Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет радиотехники и электроники

Кафедра информационных радиотехнологий

Дисциплина: Методика преподавания общетехнических

и специальных дисциплин

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

на тему

**методическая разработка раздела "Организация   
тестирования программного обеспечения" по   
дисциплине "Тестирование и отладка программного обеспечения" уровня среднего специального   
образования**

БГУИР КР 1-08 01 01-07 13 ПЗ

Студент: гр. 844691 Новосёлов Н.А.

Руководитель: кандидат педагогических наук, доцент Парафиянович Т.А.

Минск 2020

**Реферат**

методическая разработка раздела «Организация тестирования программного обеспечения» по дисциплине «Тестирование и отладка программного обеспечения» уровня среднего специального образования: курсовая работа / В.А.Сабило. – Минск: БГУИР, 2020, п.з. – 89 с., CD-R – 1 шт.

В настоящем времени актуален вопрос о методике преподавания общетехнических и специальных дисциплин. Образование нуждается в улучшении существующих методик, а также в создании новых. Это способствует улучшению качества и процесса обучения.

Объектом исследования является образовательный процесс уровня среднего специального образования, предметом исследования — методика преподавания дисциплины «Тестирование и отладка программного обеспечения».

Цель исследования — обоснование и разработка элементов содержания обучения и методики преподавания по дисциплине «Тестирование и отладка программного обеспечения» уровня среднего специального образования.

Перед курсовой работой были поставлены следующие задачи:

1. Выявить и описать основные характеристики образовательного процесса уровня среднего специального образования и используемой дидактической системы;

2. Выявить на основе анализа типовой учебной программы, частных методик, спрогнозировать и описать особенности методики преподавания дисциплины «Тестирование и отладка программного обеспечения» уровня среднего специального образования;

3. разработать и описать элементы содержания обучения и методики преподавания раздела «Организация тестирования программного обеспечения» по дисциплине «Тестирование и отладка программного обеспечения».

Методы исследования: теоретический анализ, моделирование, проектирование, индукция и дедукция, составление библиографии, конспектирование, аннотирование и цитирование, систематизация и обобщение, изучение педагогической документации.

Практическая значимость работы заключается в возможности использовать наработки в дальнейшей преподавательской деятельности педагогами.

Социальная значимость исследования состоит в ознакомлении обучающихся с особенностями организации тестирования программного обеспечения.

Экономическая значимость заключается в подготовке квалифицированных кадров для дальнейшего представления их на рынке труда как высококвалифицированных специалистов, владеющих навыками тестирования программного обеспечения.

Учреждение образования

«Белорусский государственный университет информатики

и радиоэлектроники»

Факультет радиотехники и электроники

УТВЕРЖДАЮ

Заведующий кафедрой ИРТ

*–––––––––––––*Н.И.Листопад

*24.02.2020* –

ЗАДАНИЕ

по курсовому проектированию № КР 1-08 01 01-07 19 ПЗ

Студенту    *Новосёлову Никите Александровичу* *–––––––––––\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

1. Тема *Методическая разработка раздела «Организация тестирования программного обеспечения» по дисциплине «Тестирование и отладка программного обеспечения» уровня среднего специального образования––––––––*

2. Срок сдачи студентом законченного проекта–*26.05.2020* *–––––––––––––––   \_*

3. Исходные данные к проекту *1) Кодекс Республики Беларусь об образовании от 13.01.2011 № 243-З (с изм. и доп.) // Национальный реестр правовых актов Республики Беларусь. – 2011. – № 8/24152; 2) Молчан, Л.В., Славинская, О.В. Методика преподавания общетехнических и специальных дисциплин: электронный ресурс по учебной дисциплине направления специальности 1-08 01 01-07 «Профессиональное обучение (информатика)» / Л.В. Молчан. – [Электронный ресурс] – Минск: БГУИР, 2017; 3) Тестирование и отладка программного обеспечения: типовая учебная программа по учебной дисциплине профессионального компонента типового учебного плана по специальности 2‑40 01 01 «Программное обеспечение информационных технологий»: [утв. постановлением Министерства образования Республики Беларусь от 26.01.2017 №6].–––––––––––––––––––––––––––––––––––––––––––\_–––––––––––––––––––––––––––––––––––––––––– ––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––– ––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––– ––––––––––––––––––––––––––––––––––––––––––––––––*

4. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат разработке) *1)–Титульный лист; 2) Реферат; 3) Задание по курсовой работе; 4) Содержание; 5) Введение; 6) Глава 1. Дидактические основы методики преподавания раздела «Организация тестирования программного обеспечения» дисциплины «Тестирование и отладка программного обеспечения» уровня среднего специального образования, 1.1. Организация и проведение обучения на уровне среднего специального образования в учреждениях образования Республики Беларусь, 1.2. Содержание и особенности методики преподавания дисциплины «Организация тестирования программного обеспечения» специальности 2-40 01 01 «Программное обеспечение информационных технологий»; 7) Глава 2. Разработка содержания и элементов методики преподавания дисциплины «Тестирование и отладка программного обеспечения» по разделу «Организация тестирования программного обеспечения», 2.1. Фрагмент учебной программы дисциплины «Тестирование и отладка программного обеспечения» по разделу «Организация тестирования программного обеспечения», 2.2. Фрагмент календарно-тематического плана дисциплины, 2.3. Разработка содержания занятия по разделу программы; 8) Заключение; 9) Список использованных источников; 10) Приложения; 11) Ведомость документов*

5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков) *–нет–––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––– –––*

6. Консультант по проекту (с обозначением разделов проекта)   *доцент кафедры ИРТ, канд. пед. наук, доцент Парафиянович Т.А. –––––– ––––*

7. Дата выдачи задания –*24.02.2020* ––––*––––––––––––––––––––––––––– –––––––*

8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и трудоемкости отдельных этапов): *––––– –*

*8.1. Выбор темы, формулирование целей и задач исследования, формирование оглавления, оформление и подписание задания, изучение нормативных правовых актов, учебной, научной литературы по теме – до 09.03.2020 – (5 %)*

*8.2. Проект текста первого параграфа первой главы – до 20.03.2020 – (10 %)*

*8.3. Проект текста второго параграфа первой главы – до 31.03.2020 – (15 %)*

*8.4. Фрагмент учебной программы, проект текста первого параграфа второй главы – до 13.04.2020 – (15 %) .*

*8.5. Фрагмент календарно-тематического плана, проект текста второго параграфа второй главы – до 24.04.2020 – (15 %) . \_\_\_*

*8.6. Разработка и описание занятия, проект текста третьего параграфа второй главы – до 04.05.2020 – (30 %) . \_\_*

*8.7. Оформление и сдача работы на проверку – до 11.05.2020 – (10 %)\_\_\_\_\_\_\_\_*

*8.8. Защита курсовой работы с 20.05.2020 по 26.05.2020\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

РУКОВОДИТЕЛЬ*– –––––––– Т.А. Парафиянович*

Задание принял к исполнению *– 24.02.2020–––––\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(дата и подпись студента)

**Содержание**

|  |  |
| --- | --- |
| Введение | 6 |
| Глава 1. Дидактические основы методики преподавания раздела «Организация тестирования программного обеспечения» дисциплины «Тестирование и отладка программного обеспечения» уровня среднего специального образования | 8 |
| 1.1. Организация и проведение обучения на уровне среднего специального образования в учреждениях образования Республики Беларусь | 8 |
| 1.2. Содержание и особенности методики преподавания дисциплины «Организация тестирования программного обеспечения» специальности 2-40 01 01 «Программное обеспечение информационных технологий» | 17 |
| Глава 2. Разработка содержания и элементов методики преподавания дисциплины «Тестирование и отладка программного обеспечения» по разделу «Организация тестирования программного обеспечения» | 20 |
| 2.1. Фрагмент учебной программы дисциплины «Тестирование и отладка программного обеспечения» по разделу «Организация тестирования программного обеспечения» | 20 |
| 2.2. Фрагмент календарно-тематического плана дисциплины | 21 |
| 2.3. Разработка содержания занятия по разделу программы | 22 |
| Заключение | 26 |
| Список использованных источников | 27 |
| Приложения | 28 |
| Приложение А. Фрагмент перечня действующих типовых учебных планов | 28 |
| Приложение Б. Фрагмент перечня действующих типовых учебных программ | 30 |
| Приложение В. Фрагмент типовой учебной программы дисциплины | 33 |
| Приложение Г. Фрагмент учебной программы дисциплины | 37 |
| Приложение Д. Фрагмент календарно-тематического плана | 50 |
| Приложение Е. План-конспект лекционного занятия «Модульное тестирование» | 56 |
| Приложение Ж. План-конспект лекционного занятия «Интеграционное тестирование» | 62 |
| Приложение И. План-конспект лекционного занятия «Системное тестирование» | 67 |
| Приложение К. План-конспект лабораторного занятия «Разработка плана и проведение модульного тестирования ПО» | 71 |
| Приложение Л. Инструкция для лабораторного занятия | 75 |
| Приложение М. Презентации лекционных занятий | 82 |
| Ведомость документов | 83 |

**Введение**

Тема данной курсовой работы — «Методическая разработка раздела «Организация тестирования программного обеспечения» по дисциплине «Тестирование и отладка программного обеспечения» уровня среднего специального образования». Данная курсовая работа проектировочного характера и направлена на то, чтобы обобщить, систематизировать и закрепить знания, полученные при изучении предмета «Методика преподавания общетехнических и специальных дисциплин» и применить их на практике при разработке фрагмента курса по учебной дисциплине «Тестирование и отладка программного обеспечения».

Курсовая работа способствует формированию умений подбора и анализа литературных источников, использования теоретических знаний при решении практических задач. В подготовке будущего педагога курсовая работа играет большую роль. Она направлена на формирование системы методических знаний, умений и навыков, дающих возможность целенаправленно, осознанно и эффективно обучать, и воспитывать учащихся.

Для выполнения курсовой работы необходима система теоретических знаний об этапах обучения и содержании деятельности на каждом из этапов, т.е. умение анализировать учебно-программную документацию: учебный план, учебную программу, квалификационную характеристику. После анализа учебно-программной документации необходимо уметь делать верные выводы для дальнейшего применения их при разработке дидактического материала, написания календарно-тематического плана, плана-конспекта учебного занятия, правильного выбора методов, форм и средств обучения.

Цель работы: обоснование и разработка элементов содержания обучения и методики преподавания раздела «Организация тестирования программного обеспечения» по дисциплине «Тестирование и отладка программного обеспечения» уровня среднего специального образования.

Задачи:

* выявить и описать основные характеристики образовательного процесса уровня среднего специального образования и используемой дидактической системы;
* выявить на основе анализа типовой учебной программы, частных методик, спрогнозировать и описать особенности методики преподавания дисциплины «Тестирование и отладка программного обеспечения» уровня среднего специального образования;
* разработать и описать элементы содержания обучения и методики преподавания раздела «Организация тестирования программного обеспечения» дисциплины «Тестирование и отладка программного обеспечения».

Объектом является образовательный процесс уровня среднего специального образования, предметом является методика преподавания дисциплины «Тестирование и отладка программного обеспечения».

Методы исследования: теоретический анализ педагогической, научно-технической литературы, учебников и учебных пособий, квалификационных характеристик, моделирование; проектирование; обобщение; анализ и обобщение отечественной и зарубежной практики в области профессиональной педагогики.

Представленная работа выполнена самостоятельно на основе источников и литературы, список которых представлен в соответствующем разделе, не содержит плагиата. При выполнении работы использованы источники (в том числе действующие нормативы) 2010-2020 годов.

**Глава 1**

**Дидактические основы методики преподавания   
раздела «Организация тестирования программного обеспечения» дисциплины «Тестирование и отладка программного обеспечения» уровня среднего   
специального образования**

* 1. **Организация и проведение обучения на уровне среднего**

**специального образования в учреждениях образования**

**Республики Беларусь**

Согласно Кодексу Республики Беларусь об образовании мы проанали­зировали изучаемый вопрос и сделали выводы, представленные ниже [1].

Среднее специальное образование — уровень основного образования, направленный на развитие личности учащегося, курсанта, получение ими специальной теоретической и практической подготовки, завершающийся присвоением квалификации специалиста со средним специальным образова­нием, рабочего со средним специальным образованием.

Система среднего специального образования включает в себя:

* участников образовательного процесса при реализации образовательных программ среднего специального образования;
* образовательные программы среднего специального образования;
* учреждения среднего специального образования;
* иные учреждения образования, реализующие образовательные программы среднего специального образования;
* государственные организации образования, обеспечивающие функционирование системы среднего специального образования;
* учебно-методические объединения в сфере среднего специального образования;
* организации, обеспечивающие проведение практических занятий, прохождение практики учащимися, курсантами;
* организации — заказчики кадров;
* республиканские органы государственного управления, иные государственные организации, подчиненные Правительству Республики Беларусь, местные исполнительные и распорядительные органы, иные организации и физических лиц в пределах их полномочий в сфере среднего специального образования [1, c.150-151].

