	2	6
Тема 8. Документооборот в жизненном цикле программного продукта	2	2	-
Тема 9. Общие сведения о требованиях. Синтез и анализ требований	8	4	4
<b>Раздел 3. Технологии конструирования программных продуктов</b>	<b>24</b>	<b>8</b>	<b>16</b>
Тема 10. Техническое задание на разработку программного продукта	6	2	4
Тема 11. Этапы конструирования. Подходы к конструированию программных средств	2	2	-
Тема 12. Инструменты конструирования программных продуктов	8	2	6
Тема 13. Контроль версий	8	2	6
<b>Итого:</b>	<b>64</b>	<b>30</b>	<b>34</b>

## **СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА**

### **ВВЕДЕНИЕ. ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОЙ ИНЖЕНЕРИИ**

Понятие технологического процесса. Понятия инженера и инженерии. Сущность инженерных технологий. Терминология IT-индустрии.

Понятие программной инженерии. Фазы развития программной инженерии: от ремесла к науке. Инженерная дисциплина. Характеристика дисциплин программной инженерии. Сущность системной и компьютерной инженерии. Отличие от информатики и других инженерий.

### **Раздел 1. БАЗОВЫЕ ПОНЯТИЯ ПРОГРАММНОЙ ИНЖЕНЕРИИ**

#### **Тема 1. ИСТОРИЯ РАЗВИТИЯ ПРОГРАММНОЙ ИНЖЕНЕРИИ КАК ОТРАСЛИ**

Технические средства реализации информационных процессов. Базовые принципы программного управления и построения компьютеров. История развития информационных технологий. Становление программной инженерии: от теории вычислительных процессов до суперкомпьютеров.

#### **Тема 2. ПРОГРАММЫ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, ПРОГРАММНЫЕ ПРОДУКТЫ**

Алгоритмы и их программная реализация. Свойства хорошей программы. Понятие программного обеспечения. Классификация программного обеспечения. Программное обеспечение как продукт. Основные характеристики программных продуктов (ПП).

#### **Тема 3. IT-ПРОЕКТЫ. ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ**

Рынок программного обеспечения. Составляющие IT-проекта. Критерии успешности и причины неудач IT-проектов. Понятие технологии программирования. Обзор технологий программирования.

#### **Тема 4. СТАНДАРТЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ. ПРОЕКТ SWEBOOK**

Понятие стандарта. Виды стандартов. Основные стандарты программной инженерии. Современное состояние международных стандартов программной инженерии. Образовательный стандарт программной инженерии SWEBOOK. Структура и содержание SWEBOOK. Профессиональные и этические требования. Кодекс этики IEEE-CS/ACM.

### **Раздел 2. ОРГАНИЗАЦИЯ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА РАЗРАБОТКИ ПРОГРАММНОГО ПРОДУКТА**

#### **Тема 5. ЭТАПЫ СОЗДАНИЯ ПРОГРАММНОГО ПРОДУКТА. ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММНОГО ПРОДУКТА**

Процесс создания программного обеспечения. Основные стадии типового процесса создания программного обеспечения. Участники процесса создания

программного обеспечения. Понятие жизненного цикла (ЖЦ). Процессы ЖЦ. Этапы ЖЦ программного продукта.

#### Тема 6. МЕТОДОЛОГИИ РАЗРАБОТКИ ПРОГРАММНОГО ПРОДУКТА

Модели организации ЖЦ ПП. Методологии разработки ПП. Основные характеристики методологий. Современные подходы в разработке ПП. Принципы выбора модели ЖЦ для проекта.

#### Тема 7. ОРГАНИЗАЦИЯ И ПЛАНИРОВАНИЕ ПРОЕКТА

Основные организационные задачи при планировании проекта. Инструменты управления проектами. Временное планирование проекта. Диаграмма Ганта. Дисциплина управления в программной инженерии. Общие сведения о Project Management Body of Knowledge (PMBOK).

#### Тема 8. ДОКУМЕНТООБОРОТ В ЖИЗНЕННОМ ЦИКЛЕ ПРОГРАММНОГО ПРОДУКТА

Документация программного проекта. Цели и задачи документирования. Типы и виды программной документации. Документация на разных фазах развития проекта, связь с этапами жизненного цикла.

Нормативная база в области документирования программ. Обзор стандартов документирования. Организация документооборота в проекте. Инструменты управления программной документацией.