Участниками образовательного процесса при реализации образовательных программ среднего специального образования являются учащиеся, курсанты, законные представители несовершеннолетних учащихся, курсантов, педагогические работники.

Учащимся на весь период получения среднего специального образования выдаются билет учащегося и книжка успеваемости учащегося, образцы которых устанавливаются Министерством образования Республики Беларусь.

Курсантам на весь период получения среднего специального образования выдается книжка успеваемости учащегося, образец которой устанавливается Министерством образования Республики Беларусь.

Образовательные программы среднего специального образования подразделяются на:

* образовательную программу среднего специального образования, обеспечивающую получение квалификации специалиста со средним специальным образованием;
* образовательную программу среднего специального образования, обеспечивающую получение квалификации рабочего со средним специальным образованием;
* образовательную программу среднего специального образования, обеспечивающую получение квалификации специалиста со средним специальным образованием и интегрированную с образовательными программами профессионально-технического образования;
* образовательную программу среднего специального образования, обеспечивающую получение квалификации рабочего со средним специальным образованием и интегрированную с образовательными программами профессионально-технического образования [1, c.151].

Образовательная программа среднего специального образования, обеспечивающая получение квалификации специалиста со средним специальным образованием, предусматривает базовый или повышенный уровень изучения учебных дисциплин, прохождения практики.

При освоении содержания образовательных программ среднего специального образования на основе общего базового образования обеспечивается получение общего среднего образования.

При освоении содержания образовательной программы среднего специального образования, обеспечивающей получение квалификации специалиста со средним специальным образованием, в случаях, предусмотренных образовательными стандартами среднего специального образования, осуществляется присвоение квалификации рабочего.

Образовательные программы среднего специального образования реализуются в очной и заочной формах получения образования.

Образовательные программы среднего специального образования реализуются в учреждениях среднего специального образования, профессионально-технических колледжах, гимназиях — колледжах искусств, базовых школах — колледжах искусств, средних школах — колледжах искусств, средних школах — училищах олимпийского резерва, а также могут реализовываться в учреждениях высшего образования. Виды реализуемых образовательных программ среднего специального образования в указанных учреждениях образования [1].

Образовательные стандарты среднего специального образования разрабатываются по каждой специальности (направлению специальности) и устанавливают требования к содержанию профессиональной деятельности специалиста со средним специальным образованием, рабочего со средним специальным образованием, компетентности специалиста со средним специальным образованием, рабочего со средним специальным образованием, содержанию учебно-программной документации образовательных программ среднего специального образования, уровню основного образования лиц, поступающих для получения среднего специального образования, вступительным испытаниям, формам и срокам получения среднего специального образования, организации образовательного процесса, объему учебной нагрузки учащихся, курсантов, уровню подготовки выпускников, итоговой аттестации.

Разработку образовательных стандартов среднего специального образования организует Министерство образования Республики Беларусь и осуществляет ее совместно с учебно-методическими объединениями в сфере среднего специального образования, республиканскими органами государственного управления, иными государственными организациями, подчиненными Правительству Республики Беларусь, в соответствии с перечнем закрепленных за ними специальностей, утверждаемым Правительством Республики Беларусь.

Образовательные стандарты среднего специального образования утверждаются Министерством образования Республики Беларусь по согласованию с республиканскими органами государственного управления, иными государственными организациями, подчиненными Правительству Республики Беларусь, для которых осуществляется подготовка специалистов со средним специальным образованием, рабочих со средним специальным образованием.

Срок получения среднего специального образования в дневной форме получения образования составляет:

1. На основе общего базового образования — от трех до четырех лет;
2. На основе общего среднего образования — от двух до трех лет;
3. На основе профессионально-технического образования с общим средним образованием — от одного года до трех лет [1, c.152-153].

Срок получения среднего специального образования при освоении содержания образовательной программы среднего специального образования, обеспечивающей получение квалификации специалиста со средним специальным образованием и предусматривающей повышенный уровень изучения учебных дисциплин, прохождения практики, срок получе­ния среднего специального образования в вечерней или заочной форме получения образования определяются сроком получения среднего специального образования в дневной форме получения образования и увеличиваются не более чем на один год.

Срок получения среднего специального образования при освоении содержания образовательной программы среднего специального образования, обеспечивающей получение квалификации специалиста со средним специальным образованием, для учащихся средних школ – училищ олимпийского резерва по решению учредителя средней школы – училища олимпийского резерва может быть увеличен по причине участия в течение учебного года в составе национальных и сборных команд Республики Беларусь по видам спорта в спортивных мероприятиях не более чем на два года [1].

Согласно статье 191 Кодекса Республики Беларусь об образовании было выяснено следующее:

«Учреждение среднего специального образования — учреждение образования, которое реализует образовательные программы среднего специального образования, программу воспитания и защиты прав и законных интересов детей, находящихся в социально опасном положении, и может реализовывать образовательные программы профессионально-технического образования, образовательную программу дополнительного образования детей и молодежи, образовательные программы дополнительного образования взрослых, за исключением образовательной программы переподготовки руководящих работников и специалистов, имеющих высшее образование» [1].

Учреждением среднего специального образования является колледж.

Основным структурным подразделением учреждения среднего специального образования, осуществляющим образовательную деятельность, является отделение.

Главой 34 Кодекса Республики Беларусь об образовании была рассмотрена организация образовательного процесса при реализации образовательных программ среднего специального образования [1].

Образовательный процесс при реализации образовательных программ среднего специального образования организуется в учреждениях образования по учебным годам. Учебный год при реализации образовательных программ среднего специального образования делится на семестры, которые завершаются экзаменационными сессиями.

Для учащихся, курсантов устанавливаются:

* каникулы на протяжении учебного года продолжительностью не менее 2 календарных недель;
* летние каникулы продолжительностью не менее 6 календарных недель [1, c.157].

Основной формой организации образовательного процесса при реализации образовательных программ среднего специального образования является учебное занятие: урок, лекция, курсовое проектирование, семинар­ское, лабораторное, практическое и иное занятие. Учебные занятия могут проводиться по сменам. Дополнительно проводятся факультативные занятия, консультации.

Количество часов, отводимых на учебную и производственную практи­ку, составляет не менее 20 процентов от общего количества часов, отводимых на реализацию образовательных программ среднего специального образования.

Порядок организации учебной и производственной практики определяется Положением о практике учащихся, курсантов, осваивающих содержание образовательных программ среднего специального образования, утверждаемым Правительством Республики Беларусь.

Образовательный процесс при реализации образовательных программ среднего специального образования осуществляется в учебных группах или индивидуально.

Наполняемость учебной группы при получении среднего специального образования в дневной форме получения образования составляет от 25 до 30 учащихся, курсантов, в вечерней и заочной формах получения образования – от 15 до 20 учащихся, курсантов.

Наполняемость учебной группы, в которой образовательный процесс организован только для лиц с особенностями психофизического развития, составляет от 6 до 12 учащихся, курсантов.

Наполняемость учебной группы, в которой образовательный процесс одновременно организован для лиц с особенностями психофизического развития и иных лиц, составляет от 15 до 20 учащихся, курсантов, из них не более 6 учащихся, курсантов из числа лиц с особенностями психофизического развития.

Учредитель учреждения образования, реализующего образовательные программы среднего специального образования, может устанавливать меньшую наполняемость групп. При этом для учреждений образования, функционирование которых обеспечивается за счет средств республиканского бюджета, решение об уменьшении наполняемости групп принимается по согласованию с Министерством финансов Республики Беларусь.

В случаях, предусмотренных Положением об учреждении среднего специального образования, учебные группы могут делиться на подгруппы.

Образовательный процесс осуществляется индивидуально на основании решения руководителя учреждения образования, реализующего образовательные программы среднего специального образования, в соответствии с индивидуальным учебным планом или на основании типового учебного плана по специальности (направлению специальности) [1].

Главой 35 Кодекса Республики Беларусь об образовании была рассмотрена аттестация учащихся, курсантов при освоении содержания образовательных программ среднего специального образования [1].

Учащиеся, курсанты при освоении содержания образовательных про­грамм среднего специального образования проходят текущую аттестацию.

Формами текущей аттестации являются:

* опрос учащегося, курсанта;
* контрольная работа;
* курсовой проект (курсовая работа);
* зачет (дифференцированный зачет);
* экзамен по учебной дисциплине [1, c.158].

Результаты текущей аттестации учащихся, курсантов оцениваются отметками в баллах по десятибалльной шкале либо отметками «зачтено», «не зачтено». Положительными являются отметки не ниже 4 (четырех) баллов и «зачтено».

Порядок проведения текущей аттестации учащихся, курсантов при освоении содержания образовательных программ среднего специального образования определяется Правилами проведения аттестации учащихся, курсантов при освоении содержания образовательных программ среднего специального образования.

Учащиеся, курсанты при завершении освоения содержания образовательных программ среднего специального образования проходят итоговую аттестацию.

Итоговая аттестация осуществляется государственной квалификационной комиссией.

Итоговая аттестация проводится в одной из следующих форм:

* защита дипломного проекта;
* государственные экзамены по учебным дисциплинам;
* государственный экзамен по специальности (направлению специальности);
* государственный экзамен по специальности (направлению специальности) и защита дипломного проекта (работы) [1, c.158-159].

К итоговой аттестации допускаются учащиеся, курсанты, выполнившие учебные планы и учебные программы.

Результаты итоговой аттестации учащихся, курсантов оцениваются отметками в баллах по десятибалльной шкале. Положительными являются отметки не ниже 4 (четырех) баллов.

Учащимся, курсантам, не прошедшим итоговую аттестацию в установленный срок по уважительной причине, предоставляется право прохождения итоговой аттестации в другой срок во время работы государственной квалификационной комиссии.

Учащимся, курсантам, не прошедшим итоговую аттестацию в установленный срок без уважительной причины или получившим по ее результатам отметки ниже 4 (четырех) баллов, предоставляется право прохождения итоговой аттестации во время работы государственной квалификационной комиссии, но не ранее чем через десять месяцев.

Порядок проведения итоговой аттестации учащихся, курсантов при освоении содержания образовательных программ среднего специального образования определяется Правилами проведения аттестации учащихся, кур­сантов при освоении содержания образовательных программ среднего специ­ального образования [1].

Была рассмотрена глава 36 Кодекса Республики Беларусь об образовании, где описано научно-методическое обеспечение среднего специального образования [1].

Научно-методическое обеспечение среднего специального образования включает в себя:

* учебно-программную документацию образовательных программ среднего специального образования;
* программно-планирующую документацию воспитания;
* учебно-методическую документацию;
* учебные издания;
* информационно-аналитические материалы [1, с.159].

Научно-методическое обеспечение среднего специального образования осуществляют:

* организации, осуществляющие научно-методическое обеспечение среднего специального образования;
* учреждения среднего специального образования;
* иные учреждения образования, реализующие образовательные программы среднего специального образования;
* учебно-методические объединения в сфере среднего специального образования;
* организации – заказчики кадров;
* республиканские органы государственного управления, иные государственные организации, подчиненные Правительству Республики Беларусь, иные организации и физические лица в пределах их полномочий в сфере среднего специального образования [1, c.159].

Учебно-программная документация образовательных программ среднего специального образования включает в себя учебные планы и учебные программы.

Учебные планы подразделяются на:

* типовые учебные планы по специальностям (направлениям специальностей);
* типовые учебные планы по специализациям;
* учебные планы учреждений образования, реализующих образовательные программы среднего специального образования, по специальностям (направлениям специальностей) и специализациям;
* экспериментальные учебные планы учреждений образования, реализующих образовательные программы среднего специального образования, по специальностям (направлениям специальностей) и специализациям;
* индивидуальные учебные планы [1, с.160].

Типовой учебный план по специальности (направлению специальности) является техническим нормативным правовым актом, разрабатывается на основе образовательного стандарта среднего специально­го образования по специальности (направлению специальности) и устанавливает перечень компонентов, циклов, последовательность изучения учебных дисциплин, количество учебных часов, отводимых на их изучение, формы учебных занятий, виды и сроки прохождения практики, формы и сроки проведения аттестации, минимальное количество обязательных контрольных работ, экзаменов применительно к специальности (направлению специальности), а также перечень необходимых кабинетов, лабораторий, мастерских и иных учебных объектов.

Типовые учебные планы по специальностям (направлениям специальностей) разрабатываются организациями, осуществляющими научно-методическое обеспечение среднего специального образования, и утверждаются Министерством образования Республики Беларусь по согласованию с заинтересованными республиканскими органами государственного управления, иными государственными организациями, подчиненными Правительству Республики Беларусь, в соответствии с перечнем закрепленных за ними специальностей при наличии заключений органов и учреждений, осуществляющих государственный санитарный надзор, об их соответствии санитарно-эпидемиологическому законодательству.

Типовой учебный план по специализации является техническим нормативным правовым актом, разрабатывается как приложение к типовому учебному плану по специальности (направлению специальности), содержит цикл специализации и включает в себя перечень учебных дисциплин специализации, количество учебных часов, отводимых на их изучение, формы учебных занятий, виды и сроки прохождения практики, формы и сроки проведения аттестации по учебным дисциплинам специализации, а также перечень необходимых кабинетов, лабораторий, мастерских и иных учебных объектов.

Типовые учебные планы по специализации разрабатываются организациями, осуществляющими научно-методическое обеспечение среднего специального образования, и утверждаются республиканскими органами государственного управления, иными государственными организациями, подчиненными Правительству Республики Беларусь, в соответствии с перечнем закрепленных за ними специальностей при наличии заключений органов и учреждений, осуществляющих государственный санитарный надзор, об их соответствии санитарно-эпидемиологическому законодательству.

Учебный план учреждения образования, реализующего образовательные программы среднего специального образования, по специальностям (направлениям специальностей) и специализациям разрабатывается учреждением образования, реализующим образовательные программы среднего специального образования, на основе типового учебного плана по специальности (направлению специальности) и типового учебного плана по специализации с учетом уровня изучения учебных дисциплин, прохождения практики и утверждается руководителем этого учреждения об­разования.

Учебный план учреждения образования, реализующего образователь­ные программы среднего специального образования, предусматривающий повышенный уровень изучения учебных дисциплин, прохождения практики, утверждается руководителем этого учреждения образования по согласованию с его учредителем.

Экспериментальный учебный план учреждения образования, реализующего образовательные программы среднего специального образования, по специальности (направлению специальности) и специализации апробируется в учреждении образования, реализующем образовательные программы среднего специального образования, на базе которого осуществляется экспериментальная деятельность.

Экспериментальные учебные планы учреждений образования, реализующих образовательные программы среднего специального образования, по специальностям (направлениям специальностей) и специализациям разрабатываются учреждениями образования, реализующими образовательные программы среднего специального образования, на базе которых осуществляется экспериментальная деятельность, и утверждаются Министерством образования Республики Беларусь по согласованию с учредителями этих учреждений образования.

Индивидуальный учебный план устанавливает особенности получения среднего специального образования одаренными и талантливыми учащимися, курсантами, а также учащимися, курсантами, которые по уважительным причинам не могут постоянно или временно посещать учебные занятия и (или) проходить в установленные сроки аттестацию.

Индивидуальные учебные планы разрабатываются учреждениями образования, реализующими образовательные программы среднего специального образования, на основе учебных планов учреждений образования, реализующих образовательные программы среднего специального образования, по специальностям (направлениям специальностей) и специализациям и утверждаются их руководителями.

Учебные программы подразделяются на:

* типовые учебные программы по учебным дисциплинам;
* типовые учебные программы по практике;
* учебные программы учреждений образования, реализующих образовательные программы среднего специального образования, по учебным дисциплинам;
* учебные программы учреждений образования, реализующих образовательные программы среднего специального образования, по практике;
* экспериментальные учебные программы учреждений образования, реализующих образовательные программы среднего специального образования, по учебным дисциплинам;
* экспериментальные учебные программы учреждений образования, реализующих образовательные программы среднего специального образова­ния, по практике [1, c.161].

Типовая учебная программа по учебной дисциплине является техническим нормативным правовым актом и определяет содержание учебной дисциплины, время, отведенное на изучение отдельных тем, основные требования к результатам учебной деятельности учащихся, курсантов, рекомендуемые формы и методы обучения и воспитания, перечень учебных изданий и средств обучения.

Типовые учебные программы по учебным дисциплинам профессионального компонента разрабатываются организациями, осуществляющими научно-методическое обеспечение среднего специального образования, и утверждаются республиканскими органами государственного управления, иными государственными организациями, подчиненными Правительству Республики Беларусь, в соответствии с перечнем закрепленных за ними специальностей.

Типовые учебные программы по учебным дисциплинам общеобразовательного компонента разрабатываются и утверждаются Министерством образования Республики Беларусь.

Типовая учебная программа по практике является техническим нормативным правовым актом и определяет содержание, сроки и последовательность прохождения учащимися, курсантами практики, основные требования к условиям организации практики, результатам прохождения практики учащимися, курсантами, рекомендуемые формы и методы обучения и воспитания, перечень учебных изданий и средств обучения.

Типовые учебные программы по практике разрабатываются организациями, осуществляющими научно-методическое обеспечение среднего специального образования, и утверждаются республиканскими органами государственного управления, иными государственными организациями, подчиненными Правительству Республики Беларусь, в соответствии с перечнем закрепленных за ними специальностей.

Учебная программа учреждения образования, реализующего образовательные программы среднего специального образования, по учебной дисциплине разрабатывается учреждением образования, реализующим образовательные программы среднего специального образования, на основе типовой учебной программы по учебной дисциплине, предусматривает базовый или повышенный уровень изучения учебной дисциплины и утверждается его руководителем.

Учебная программа учреждения образования, реализующего образовательные программы среднего специального образования, по практике разрабатывается этим учреждением образования на основе типовой учебной программы по практике, предусматривает базовый или повышенный уровень прохождения практики и утверждается его руководителем.

Экспериментальная учебная программа учреждения образования, реа­лизующего образовательные программы среднего специального образования, по учебной дисциплине, экспериментальная учебная программа учреждения образования, реализующего образовательные программы среднего специального образования, по практике апробируются в учреждении образования, реализующем образовательные программы среднего специального образования, на базе которого осуществляется экспериментальная деятельность.

Экспериментальные учебные программы учреждений образования, реализующих образовательные программы среднего специального образования, по учебным дисциплинам, экспериментальные учебные программы учреждений образования, реализующих образовательные программы среднего специального образования, по практике разрабатываются учреждениями образования, реализующими образовательные программы среднего специального образования, на базе которых осуществляется экспериментальная деятельность, и утверждаются Министерством образования Республики Беларусь по согласованию с учредителями этих учреждений образования [1].

* 1. **Содержание и особенности методики преподавания** **дисциплины «Тестирование и отладка программного обеспечения» специальности**

**2-40 01 01 «Программное обеспечение информационных технологий»**

Содержание и особенности методики преподавания учебной дисциплины «Тестирование и отладка программного обеспечения» специальности 2-40 01 01 «Программное обеспечение информационных технологий» рассмотрены на основе типовой учебной программы, утвержденной постановлением Министерства образования Республики Беларусь от 26.01.2017 №6 [2].

Типовая учебная программа по учебной дисциплине «Тестирование и отладка программного обеспечения» (далее – программа) предусматривает изучение видов, назначения и применения методов тестирования программного обеспечения.

Цель преподавания учебной дисциплины – развитие профессиональной компетентности в области тестирования и отладки сложных программных средств.

Изучение программного учебного материала базируется на знаниях, полученных учащимися в ходе изучения таких учебных дисциплин, как «Основы алгоритмизации и программирование», «Конструирование программ и языки программирования», «Структуры и алгоритмы обработки данных», «Технология разработки программного обеспечения»

Для закрепления теоретического материала и формирования у учащихся необходимых умений и навыков программой предусматривается проведение лабораторных занятий. Форма проведения лабораторных занятий определяется преподавателем исходя из цели обучения и содержания учебного материала.

В целях контроля усвоения программного учебного материала предусматривается проведение одной обязательной контрольной работы (ОКР). Содержание и конкретные сроки проведения ОКР определяются преподавателем, обсуждаются на заседании предметной (цикловой) комиссии и утверждаются в установленном порядке.

Программой определены цели изучения каждой темы, спрогнозированы результаты их достижения в соответствии с уровнями усвоения учебного материала.

В результате изучения учебной дисциплины учащиеся должны:

1. Знать на уровне представления:

* виды, уровни, направления и методы тестирования;
* критерии выбора тестов и оценки качества программного обеспечения;
* понятие верификации программного обеспечения;
* особенности документирования дефектов с использованием систем отслеживания проблем;

1. Знать на уровне понимания:
   * значение основных терминов, используемых в области тестирования и отладка программного обеспечения;
   * особенности проведения модульного, системного и интеграционного тестирования;
   * требования к составлению отчётов об ошибках;
   * особенности тестирования веб-приложений;
   * основы тестирования безопасности, производительности, регрессионного тестирования;
   * особенности выполнения автоматизированного тестирования;
2. Уметь:
   * проводить тестирование структуры программных модулей и их взаимодействия;
   * выполнять разработку тестовых сценариев;
   * составлять отчёты об ошибках;
   * проводить отладку и функциональное тестирование веб-ориентированных приложений;
   * использовать инструментальные средства при проведении автоматизированного тестирования и отладки программного обеспечения [2, c.5].

В программе приведены примерные критерии оценки результатов учебной деятельности учащихся по учебной дисциплине, которые разработаны на основе десятибалльной шкалы и показателей оценки результатов учебной деятельности, обучающихся в учреждениях среднего специального образования.

Программа содержит примерный перечень оснащения лаборатории техническими и программными средствами, необходимыми для обеспечения образовательного процесса.

Приведенный в программе тематический план является рекомендательным. Предметная (цикловая) комиссия учреждения образования может вносить обоснованные изменения в содержание и последовательность изложения программного учебного материала, распределение учебных часов по темам в пределах общего бюджета времени, отведенного на изучение учебной дисциплины. Все изменения должны быть утверждены заместителем руководителя учреждения образования по учебной работе.

**Глава 2**

**Разработка содержания и элементов методики   
преподавания дисциплины «Тестирование и отладка программного обеспечения» по разделу «Организация тестирования программного обеспечения»**

**2.1. Фрагмент учебной программы дисциплины «Тестирование и   
отладка программного обеспечения» по разделу «Организация   
тестирования программного обеспечения»**

Данный фрагмент учебной программы был разработан на основе типовой учебной программы по дисциплине «Тестирование и отладка программного обеспечения» утвержденной Министерством Образования Республики Беларусь 26 января 2017 году. Типовая учебная программа по дисциплине «Тестирование и отладка программного обеспечения» представлена в перечне действующих типовых учебных программ специальности 2-40 01 01 «Программное обеспечение информационных технологий» [3]. Перечень программ специальности 2‑40 01 01 «Программное обеспечение информационных технологий» подтвержден типовым учебным планом данной специальности [4]. Фрагмент перечня действующих типовых учебных планов представлен в приложении А, а фрагмент перечня действующих типовых учебных программ представлен в приложении Б.

Типовая учебная программа по учебной дисциплине является техническим нормативным правовым актом и определяет содержание учебной дисциплины, время, отведенное на изучение отдельных тем, основные требования к результатам учебной деятельности учащихся, курсантов, рекомендуемые формы и методы обучения и воспитания, перечень учебных изданий и средств обучения.

Типовые учебные программы по учебным дисциплинам профессионального компонента разрабатываются организациями, осуществляющими научно-методическое обеспечение среднего специального образования, и утверждаются республиканскими органами государственного управления, иными государственными организациями, подчиненными Правительству Республики Беларусь, в соответствии с перечнем закрепленных за ними специальностей.

Учебная программа учреждения образования, реализующего образовательные программы среднего специального образования, по учебной дисциплине разрабатывается учреждением образования, реализующим образовательные программы среднего специального образования, на основе типовой учебной программы по учебной дисциплине, предусматривает базовый или повышенный уровень изучения учебной дисциплины и утверждается его руководителем.

Разработанный фрагмент учебной программы состоит из:

* примерно-тематического плана;
* пояснительной записки;
* содержания дисциплины;
* литературы.

В примерно-тематическом плане указывается распределение часов на изучение дисциплины по разделам и темам.

Утвержденная учебная программа дисциплины «Тестирование и отладка программного обеспечения» представлена в приложении В.

Разработанный фрагмент учебной программы дисциплины «Тестирование и отладка программного обеспечения» представлен в приложении Г.

Основная часть содержания программы представляет собой таблицу из трех основных столбцов, в которых указываются цели изучения тем, их со­держание и результат изучения. При составлении календарно-тематического плана и дальнейшего проведения занятий преподаватель должен руковод­ствоваться содержанием программы. То есть разработанные преподавателем учебные занятия, выбранные их организационные формы и методы обучения должны быть оптимальны, и направлены на достижения целей и результатов учебной программы.