#### Тема 9. ОБЩИЕ СВЕДЕНИЯ О ТРЕБОВАНИЯХ. СИНТЕЗ И АНАЛИЗ ТРЕБОВАНИЙ

Понятие требований к программному обеспечению. Свойства требований. Классификация требований. Функциональные и нефункциональные требования. Проблемы определения требований. Процесс сбора требований к программному обеспечению. Методы сбора требований. Разработка требований. Анализ требований. Показатели качества требований к программному обеспечению. Факторы успешности реализации программного продукта. Типовой план управления требованиями. Организация требований к программному обеспечению. Задачи и методы контроля за требованиями.

#### Раздел 3. ТЕХНОЛОГИИ КОНСТРУИРОВАНИЯ ПРОГРАММНЫХ ПРОДУКТОВ

#### Тема 10. ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ ПРОГРАММНОГО ПРОДУКТА

Функциональная спецификация программного продукта. Типовое техническое задание (ТЗ) на разработку программы или программного обеспечения. Требования к содержанию и оформлению ТЗ. Спецификация требований к программному обеспечению (Software Requirements Specification, SRS).

Единая система программной документации.



## Тема 11. ЭТАПЫ КОНСТРУИРОВАНИЯ. ПОДХОДЫ К КОНСТРУИРОВАНИЮ ПРОГРАММНЫХ СРЕДСТВ

Фазы конструирования программных продуктов. Описание этапов конструирования. Связь конструирования с прочими стадиями жизненного цикла.

Стандарты в конструировании программного обеспечения. Стратегии конструирования программного обеспечения. Нисходящая и восходящая разработка. Практики использования типовых решений. Понятие «реинжиниринг». Точка зрения и характеристики точек зрения.

## Тема 12. ИНСТРУМЕНТЫ КОНСТРУИРОВАНИЯ ПРОГРАММНЫХ ПРОДУКТОВ

Компиляторы и интерпретаторы, генераторы кода. Языки программирования и платформы разработки. Отладчики. Инструменты тестирования программного кода. Интегрированная программная среда (Integrated Development Environment, IDE). Возможности, недостатки и преимущества использования IDE. Инструменты инспектирования и анализа кода.

## Тема 13. КОНТРОЛЬ ВЕРСИЙ

Понятие конфигурационного управления. Управление версиями. Понятие «ветки» проекта. Управление сборками. Средства версионного контроля. Понятие «baseline». Системы контроля версий (СКВ). Классификация СКВ. Обзор современных систем контроля версий. Работа над проектом в команде.

## ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

### ЛИТЕРАТУРА

#### ОСНОВНАЯ

1. Липаев, В. В. Программная инженерия. Методологические основы / В. В. Липаев. – М. : ТЕИС, 2006. – 608 с.
2. Липаев, В. В. Процессы и стандарты жизненного цикла сложных программных средств : справочник / В. В. Липаев. – М. : Синтег, 2006. – 276 с.
3. Мацяшек, Л. А. Практическая программная инженерия на основе учебного примера / Л. А. Мацяшек. – М. : БИНОМ, 2009. – 956 с.
4. Вигерс, К. И. Разработка требований к программному обеспечению / К. И. Вигерс. – М. : Русская редакция, 2004. – 576 с.

#### ДОПОЛНИТЕЛЬНАЯ

5. Батоврин, В. К. Толковый словарь по системной и программной инженерии / В. К. Батоврин. – М. : ДМК Пресс, 2012. – 280 с.
6. Геци, К. Основы инженерии программного обеспечения / К. Геци, М. Джазайери, Д. Мандртоли. – 2-е изд. – СПб. : БХВ-Петербург, 2005. – 832 с.
7. Sommerville, Ian. Software Engineering / Ian. Sommerville. – 9th Edition, Addison-Wesley, 2011. – 773 p.
8. Брауде, Э. Технология разработки программного обеспечения / Э. Брауде. – СПб. : Питер, 2004. – 655 с. : ил.
9. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул. – М. : Форум, ИНФРА-М, 2008. – 400 с. : ил.
10. Фатрелл, Р. Т. Управление программными проектами. Достижение оптимального качества при минимуме затрат / Р. Т. Фатрелл, Д. Ф. Шафер, Л. И. Шафер. – М. : Вильямс, 2004. – 1136 с.
11. Кагарлицкий, Ю. В. Разработка документации пользователя программного продукта. Методика и стиль изложения / Ю. В. Кагарлицкий. – 2-е изд. – М. : Философт, 2012. – 232 с.
12. Единая система программной документации. ИПК Издательство стандартов, 2001. – 164 с.
13. Леффингуэлл Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход / Д. Леффингуэлл, Д. Уидриг ; пер. с англ. – М. : Вильямс, 2002. – 448 с.
14. Орам Энди. Идеальная разработка ПО. Рецепты лучших программистов / Энди Орам, Грегори Уилсон. – СПб. : Питер, 2012. – 592 с.