**2.2. Фрагмент календарно-тематического плана дисциплины**

На титульном листе календарно-тематического плана предусмотреть гриф «УТВЕРЖДАЮ», прописать наименование учебной дисциплины, номер групп, указать форму обучения, фамилию, имя и отчество преподавателя, дату утверждения учебной программы, в соответствии с которой составлен календарно-тематический план, количество учебных часов по учебному плану, отведенных на изучение учебной дисциплины, в том числе на проведение обзорных, лабораторных, практических занятий, на курсовое проектирование.

Указать количество домашних контрольных работы по учебной дисциплине.

На первой странице в графе 1 записывается порядковый номер учебного занятия.

Графы 2, 3, 4, 5, 6 заполняются в соответствии с расписанием учебных занятий для каждой учебной группы.

В графу 7 «Наименование разделов, тем по учебной программе, тем отдельных учебных занятий» вписываются названия разделов, тем, отдельных учебных занятий программного материала, наиболее сложного, трудно усваиваемого, требующего конкретизации, объяснения и обобщения со стороны преподавателя, рассматриваемый на обзорных, лабораторных и практических занятиях. Содержательная и исполнительная часть лабораторных и практических занятий (работ) должна быть наполнена деятельностью учащихся, направленной на закрепление теоретических знаний на практике, на формирование профессиональных умений и навыков пользования установками, оборудованием, приборами и т.д.

В графе 8 указывается количество учебных часов, предусмотренных на проведение учебного занятия. Учебный материал по учебной дисциплине разбивается на темы (подтемы), рассчитанные, как правило, на 2 учебных часа.

На второй странице в графе 9 указывается тип учебного занятия.

В графе 10 указывается учебно-методическое обеспечение учебного занятия (средства обучения) в соответствии с выбранным методом учебно-познавательной деятельности.

В графу 11 пописываются задания для самостоятельной работы учащихся (наименование тем учебной программы учебной дисциплины, выдаваемые учащимся для самостоятельного изучения, графические расчеты и другие задания, а также задания для проведения практических и лабораторных работы), как в межсессионный период, так и в период лабораторно-экзаменационных сессий.

В графу 12 вносится основная и дополнительная литература, методические пособия, практикумы, конспекты лекций и электронный адрес, по которому учащиеся могут получить дополнительную информацию по изучению учебной дисциплины.

В графу 13 прописываются коррективы, внесенные с учетом достижений науки и техники, изменений, вносимых с содержание типовой учебной программы учебной дисциплины и др. Изменения в дополнения должны быть обсуждены на заседании предметной (цикловой) комиссии, утверждены заместителем директора по учебной работе.

На последней странице размещается раздел «Литература», который, как правило, представляется следующими подразделами: «Основная литература», «Дополнительная литература». В подразделы вносятся учебники и другие печатные текстовые средства, специальное учебно-методическое обеспечение, практикумы по лабораторным и практическим занятиям (работам), автоматизированные обучающие курсы, профессиональные прикладные программы электронные источники учебной информации и др.

Календарно-тематический план составляется на учебный год (семестр), рассматривается предметной (цикловой) комиссией и утверждается заместителем директора по учебной работе.

Разработанный фрагмент календарно-тематического плана представлен в приложении Д.

**2.3. Разработка содержания занятия по разделу программы**

В процессе разработки содержания занятия по разделу программы был изучен образовательный стандарт среднего специального образования по специальности 2-40 01 01 «Программное обеспечение информационных технологий», а также была рассмотрена разработка учебных программ допрофессиональной и профессиональной подготовки учащихся учреждений общего среднего и специального образования [5].

Согласно курсовому проекту необходимо разработать три лекционных занятия, а также одно лабораторное занятие в условиях лекционно-семинарской дидактической системы. В данной работе предложено содержа­ние лабораторной работы № 5 учебной программы «Разработка плана и проведение модульного тестирования ПО».

На данный момент традиционная педагогика предлагает несколько способов организации занятий по отработке навыков и два из них – лекционные и лабораторные занятия. На лекционных занятиях учащиеся приобретают новые знания, на лабораторных – развивают свою учебную и познавательную деятельность в специально созданных для этого условиях. Сначала рассмотрим лекционные занятия.

По характеру лекционное занятие относится к лекционно-семинарским формам обучения.

В начале занятия педагог сообщает тему и цели занятия, мотивирует учащихся на успешную учебную деятельность.

В конце занятия педагог обсуждает с учащимися результаты их теоретической деятельности, спрашивает, что было непонятно, что вызвало трудности, при необходимости отвечает на вопросы, возникшие у учащихся.

Учащиеся должны иметь при себе конспект для записи лекции, предварительно повторить предыдущий материал для качественного усвоения материала.

Процесс приобретения знаний в рамках образовательного процесса осуществляется по следующей схеме:

1. Лекционное занятие, которое включает: ознакомление с новым материалом и конспектирование лекции учащимися.
2. Обсуждение результатов проведенного лекционного занятия (преподавателем и учащимися).
3. Оценка теоретической познавательной деятельности.

Таким образом, лекционное занятие является важной составляющей процесса обучения. В ходе его проведения у учащихся формируются умения и навыки анализировать и делать выводы из наблюдений. Проявляются такие качества личности как терпение, культура поведения и находчивость.

Разработанный план-конспект лекционного занятия «Модульное тестирование» представлен в приложении Е, план-конспект лекционного занятия «Интеграционное тестирование» — в приложении Ж, план-конспект лекционного занятия «Системное тестирование» — в приложении И. Каждый из разработанных план-конспектов содержит название дисциплины, тему учебного занятия, форму проведения учебного занятия, поставленные при разработке цели лекции: обучающая, воспитательная и развивающая. Также стоит отметить, что для структурирования и повышения качества изложения лекционного материала, содержание материала систематизировано в соответствии с разработанным планом каждой лекции.

Теперь рассмотрим лабораторные занятия.

Лабораторные работы планируются в процессе изучения наиболее сложных тем учебной дисциплины, если необходимо разъяснение или подтверждение сущности законов, процессов, явлений, проведение эксперимента и т. д.

Лабораторная работа может выполняться после прохождения всего теоретического курса (последовательный метод) или в процессе его освоения (параллельный метод). Организационно такие занятия могут быть фронтальными (когда все обучающиеся выполняют одну и ту же работу) или групповыми (когда обучающиеся разделены на небольшие подгруппы из 2–5 человек, выполняющие разные работы) [6, с.4].

Лабораторные работы по учебным дисциплинам теоретического обучения носят как иллюстративный, так и исследовательский характер [6, с.5].

Иллюстративный характер проявляется в том, что обучающиеся в процессе проведения опытов воспроизводят то, что им известно из теоретических учебных занятий или учебных пособий. Исследовательский характер заключается в том, что на основании опытов практической деятельности обучающиеся делают самостоятельные выводы [6, с.5].

По характеру занятия лабораторная работа также относится к лекционно-семинарским формам обучения. Она также содержит элемент самостоятельной работы учащихся.

Учащиеся должны быть заранее предупреждены о предстоящей лабораторной работе. Их задача состоит в подготовке к данному виду работы. Они должны просмотреть материал в учебных пособиях, своих конспектах, дополнительной литературе, при необходимости проконсультироваться с преподавателем. Далее практические действия учащийся совершает на занятии самостоятельно под контролем преподавателя. Учащиеся успешно справятся с работой, если будут знать порядок её выполнения. Им надо не только рассказать, но и показать, как лучше выполнить работу.

В начале занятия преподаватель сообщает тему и цели занятия, мотивирует учащихся на успешную учебную деятельность.

В конце занятия преподаватель обсуждает с учащимися результаты их практической деятельности, спрашивает, что было непонятно, что вызвало трудности, при необходимости отвечает на вопросы, возникшие у учащихся. Далее преподаватель принимает отчеты о выполненной работе, выставляет учащимся отметки и сообщает о результатах достижения целей занятия.

Процесс приобретения навыков в рамках образовательного процесса осуществляется по следующей схеме:

1. Практическое занятие, которое включает ознакомление с предстоя­щей работой и практическую деятельность учащихся.
2. Обсуждение результатов практической деятельности (преподавателем и учащимися)
3. Оценка результатов практической деятельности.

Несомненно, многолетняя практика такой схемы преподавания прикладных дисциплин оправдывает себя.

Таким образом, лабораторная работа является важной составляющей процесса обучения. В ходе выполнения у учащихся формируются умения и навыки работы с прикладными программами, вырабатываются самостоятельность, а также способность анализировать и делать выводы из наблюдений. Проявляются такие качества личности как терпение, усидчивость, целеустремлённость и может быть даже находчивость, а также самостоятельность и ответственность.

Разработанный план-конспект лабораторного занятия «Разработка плана и проведение модульного тестирования ПО» представлен в приложении К.

Разработанная инструкция к лабораторному занятию представлена в приложении Л.

Разработанные презентации к лекционным занятиям «Модульное тестирование», «Интеграционное тестирование» и «Системное тестирование» представлены в приложении М.

**Заключение**

В ходе курсовой работы были рассмотрены дидактические основы ме­тодики преподавания раздела «Организация тестирования программного обеспечения» дисциплины «Тестирование и отладка программного обеспечения» уровня среднего специального образования, а также разработано содержание и элементы методик преподавания дисциплины «Тестирование и отладка программного обеспечения» по разделу «Организация тестирования программного обеспечения».

Цели и задачи исследования выполнены.

В данной курсовой работе были:

а) выявлены и описаны основные характеристики образовательного процесса уровня среднего специального образования;

б) на основе анализа типовой учебной программы, частных методик, были выявлены, спрогнозированы и описаны особенности методики преподавания раздела «Организация тестирования программного обеспечения» дисциплины «Тестирование и отладка программного обеспечения» уровня среднего специального образования;

в) разработаны:

1. фрагмент учебной программы,
2. фрагмент календарно-тематического плана и содержания
3. лекционное занятие по теме «Модульное тестирование»,
4. лекционное занятие по теме «Интеграционное тестирование»,
5. лекционное занятие по теме «Системное тестирование», а также
6. лабораторное занятие по теме «Разработка плана и проведение модульного тестирования ПО».

Все разработанные в курсовой работе учебно-программные документы, их фрагменты находятся во взаимосвязи и соответствуют друг другу.

При разработке элементов обучения были проанализированы и использованы современная учебно-программная документация, учебно-методические источники и нормативно-правовые акты, Кодекс Республики Беларусь об образовании.

Данные элементы обучения могут быть использованы при проведении занятий по дисциплине «Тестирование и отладка программного обеспечения» в учреждениях со средним специальным образованием, обеспечивающих обучение по специальности 2-40 01 01 «Программное обеспечение информационных технологий».

**Список использованных источников**

[1] Кодекс Республики Беларусь об образовании от 13.01.2011 № 243-З (с изм. и доп.) // Национальный реестр правовых актов Республики Беларусь. – 2011. – № 2/1795.

[2] Тестирование и отладка программного обеспечения: типовая учебная программа по учебной дисциплине профессионального компонента типового учебного пла­на по специальности 2-40 01 01 «Программное обеспечение информационных технологий»: утверждено постановлением Министерства образования Республики Беларусь от 26.01.2017 №6 // Республиканский институт профессионального образования [Электронный ресурс]. - Режим доступа: http://ripo.unibel.by/umosso/bankk/2429%20Тестирование%20и%20отладка%20программного%20обеспечения%20Спец%202-40%2001%2001.pdf - Дата доступа: 28.04.2020.

[3] Перечень действующих типовых учебных программ по учебным дисциплинам профессионального компонента и практике, типовых учебных планов по специальностям (направлениям специальностей), закрепленных за Министерством образования Республики Беларусь, для учреждений образования, реализующих образовательные программы среднего специального образования: утверждено постановлением Министерства образования Республики Беларусь от 21.08.2019 // Образовательный портал «Профессиональное образование» [Электронный ресурс]. - Режим доступа: http://ripo.unibel.by/index.php?id=1657 - Дата доступа: 28.04.2020.

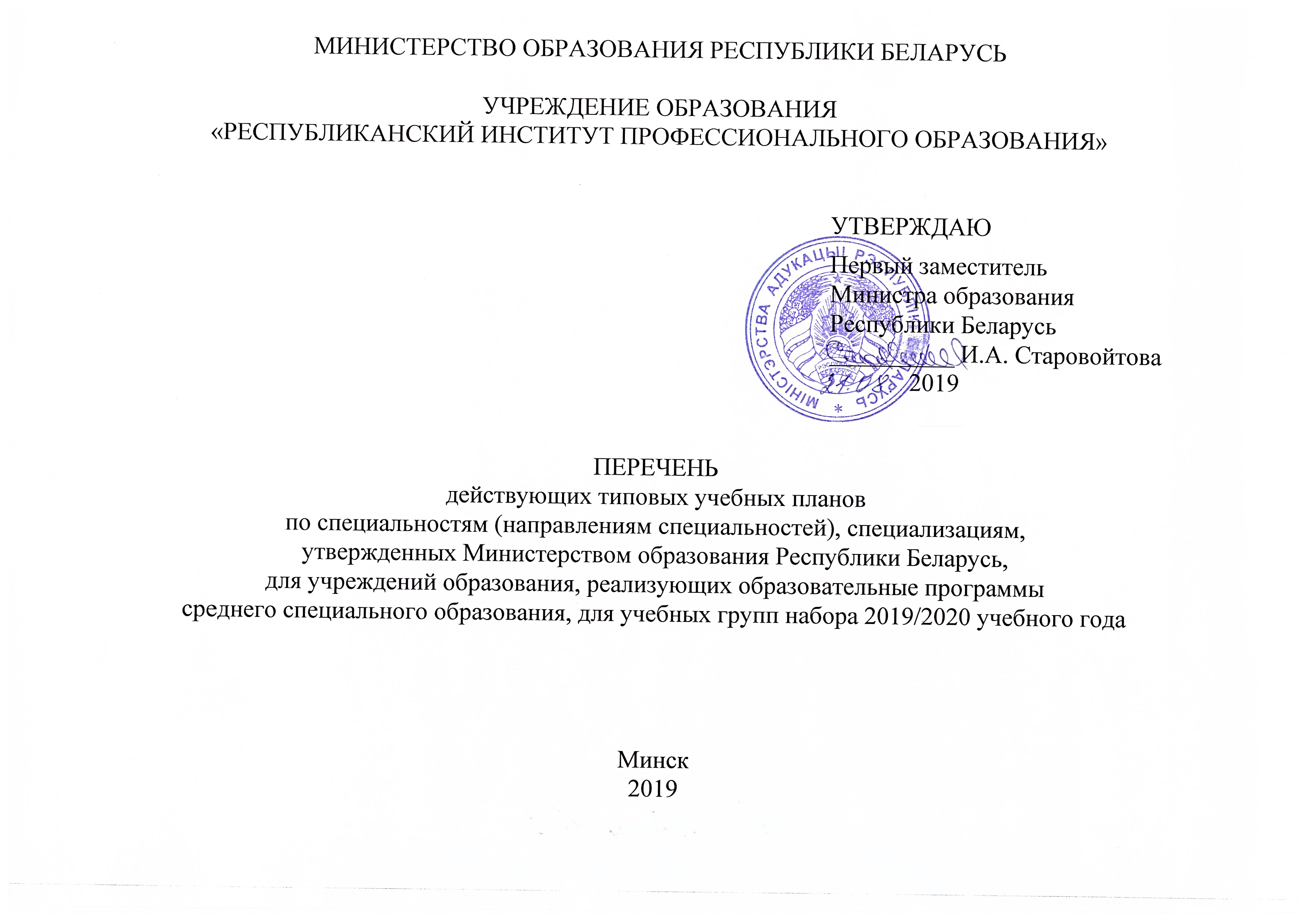
[4] Перечень действующих типовых учебных планов по специальностям (направлениям специальностей), специализациям, утвержденных Министерством образования Республики Беларусь, для учреждений образования, реализующих образовательные программы среднего специального образования, для учебных групп набора 2019/2020 учебного года: утверждено постановлением Министерства образования Республики Беларусь от 21.08.2019 // Образовательный портал «Профессиональное образование» [Электронный ресурс]. - Режим доступа: http://ripo.unibel.by/index.php?id=1656 - Дата доступа: 28.04.2020.

[5] Образовательный стандарт среднего специального образования по специальности 2-40 01 01 «Программное обеспечение информационных технологий»: ОС РБ 2-40 01 01-2014. – Утв. постановлением Министерства образования Респ. Беларусь № 54 от 28.04.2014 г. – Минск : М-во обр. Респ. Беларусь : УО РИПО, 2014. – 30с.

[6] Ильин, М.В. Разработка учебных программ допрофессиональной и профессиональной подготовки учащихся учреждений общего среднего и специального образования : метод. рекомендации / М. В. Ильин, А. Н. Петрова, П. Н. Сидорович ; под общ. ред. М. В. Ильина. – Минск : РИПО, 2019. – 44 c.

Приложение А

**Фрагмент перечня действующих типовых учебных планов**



**ПЕРЕЧЕНЬ**

**действующих типовых учебных планов по специальностям (направлениям специальностей), специализациям,   
утвержденных Министерством образования Республики Беларусь, для учреждений образования,   
реализующих образовательные программы среднего специального образования,  
для учебных групп набора 2019/2020 учебного года**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код профиля и направления образования, группы специальностей, специальности, направления специальности, спе­циализации | Наименование специальности,  направления специальности,  специализации | Типовые учебные планы по специальностям (направлениям специальностей) и специализациям | | Примечание |
| Регистрационный  индекс | Дата утверждения и номер постановления |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **40** | **ИНФОРМАТИКА И вычисли­тельная техника** |  |  |  |
| ***40 01*** | ***программные средства и математические*** |  |  |  |
| 2-40 01 01 | Программное обеспечение информационных технологий | РБ ст. № 930 Д/тип. | 14.05.2019 № 50 |  |
| 2-40 01 31 | Тестирование программного обеспеченпия | РБ ст. № 944 Д/тип. | 22.07.2019 № 114 |  |



Начальник Главного управления

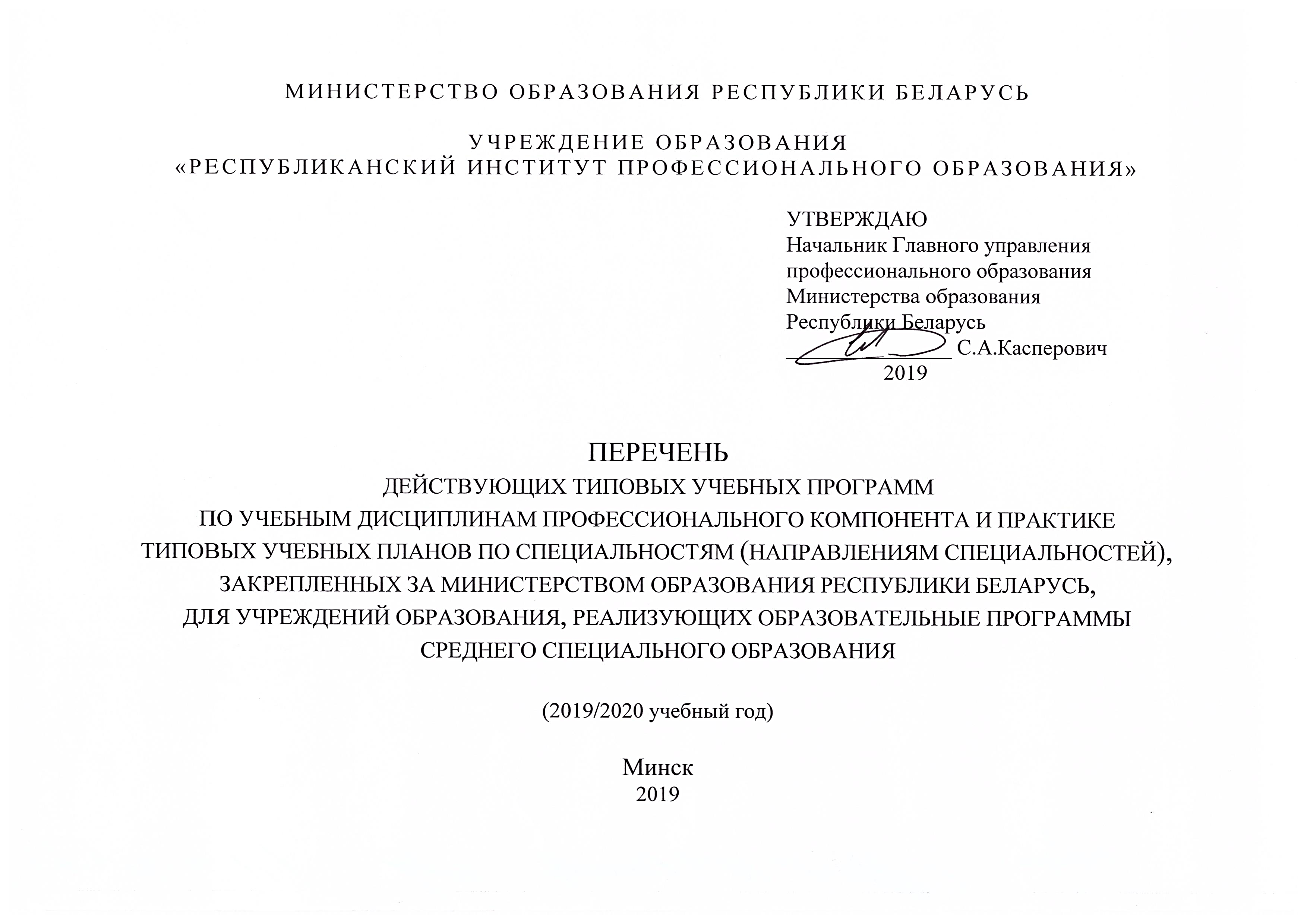
профессионального образования

Министерства образования Республики Беларусь С.А.Касперович

Ректор учреждения образования

Приложение Б

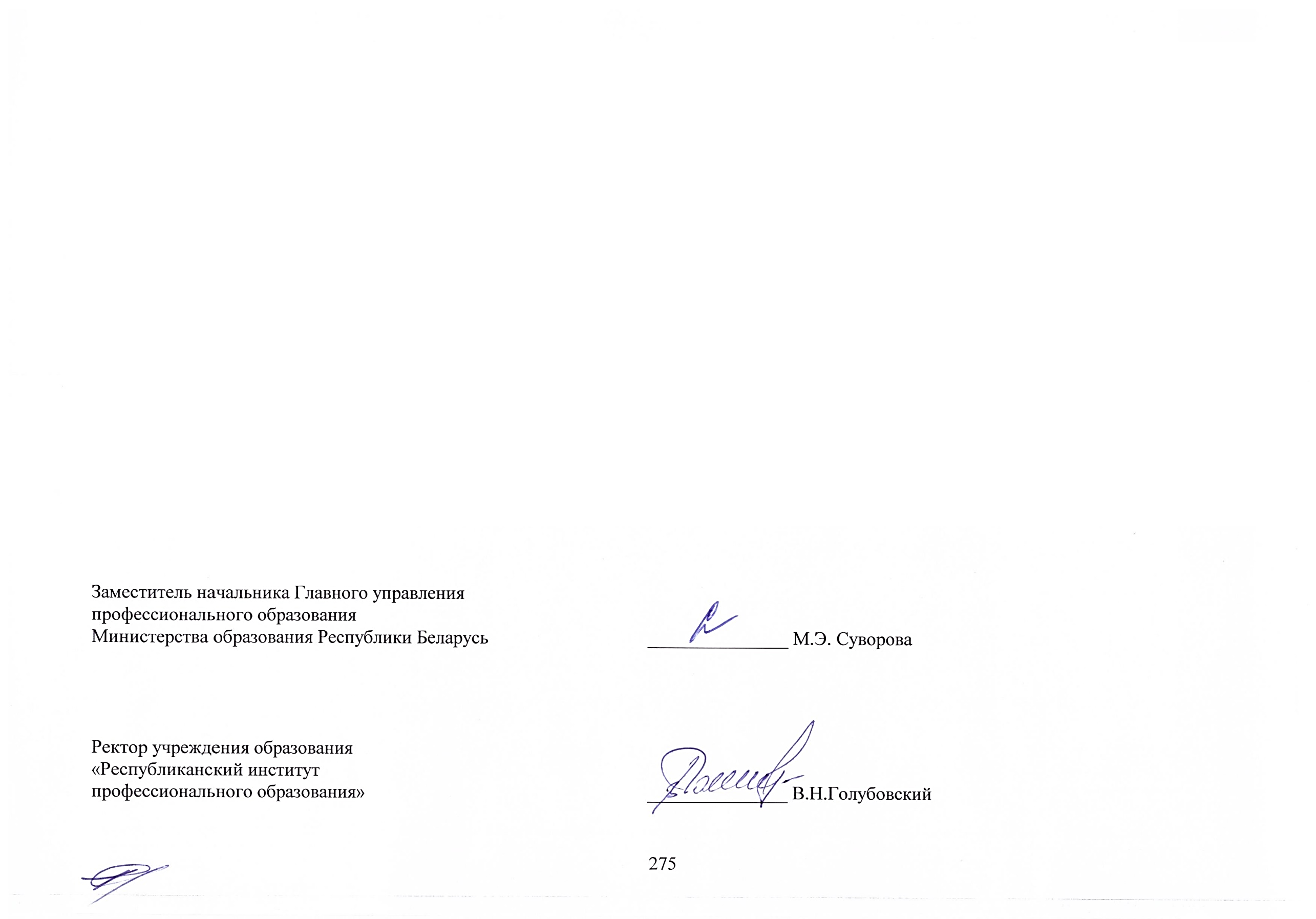
**Фрагмент перечня действующих типовых учебных программ**