#### ЭЛЕКТРОННЫЕ РЕСУРСЫ

15. About Git [Электронный ресурс]. – Режим доступа: <http://git-scm.com/about/>. – Дата доступа: 23.04.2021

16. Эли М Доу, Управление исходным кодом с помощью Git [Электронный ресурс]. – Режим доступа: <http://www.ibm.com/developerworks/ru/library/l-git/index.html>. – Дата доступа: 23.04.2021

17. About TortoiseSVN [Электронный ресурс]. – Режим доступа: <https://tortoisesvn.net/about.html>. – Дата доступа: 23.04.2021

18. CVS. Система Управления Параллельными Версиями [Электронный ресурс]. – Режим доступа: <http://gnu-doc-ru.sourceforge.net/cvs/index.html>. – Дата доступа: 23.04.2021

19. Atlassian. JIRA Software [Электронный ресурс]. – Режим доступа: <https://www.atlassian.com/software/jira/>. – Дата доступа: 23.04.2021

20. Redmine [Электронный ресурс]. – Режим доступа: <http://www.redmine.org/>. – Дата доступа: 23.04.2021

21. About Bug-Tracking System Bugzilla [Электронный ресурс]. – Режим доступа: <https://www.bugzilla.org/about/>. – Дата доступа: 23.04.2021

## МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

При изучении учебной дисциплины рекомендуется использовать следующие формы самостоятельной работы студентов:

контролируемая самостоятельная работа в виде решения индивидуальных задач в аудитории во время проведения практических занятий под контролем преподавателя;

выполнение индивидуальных заданий с возможностью самопроверки;  
командная домашняя работа по разноуровневым заданиям.

## ПЕРЕЧЕНЬ РЕКОМЕНДУЕМЫХ СРЕДСТВ ДИАГНОСТИКИ КОМПЕТЕНЦИЙ СТУДЕНТА

Типовым учебным планом по специальности 1-40 01 01 «Программное обеспечение информационных технологий» в качестве формы текущей аттестации по учебной дисциплине «Основы программной инженерии» рекомендуется экзамен.

Оценка учебных достижений студента производится по десятибалльной шкале.

Для промежуточного контроля по учебной дисциплине и диагностики компетенций студентов могут использоваться следующие формы:

отчеты по аудиторным практическим упражнениям с устной защитой;  
отчеты по домашним практическим упражнениям с устной защитой;  
контрольные работы;  
устные опросы на лекции;  
электронные тесты по разделам;  
доклады на практических занятиях и конференциях.

## РЕКОМЕНДУЕМЫЕ МЕТОДЫ (ТЕХНОЛОГИИ) ОБУЧЕНИЯ

Основные рекомендуемые методы (технологии) обучения, отвечающие целям и задачам учебной дисциплины:

проблемное обучения (проблемное изложение, вариативное изложение, частично-поисковый метод), реализуемое на лекционных занятиях;

учебно-исследовательская деятельность, творческий подход, реализуемые на практических занятиях;

проектные технологии, используемые при проектировании конкретного объекта, реализуемые при выполнении самостоятельной работы.

## ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

1. Отличительные черты программного обеспечения.
2. Классификация программных средств. Особенности подходов к разработке различных видов программных средств.
3. Сравнительный анализ информационных систем.
4. Программная инженерия в жизненном цикле программных средств.
5. Обзор методологий проектирования программных продуктов.
6. Выбор методологии и модели разработки программного средства.
7. Организация работы над проектом.
8. Инструментарий менеджера проекта.
9. Анализ предметной области с выделением процессов и объектов для автоматизации.
10. Формирование функциональных и технологических требований к программному средству.
11. Управление требованиями к программному обеспечению.
12. Документирование программного обеспечения. Формирование технического задания на разработку программного средства.
13. Современные инструментальные средства конструирования программных средств.
14. Системы контроля версий. Практические навыки организации версииности проектов.
15. Организация работы в команде.

## ПРИМЕРНЫЙ ПЕРЕЧЕНЬ КОМПЬЮТЕРНЫХ ПРОГРАММ

*(необходимого оборудования, наглядных пособий и т. п.)*

1. Пакет офисных приложений, включающий текстовый и табличный процессоры (например: Microsoft Office).
2. Клиентское приложение или терминал СКВ (например: Git, SVN)
3. Редактор блок-схем (например: Microsoft Visio, Dia, Flowchart и пр.).
4. Пакет для планирования проектов (например: Microsoft Project).
5. Современная среда разработки (например: Microsoft Visual Studio).