Специальность 2-40 01 31 «Тестирование программного обеспечения»

Типовой учебный план РБ ст. № 994 Д/тип., утвержденный постановлением Министерства образования Республики Беларусь 22.07.2019 № 114

| Наименование типовых учебных программ | Количество учебных часов по | | Год  утверждения | Кем утверждена | Примечание |
| --- | --- | --- | --- | --- | --- |
| типовому  учебному  плану | типовой  учебной  программе |
| 1. Математика | 100 | 90-120 | 2014 | Министерством образования  Республики Беларусь |  |
| 1. Основы инженерной графики | 40 |  |  |  | ПТП |
| 1. Основы алгоритмизации и программирования | 172 |  |  |  | ПТП |
| 1. Арифметико-логические основы вычислительной техники | 106 |  |  |  | ПТП |
| 1. Стандартизация и сертификация программного обеспечения | 40 | 40 | 2017 | Министерством образования  Республики Беларусь |  |
| 1. Сети телекоммуникаций | 48 |  |  |  | ПТП |
| 1. Информационная безопасность телекоммуникационных систем | 40 | 40 | 2017 | Министерством образования  Республики Беларусь |  |
| 1. Охрана труда | 60 | 60 | 2012 | Министерством образования  Республики Беларусь |  |
| 1. Охрана окружающей среды и энергосбережение | 40 | 40 | 2019 | Министерством образования  Республики Беларусь |  |
| 1. Арифметико-логические основы вычислительной техники | 96 |  |  |  | ПТП |
| 1. Техника коммуникации и основы командообразования | 72 |  |  |  | ПТП |
| 1. Белорусский язык (профессиональная   лексика) | 20 | 20 | 2016 | Министерством образования Республики Беларусь |  |
| 1. Английский язык (профессиональная   лексика) | 102 |  |  |  | ПТП |
| 1. Математическое моделирование | 82 |  |  |  | ПТП |
| 1. Объектно-ориентированное программирование | 64 |  |  |  | ПТП |
| 1. Конструирование программ и языки программирования | 96 |  |  |  | ПТП |
| 1. Модели данных и системы управления   базами данных | 64 |  |  |  | ПТП |
| 1. Программирование для Интернета | 96 |  |  |  | ПТП |
| 1. Технология разработки программного   обеспечения | 64 |  |  |  | ПТП |
| 1. Технология тестирования программного обеспечения | 132 |  |  |  | ПТП |
| 1. Автоматизированное тестирование   программного обеспечения | 96 | 96 | 2017 | Министерством образования Республики Беларусь | ПТП |
| 1. Учебная и производственная практика |  |  |  |  | ПТП |



Заместитель начальника Главного управления

профессионального образования

Министерства образования Республики Беларусь \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.Э. Суворова

Ректор учреждения образования

«Республиканский институт

профессионального образования» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Приложение В

**Фрагмент типовой учебной программы дисциплины**

МИНИИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

**УТВЕРЖДЕНО**

Постановление

Министерства образования

Республики Беларусь

26.01.2017 №6

**ТИПОВАЯ УЧЕБНАЯ ПРОГРАММА**

**ПО УЧЕБНОЙ ДИСЦИПЛИНЕ (фрагмент)**

**«**ТЕСТИРОВАНИЕ И ОТЛАДКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

профессионального компонента типового учебного плана

по специальности 2-40 01 01

«Программное обеспечение информационных технологий»

для реализации образовательной программы среднего специального

образования, обеспечивающей получение квалификации специалиста

со средним специальным образованием

Минск

2017

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Типовая учебная программа по учебной дисциплине «Тестирование и отладка программного обеспечения» (далее – программа) предусматривает изучение видов, назначения и применения методов тестирования программного обеспечения.

Цель преподавания учебной дисциплины – развитие профессиональной компетентности в области тестирования и отладки сложных программных средств.

Изучение программного учебного материала базируется на знаниях, полученных учащимися в ходе изучения таких учебных дисциплин, как «Основы алгоритмизации и программирование», «Конструирование программ и языки программирования», «Структуры и алгоритмы обработки данных», «Технология разработки программного обеспечения».

Учебная дисциплина «Тестирование и отладка программного обеспечения» является практикоориентированной. Для закрепления теоретического материала и формирования у учащихся необходимых умений и навыков программой предусматривается проведение лабораторных занятий. Форма проведения лабораторных занятий определяется преподавателем исходя из цели обучения и содержания учебного материала.

В целях контроля усвоения программного учебного материала предусматривается проведение одной обязательной контрольной работы (ОКР). Содержание и конкретные сроки проведения ОКР определяются преподавателем, обсуждаются на заседании предметной (цикловой) комиссии и утверждаются в установленном порядке.

Программой определены цели изучения каждой темы, спрогнозированы результаты их достижения в соответствии с уровнями усвоения учебного материала.

Цели изучения учебной дисциплины

образовательные:

– обеспечить получение знаний по тестированию и отладке программного обеспечения и формирование умений и навыков тестирования программного обеспечения с использованием современных технологий и подходов;

– способствовать развитию логического мышления;

– сформировать умение использовать современные программы для автоматизации тестирования;

– отработать навыки на различных видах тестирования;

– сформировать умение составлять документацию о найденных ошибках;

развивающие:

– обеспечить условия для развития наблюдательности, внимательности и умений выделять существенные недочёты при тестировании программного обеспечения;

– способствовать развитию умения составлять необходимую документацию;

– способствовать развитию умений творческого подхода к решению практических задач;

– развивать критическое мышление, групповую самоорганизацию, умение вести диалог;

воспитательные:

– сформировать умения общаться и работать в команде;

– создать условия, обеспечивающие воспитание интереса к будущей профессии;

– обеспечить условия для воспитания творческого отношения к избранной профессии;

– воспитание трудолюбия, инициативности и настойчивости в преодолении трудностей.

В результате изучения учебной дисциплины учащиеся должны знать на уровне представления:

Знать на уровне представления:

* виды, уровни, направления и методы тестирования;
* критерии выбора тестов и оценки качества программного обеспечения;
* понятие верификации программного обеспечения;
* особенности документирования дефектов с использованием систем отслеживания проблем;

Знать на уровне понимания:

* + значение основных терминов, используемых в области тестирования и отладка программного обеспечения;
  + особенности проведения модульного, системного и интеграционного тестирования;
  + требования к составлению отчётов об ошибках;
  + особенности тестирования веб-приложений;
  + основы тестирования безопасности, производительности, регрессионного тестирования;
  + особенности выполнения автоматизированного тестирования;

Уметь:

* + проводить тестирование структуры программных модулей и их взаимодействия;
  + выполнять разработку тестовых сценариев;
  + составлять отчёты об ошибках;
  + проводить отладку и функциональное тестирование веб-ориентированных приложений;
  + использовать инструментальные средства при проведении автоматизированного тестирования и отладки программного обеспечения.

Дисциплина «Тестирование и отладка программного обеспечения» рассчитана на 40 часов занятий (20 часов лекций и 20 часа лабораторных занятий).

ПРИМЕРНЫЙТЕМАТИЧЕСКИЙ ПЛАН

|  |  |  |
| --- | --- | --- |
| Раздел, тема | Количество часов | |
| всего | в том числе  на ла­бора­торные  работы |
| **Введение** | **1** |  |
| Раздел 1. **Основы тестирования ПО** | **9** | **2** |
| 1.1. Тестирование как элемент жизненного цикла ПО. Дефекты ПО и их жизненный цикл | 1 |  |
| 1.2. Типы процессов тестирования | 2 |  |
| 1.3. Критерии выбора тестов | 4 | 2 |
| 1.4. Методы тестирования | 2 |  |
| Раздел 2. **Структурное и функциональное тестирование ПО** | **20** | **8** |
| 2.1. Структурное тестирование ПО (тестирование методом «белого ящика») | 10 | 6 |
| 2.2. Тестирование потоков данных | 2 |  |
| 2.3. Повторяемость тестирования | 2 |  |
| 2.4. Функциональное тестирование ПО (тестирование методом «черного ящика») | 5 | 2 |
| Обязательная контрольная работа | 1 |  |
| Раздел 3. **Организация тестирования ПО** | **40** | **20** |
| 3.1. Методика тестирования программных систем | 2 |  |
| 3.2. Модульное тестирование | 6 | 4 |
| 3.3. Интеграционное тестирование | 6 | 4 |
| 3.4. Системное тестирование | 4 | 2 |
| 3.5. Отладка ПО, ее виды | 2 |  |
| 3.6. Тестирование пользовательского интерфейса | 4 | 2 |
| 3.7. Тестирование объектно-ориентированных ПС | 6 | 4 |
| 3.8. Особенности тестирования Web-приложений | 2 |  |
| 3.9. Регрессионное тестирование | 2 |  |
| 3.10. Документирование как основа тестирования | 6 | 4 |
| Раздел 4. **Тестирование при промышленной разработке ПО** | **6** |  |
| 4.1. Автоматизация тестирования | 4 |  |
| 4.2. Особенности индустриального тестирования | 2 | 14 |
| **ИТОГО** | **76** | **30** |

Приложение Г

**Фрагмент учебной программы дисциплины**

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

ФИЛИАЛ БЕЛОРУССКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

«МИНСКИЙ РАДИОТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

**УТВЕРЖДАЮ**

Директор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ф.И.О

«\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

      (дата утверждения)

Регистрационный № \_\_\_\_\_\_\_\_\_

ТЕСТИРОВАНИЕ И ОТЛАДКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Учебная программа

для учреждений, обеспечивающих получение среднего специального

образования по специальности

2-40 01 01 «Программное обеспечение информационных технологий»

Минск, 2020

СОСТАВИТЕЛЬ:\_\_\_Новосёлов Н.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Предметно-методической комиссией «\_\_\_\_\_\_\_\_\_» Учреждения образования «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_» (протокол № \_\_\_ от \_\_\_. \_\_\_\_\_\_202\_\_ г.)

Педагогическим советом Учреждения образования «\_\_\_\_\_\_\_\_\_\_\_» (протокол №\_\_\_от\_\_\_.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.202\_\_\_ г.)

Методическая экспертиза \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (ФИО методист УМО)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Типовая учебная программа учебной дисциплины «Тестирование и отладка программного обеспечения» (далее — программа) предусматривает изучение видов, назначения и применения методов тестирования программного обеспечения.

Цель преподавания учебной дисциплины – развитие профессиональной компетентности в области тестирования и отладки сложных программных средств.

Изучение программного учебного материала базируется на знаниях, полученных учащимися в ходе изучения таких учебных дисциплин, как «Основы алгоритмизации и программирование», «Конструирование программ и языки программирования», «Структуры и алгоритмы обработки данных», «Технология разработки программного обеспечения».

Для закрепления теоретического материала и формирования у учащихся необходимых умений и навыков программой предусматривается проведение лабораторных занятий.

В результате изучения учебной дисциплины учащиеся должны:

Знать на уровне представления:

* виды, уровни, направления и методы тестирования;
* критерии выбора тестов и оценки качества программного обеспечения;
* понятие верификации программного обеспечения;
* особенности документирования дефектов с использованием систем отслеживания проблем;

Знать на уровне понимания:

* + значение основных терминов, используемых в области тестирования и отладка программного обеспечения;
  + особенности проведения модульного, системного и интеграционного тестирования;
  + требования к составлению отчётов об ошибках;
  + особенности тестирования веб-приложений;
  + основы тестирования безопасности, производительности, регрессионного тестирования;
  + особенности выполнения автоматизированного тестирования;

Уметь:

* + проводить тестирование структуры программных модулей и их взаимодействия;
  + выполнять разработку тестовых сценариев;
  + составлять отчёты об ошибках;
  + проводить отладку и функциональное тестирование веб-ориентированных приложений;
  + использовать инструментальные средства при проведении автоматизированного тестирования и отладки программного обеспечения [2, c.5].

В программе приведены примерные критерии оценки результатов учебной деятельности учащихся по учебной дисциплине, которые разработаны на основе десятибалльной шкалы и показателей оценки результатов учебной деятельности, обучающихся в учреждениях среднего специального образования.

Программа содержит примерный перечень оснащения лаборатории техническими и программными средствами, необходимыми для обеспечения образовательного процесса.

Приведенный в программе тематический план является рекомендательным. Предметная (цикловая) комиссия учреждения образования может вносить обоснованные изменения в содержание и последовательность изложения программного учебного материала, распределение учебных часов по темам в пределах общего бюджета времени, отведенного на изучение учебной дисциплины. Все изменения должны быть утверждены заместителем руководителя учреждения образования по учебной работе.

Дисциплина «Тестирование и отладка программного обеспечения» рассчитана на 40 часов занятий (20 часов лекций и 20 часов лабораторных занятий).

**ПРИМЕРНЫЙ ТЕМАТИЧЕСКИЙ ПЛАН**

Дисциплины «Тестирование и отладка программного обеспечения»

|  |  |  |  |
| --- | --- | --- | --- |
| Раздел, тема | Количество часов | | |
| всего | в том числе | |
| на ла­бора­торные  работы | на практические  работы |
| **Раздел 3. Организация тестирования ПО** | **40** | **20** |  |
| 3.1. Методика тестирования программных систем | 2 |  |  |
| 3.2. Модульное тестирование | 6 | 4 |  |
| 3.3. Интеграционное тестирование | 6 | 4 |  |
| 3.4. Системное тестирование | 4 | 2 |  |
| 3.5. Отладка ПО, ее виды | 2 |  |  |
| 3.6. Тестирование пользовательского интерфейса | 4 | 2 |  |
| 3.7. Тестирование объектно-ориентированных ПС | 6 | 4 |  |
| 3.8. Особенности тестирования Web-приложений | 2 |  |  |
| 3.9. Регрессионное тестирование | 2 |  |  |
| 3.10. Документирование как основа тестирования | 6 | 4 |  |
| **ИТОГО** | **40** | **20** |  |

**СОДЕРЖАНИЕ ПРОГРАММЫ**

|  |  |  |
| --- | --- | --- |
| Цель изучения темы | Содержание темы | Результат |
| Раздел 3.Организация тестирования ПО | | |
| Тема 3.1. Методика тестирования программных систем | | |
| Дать понятие об организации тестирования ПО и фазах тестирования. Сформировать знания о методике тестирования ПС, о нисходящей и восходящей стратегиях тестирования. | Организация тестирования. Три фазы тестирования. Методика тестирования программных систем (ПС). Нисходящая и восходящая стратегии тестирования, особенности тестового окружения для них. | Излагает сведения об организации тестирования ПО и фазах тестирования. Описывает методику тестирования ПС. Объясняет особенности нисходящей и восходящей стратегий тестирования. |
| Тема 3.2. **Модульное тестирование** | | |
| Сформировать понятие об особенностях модульного тестирования. Дать понятие об организации модульного тестирования. | Цели и задачи модульного тестирования. Понятие о модуле и его границах. Подходы к проектированию тестового окружения. Организация модульного тестирования. | Излагает цели и задачи модульного тестирования, подходы к проектированию тестового окружения. Объясняет особенности организации модульного тестирования. |
| *Лабораторная работа № 5 (4 часа)* | | |
| Сформировать умение разрабатывать план и проводить модульное тестирование ПО. | Разработка плана и проведение модульного тестирования ПО. | Разрабатывает план, проводит модульное тестирование ПО, анализирует результаты. |
| Тема 3.3. **Интеграционное тестирование** | | |
| Сформировать знания об особенностях интеграционного тестирования. Дать понятие об организации интеграционного тестирования. | Цели и задачи интеграционного тестирования. Организация интеграционного тестирования. Структурная классификация методов интеграционного тестирования. Временная классификация методов интеграционного тестирования. Планирование интеграционного тестирования. | Излагает цели и задачи интеграционного тестирования, подходы к проектированию тестового окружения. Объясняет особенности организации интеграционного тестирования. |
| *Лабораторная работа № 6 (4 часа)* | | |
| Сформировать умение разрабатывать план и проводить интеграционное тестирование ПО. | Разработка плана и проведение интеграционного тестирования ПО. | Разрабатывает план, проводит интеграционное тестирование ПО, анализирует результаты. |
| Тема 3.4. **Системное тестирование** | | |
| Сформировать знания об особенностях системного тестирования. Дать понятие об организации системного тестирования. | Цели и задачи системного тестирования. Виды системного тестирования. Системное тестирование, приемо-сдаточные и сертификационные испытания при разработке сертифицируемого программного обеспечения. | Излагает цели и задачи системного тестирования, описывает его виды. Объясняет особенности организации системного тестирования. |
| *Лабораторная работа № 7* | | |
| Сформировать умение разрабатывать план и проводить системное тестирование ПО. | Разработка плана и проведение системного тестирования ПО. | Разрабатывает план, проводит системное тестирование ПО, анализирует результаты. |
| Тема 3.5. **Отладка ПО, ее виды** | | |
| Сформировать понятие о классификации ошибок, методах отладки ПО. Дать понятие об общей методике отладки ПО. | Классификация ошибок. Методы отладки ПО. Методы и средства получения дополнительной информации. Общая методика отладки ПО. | Описывает типы ошибок, излагает методы отладки ПО. Объясняет особенности общей методики отладки ПО. |
| Тема 3.6. **Тестирование пользовательского интерфейса** | | |
| Сформировать знания о целях, задачах и особенностях тестирования пользовательского интерфейса. Дать понятие о содержании работ при функциональном тестировании пользовательских интерфейсов, о содержании работ при тестировании удобства использования пользовательских интерфейсов. | Цели, задачи и особенности тестирования пользовательского интерфейса. Функциональное тестирование пользовательских интерфейсов. Проверка требований к пользовательскому интерфейсу. Полнота покрытия. Методы проведения, повторяемость тестирования пользовательского интерфейса. Тестирование удобства использования пользовательских интерфейсов. | Излагает цели, задачи и особенности тестирования пользовательского интерфейса. Объясняет содержание работ при функциональном тестировании пользовательских интерфейсов, содержание работ при тестировании удобства использования пользовательских интерфейсов. |
| *Лабораторная работа № 8* | | |
| Сформировать умение разрабатывать план и проводить тестирование пользовательского интерфейса. | Разработка плана и проведение тестирования пользовательского интерфейса. | Разрабатывает план, проводит тестирование пользовательского интерфейса, анализирует результаты. |
| Тема 3.7. **Тестирование объектно-ориентированных ПС** | | |
| Сформировать знания об особенностях тестирования объектноориентированных ПС. Дать понятие об особенностях проектирования объектно-ориентированных тестов, о тестировании содержания и взаимодействия классов. Сформировать знания об особенностях интеграционного тестирования объектно-ориентированных ПС. | Объектно-ориентированное тестирование. Проектирование объектно-ориентированных тестовых вариантов. Тестирование содержания классов. Тестирование взаимодействия классов. Предваряющее тестирование при экстремальной разработке. Особенности интеграционного тестирования объектно-ориентированных ПС. | Излагает особенности тестирования объектно-ориентированных ПС. Объясняет особенности проектирования объектно-ориентированных тестов, особенности тестирования содержания и взаимодействия классов. Излагает особенности интеграционного тестирования объектноориентированных ПС. |
| *Лабораторная работа № 9 (4 часа)* | | |
| Сформировать умение разрабатывать план и проводить тестирование объектно-ориентированных ПС. | Разработка плана и проведение тестирования объектно-ориентированных ПС. | Разрабатывает план, проводит тестирование объектно-ориентированных ПС, анализирует результаты. |
| Тема 3.8. **Особенности тестирования Web-приложений** | | |
| Сформировать знания об особенностях тестирования Web-приложений. Дать понятие об особенностях нагрузочного тестирования, тестирования безопасности. Сформировать знания об организации тестирования Web-приложений. | Цели, задачи и особенности тестирования Web-приложений. Нагрузочное тестирование. Тестирование безопасности. Подходы к проектированию тестового окружения. Организация тестирования Web-приложений. | Излагает цели, задачи и особенности тестирования Web-приложений. Объясняет особенности нагрузочного тестирования, тестирования безопасности. Излагает организацию тестирования Web-приложений. |
| Тема 3.9. **Регрессионное тестирование** | | |
| Сформировать знания об особенностях регрессионного тестирования. Дать понятие о видах регрессионного тестирования, о его организации. | Цели, задачи и особенности регрессионного тестирования. Виды регрессионного тестирования. Управляемое регрессионное тестирование. Методы отбора тестов. | Излагает цели, задачи и особенности регрессионного тестирования. Описывает виды регрессионного тестирования, особенности его организации. |
| Тема 3.10. **Документирование как основа тестирования** | | |
| Сформировать представление о документах, сопровождающих тестирование и верификацию ПО. | Документация, создаваемая на различных этапах жизненного цикла. Документация, сопровождающая процессы верификации и тестирования. Стратегия и планы верификации. Тест-требования. Тест-планы. Отчеты о прохождении тестов. Отчеты о покрытии программного кода. Отчеты о проблемах. Трассировочные таблицы | Высказывает общее суждение о составе документов, создаваемых на различных этапах жизненного цикла ПО, о структуре и содержа11 нии документов, сопровождающих тестирование и верификацию ПО |
| *Лабораторная работа № 10 (4 часа)* | | |
| Сформировать навыки разработки отчетов о прохождении тестов и покрытии программного кода, отчета о проблемах в ПС. | Документирование тестирования. | Разрабатывает отчеты о прохождении тестов и покрытии программного кода, отчет о проблемах в ПС. |

**ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ОСНАЩЕНИЯ ЛАБОРАТОРИИ**

|  |  |
| --- | --- |
| Наименование средств обучения | Количество |
| **Технические средства обучения** | |
| Компьютер | 16 |
| Комплект сетевого оборудования | 1 |
| **Программное обеспечение** | |
| Операционная система Windows |  |
| Интегрированная среда разработки ПО для Microsoft Windows и язык программирования Delphi 7.0 |  |
| Среда разработки PascalABC и язык программирования Pascal |  |
| Интегрированная среда разработки ПО для Microsoft Windows |  |
| Visual Studio 2008 и языки программирования C/C++/C# |  |
| Система автоматизированного тестирования |  |
| Текстовый редактор |  |
| **Средства защиты** | |
| Аптечка медицинская | 1 |
| Заземление | 1 |
| Огнетушитель | 1 |
| **Оборудование помещения** | |
| Доска аудиторная | 1 |
| Стол для преподавателя | 1 |
| Стол аудиторный | 15 |
| Стул | 31 |
| Шкаф книжный | 1 |

**ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ**

**Основная**

Бейзер, Б. Тестирование «черного ящика». Технологии функционального тестирования программного обеспечения и систем / Б. Бейзер. СПб., 2004.

Иванова, Г.С. Технология программирования / Г.С. Иванова. М., 2002.

Канер, С. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений / С. Канер, Д. Фолк, К.Н. Енг. Киев, 2001.

Котляров, В.П. Основы тестирования программного обеспечения / В.П. Котляров, Т.В. Коликова. М., 2006.

Степанченко, И.В. Методы тестирования программного обеспечения : учеб. пособие / И.В. Степанченко. Волгоград, 2006.

Тамре, Л. Введение в тестирование программного обеспечения / Л. Тамре. М., 2003.

**Дополнительная**

Дастин, Э. Автоматизированное тестирование программного обеспечения / Э. Дастин, Д. Рэшка, Д. Пол. М., 2003.

Калбертсон, Р. Быстрое тестирование / Р. Калбертсон, К. Браун, Г. Кобб. М., 2006.

Макгрегор, Д. Тестирование объектно-ориентированного программного обеспечения : практ. пособие / Д. Макгрегор, Д. Сайкс. Киев, 2002.

**Нормативные правовые акты**

Конституция Республики Беларусь 1994 года (с изм. и доп., принятыми на республиканских референдумах 24.11.1996 и 17.10.2004 // Нац. реестр правовых актов Респ. Беларусь. 04.01.1999. № 1/0.

Кодекс Республики Беларусь об образовании от 13.01.2011 № 243-З // Нац. реестр правовых актов Респ. Беларусь. 17.01.2011. № 2/1795.

Единый тарифно-квалификационный справочник работ и профессий рабочих (ЕТКС): вып. 2: постановление М-ва труда Респ. Беларусь, 28 декабря 2000 г., № 160 // Нац. реестр правовых актов Респ. Беларусь. – 2012. – 8/25723.

**Технические нормативные правовые акты**

СТБ 982-94 Информационная технология. Термины и определения

СТБ ИСО 9000-2006 Системы менеджмента качества. Основные положения и словарь

ОКРБ 006-2009 Профессии рабочих и должности служащих

ОКРБ 011-2009 Специальности и квалификации

ГОСТ ИСО/МЭК 2382-1-99 Информационная технология. Словарь

**Стандарты**

Образовательный стандарт среднего специального образования по специальности 2-40 01 01 Программное обеспечение информационных технологий: ОС РБ 2-40 01 01-2014. – Утв. постановлением Министерства образования Респ. Беларусь № 54 от 28.04.2014 г. – Минск : М-во обр. Респ. Беларусь : УО РИПО, 2014. – 30с.

Приложение Д

**Фрагмент календарно-тематического плана**

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

ФИЛИАЛ БЕЛОРУССКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

«МИНСКИЙ РАДИОТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

|  |  |  |
| --- | --- | --- |
|  |  | УТВЕРЖДАЮ |
|  |  | Зам. директора  по учебной работе  \_\_\_\_\_\_\_\_\_\_\_\_\_ Ф.С.Шумчик  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_г. |

**КАЛЕНДАРНО-ТЕМАТИЧЕСКИЙ ПЛАН**

(очная форма получения образования)

по учебной дисциплине «Тестирование и отладка программного обеспечения»

для групп № 844691, 944691 очной формы получения образования

Преподаватель Новосёлов Никита Александрович

Составлен в соответствии с типовой учебной программой по учебной дисциплине «Тестирование и отладка программного обеспечения» профессионального компонента типового учебного плана по специальности 2-40 01 01 «Программное обеспечение информа­ционных технологий» для реализации образовательной программы среднего специального образования, обеспечивающей получение квалификации специалиста со средним специальным образованием, утвержденной постановлением Министерства образования Республики Беларусь 26.01.2017 № 6

Количество учебных часов по учебному плану 40

Из них:

на обзорные занятия 20

на лабораторные занятия (работы) 20

на практические занятия (работы) -

на курсовое проектирование -

Количество домашних контрольных работ -

Рассмотрен на заседании цикловой комиссии специальности «Программное обеспечение информационных технологий»

Протокол № \_\_\_\_\_ от «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г.

Председатель цикловой комиссии \_\_\_\_\_\_\_\_\_\_\_\_ /Лазицкас Е.А./

Разработчик \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Новосёлов Н.А./

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| №№ учеб­ных заня­тий | №№ учебных групп | | | | | Наименование разделов, наименование тем по учебной программе, наименование тем отдельных учебных занятий | Количество учебных часов | Тип учебного занятия | Учебно-методическое обеспечение | Задание для самостоятельной работы учащимся | Литература  (печатная, электронная) | Примечание |
| 844691 | 944691 |  |  |  |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|  |  |  |  |  |  | **Раздел 3. Организация тестирования ПО** |  |  |  |  |  |  |
| 1 |  |  |  |  |  | 3.1. Методика тестирования программных систем | 2 | УЗФНЗ | Презентация «Методика тестирования программных систем» |  | [1], c.15-28 |  |
| 2 |  |  |  |  |  | 3.2. Модульное тестирование | 2 | УЗФНЗ | Набор модульных тестов с пояснительными комментариями |  | [1], c.45-72 |  |
| 3,4 |  |  |  |  |  | ЛР №5 Разработка плана и проведение модульного  тестирования ПО | 4 | УЗППЗ |  |  |  |  |
| 5 |  |  |  |  |  | 3.3. Интеграционное тестирование | 2 | УЗФНЗ | Набор интеграционных тестов с пояснительными комментариями |  | [3], c.75-86 |  |
| 6,7 |  |  |  |  |  | ЛР №6 Разработка плана и проведение интеграционного тестирования ПО | 4 | УЗППЗ |  |  |  |  |
| 8 |  |  |  |  |  | 3.4. Системное тестирование | 2 | УЗФНЗ | Презентация по теме «Системное тестирование» |  | [2], c.82-96 |  |
| 9 |  |  |  |  |  | ЛР №7 Разработка плана и проведение системного  тестирования ПО | 2 | УЗППЗ |  |  |  |  |
| 10 |  |  |  |  |  | 3.5. Отладка ПО, ее виды | 2 | УЗФНЗ | Набор программ с ошибками для тестирования и отладки |  |  |  |
| 11 |  |  |  |  |  | 3.6. Тестирование пользовательского интерфейса | 2 | УЗФНЗ | Презентация по теме «Тестирование интерфейсов» |  |  |  |
| 12 |  |  |  |  |  | ЛР №8 Разработка плана и проведение тестирования  пользовательского интерфейса | 2 | УЗППЗ |  |  |  |  |
| 13 |  |  |  |  |  | 3.7. Тестирование объектно-ориентированных ПС | 2 | УЗФНЗ | Презентация по теме «Тестирование объектно-ориентированных ПС» |  |  |  |
| 14,15 |  |  |  |  |  | ЛР №9 Разработка плана и проведение тестирования  объектно-ориентированных ПС | 4 | УЗППЗ |  |  |  |  |
| 16 |  |  |  |  |  | 3.8. Особенности тестирования Web-приложений | 2 | УЗФНЗ | Презентация по теме «Особенности тестирования Web-приложений» |  | [4], c.102-112 |  |
| 17 |  |  |  |  |  | 3.9. Регрессионное тестирование | 2 | УЗФНЗ | Презентация по теме «Регрессионное тестирование» |  | [5], c.201-240 |  |
| 18 |  |  |  |  |  | 3.10. Документирование как основа тестирования | 2 | УЗФНЗ | Презентация по теме «Документирование как основа тестирования» |  | [6], c.151-166 | 20 |
| 19,20 |  |  |  |  |  | ЛР №10 Документирование тестирования | 4 | УЗППЗ |  |  |  |  |
|  |  |  |  |  |  | **Всего** | **40** |  |  |  |  |  |

Условные обозначения:

ЛР – лабораторная работа;

ПЗ – практическое занятие

ОКР – обязательная контрольная работа;

УЗФНЗ – учебное занятие формирования новых знаний;

УЗППЗ – учебное занятие практического применения знаний;

КУЗ – комбинированное учебное занятие;

УЗКЗ – учебное занятие контроля знаний.

**ЛИТЕРАТУРА**

|  |  |  |  |
| --- | --- | --- | --- |
| № №  п/п | Наименование | Автор | Издательство, год издания |
| Основная литература | | | |
| 1 | Тестирование «черного ящика». Технологии функционального  тестирования программного обеспечения и систем | Бейзер. Б. | СПб., 2014 |
| 2 | Технология программирования | Иванова. Г.С. | М., 2012 |
| 3 | Тестирование программного обеспечения. Фундаментальные  концепции менеджмента бизнес-приложений | Канер. С. | Киев, 2011 |
| 4 | Основы тестирования программного обеспечения | Котляров. В.П. | М., 2016 |
| 5 | Методы тестирования программного обеспечения | Степанченко. И.В. | Волгоград, 2016 |
| 6 | Введение в тестирование программного обеспечения | Тамре. Л. | М., 2013 |

Приложение Е

**ПЛАН-КОНСПЕКТ лекционного занятия**

**«Модульное тестирование» по дисциплине**

**«Тестирование и отладка программного обеспечения»**

**направления специальности 2-40 01 01 «программное**

**обеспечение информационных технологий»**

Дисциплина: Тестирование и отладка программного обеспечения

Тема учебного занятия: «**Модульное тестирование» (2 часа)**

**Цели лекции**: Сформировать понятие об особенностях модульного тестирования. Дать понятие об организации модульного тестирования.

**Форма проведения учебного занятия:** лекция

**Задачи лекции**:

1. обучающая:
2. сформировать представление об особенностях модульного тестирования;
3. сформировать понятие об организации модульного тестирования;
4. сформировать понятие о сущности и способах организации модульного тестирования;
5. воспитательная:
6. способствовать воспитанию усидчивости, организованности, дисциплинированности;
7. развивающая:
8. способствовать развитию умений учащихся обобщать полученные знания, проводить анализ, синтез, сравнения, делать необходимые выводы.

**План лекции:**

1. Понятие тестирования, виды тестирования.
2. Понятие модульного тестирования.
3. Преимущества модульного тестирования.
4. Недостатки модульного тестирования.
5. Инструментарий тестирования.
6. Заключение.
7. Вопросы для контроля знаний.

**Используемые средства обучения:** мультимедийная презентация.

**Конспект лекции:**

**1. Понятие тестирования, виды тестирования**

**Тестирование программного обеспечения** — проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом. В более широком смысле, тестирование — это одна из техник контроля качества, включающая в себя активности по планированию работ (Test Management), проектированию тестов (Test Design), выполнению тестирования (Test Execution) и анализу полученных результатов (Test Analysis).

**Качество программного обеспечения (Software Quality)** — это совокупность характеристик программного обеспечения, относящихся к его способности удовлетворять установленные и предполагаемые потребности.

**Цели тестирования:**

* Повысить вероятность того, что приложение, предназначенное для тестирования, будет работать правильно при любых обстоятельствах.
* Повысить вероятность того, что приложение, предназначенное для тестирования, будет соответствовать всем описанным требованиям.
* Предоставление актуальной информации о состоянии продукта на данный момент.

**Этапы тестирования:**

1. Анализ продукта;
2. Работа с требованиями;
3. Разработка стратегии тестирования и планирование процедур контроля качества;
4. Создание тестовой документации;
5. Тестирование прототипа;
6. Основное тестирование;
7. Стабилизация;
8. Эксплуатация.

**Уровни тестирования:**

1. **Модульное тестирование (Unit Testing)**
2. **Интеграционное тестирование (Integration Testing)**
3. **Системное тестирование (System Testing)**
4. **Операционное тестирование (Release Testing)**
5. **Приемочное тестирование (Acceptance Testing)**

**2. Понятие модульного тестирования**

**Модульное тестирование (Unit Testing)** — **проверяет функциональность и ищет дефекты в частях приложения, которые доступны и могут быть протестированы по-отдельности (модули программ, объекты, классы, функции и т.д.).**

**Это низший уровень тестирования, который применяется как при тестировании «белого ящика», так и при тестировании типа «чёрный ящик», поэтому является наиболее универсальным и распространённым.**

**На практике именно покрытие кода проекта модульными тестами является метрикой его качества и другие виды тестирования могут не применяться вовсе, хотя такая ситуация и редка.**

**Такое отношение к модульному тестированию сложилось на основании того, что корректная работа отдельных модулей системы зачастую означает корректную работу модулей в составе проекта.**

**3. Преимущества модульного тестирования**

**Цель модульного тестирования — изолировать отдельные части программы и показать, что по отдельности эти части работоспособны.**

**Этот тип тестирования обычно выполняется программистами.**

**Преимущества модульного тестирования:**

1. **Поощрение изменений** — **модульное тестирование позже позволяет программистам проводить рефакторинг, будучи уверенными, что модуль по-прежнему работает корректно (регрессионное тестирование). Это поощряет программистов к изменениям кода, поскольку достаточно легко проверить, что код работает и после изменений;**
2. **Упрощение интеграции** — **модульное тестирование помогает устранить сомнения по поводу отдельных модулей и может быть использовано для подхода к тестированию «снизу-вверх»: сначала тестируя отдельные части программы, а затем программу в целом;**
3. **Документирование кода** — **модульные тесты можно рассматривать как «живой документ» для тестируемого класса. Клиенты, которые не знают, как использовать данный класс, могут использовать юнит-тест в качестве примера;**
4. **Отделение интерфейса от реализации** — **поскольку некоторые классы могут использовать другие классы, тестирование отдельного класса часто распространяется на связанные с ним. Например, класс пользуется базой данных; в ходе написания теста программист обнаруживает, что тесту приходится взаимодействовать с базой. Это ошибка, поскольку тест не должен выходить за границу класса. В результате разработчик абстрагируется от соединения с базой данных и реализует этот интерфейс, используя свой собственный mock-объект. Это приводит к менее связанному коду, минимизируя зависимости в системе.**

**4. Недостатки модульного тестирования**

Несмотря на преимущества модульного тестирования, оно имеет ряд случаев, когда оно не работает, либо работает плохо.

Когда модульное тестирование не работает:

1. **Сложный код** — тестирование программного обеспечения — комбинаторная задача. Например, каждое возможное значение булевской переменной потребует двух тестов: один на вариант TRUE, другой — на вариант FALSE. В результате на каждую строку исходного кода потребуется 3−5 строк тестового кода.
2. **Результат известен лишь приблизительно** — бизнес-приложения зачастую работают с конечными и счётными множествами, научные — с континуальными. Поэтому сложно подобрать тесты для каждой из ветвей программы, сложно сказать, верен ли результат, выдерживается ли точность, и т. д. А во многих случаях качество моделирования определяется «на глаз», и последний результат записывается как «опорный». Если найдено расхождение, новый результат проверяют вручную и выясняют, какой качественнее: старый или новый.
3. **Код, взаимодействующий с системой** — код, взаимодействующий с портами, таймерами и прочими «нестабильными» частями системы, крайне сложно проверить в изолированном окружении.
4. **Ошибки интеграции и производительности** — при выполнении юнит-тестов происходит тестирование каждого из модулей по отдельности. Это означает, что ошибки интеграции, системного уровня, функций, исполняемых в нескольких модулях, не будут определены. Кроме того, данная технология бесполезна для проведения тестов на производительность. Таким образом, модульное тестирование более эффективно при использовании в сочетании с другими методиками тестирования.
5. **При общей низкой культуре программирования** — для получения выгоды от модульного тестирования требуется строго следовать технологии тестирования на всём протяжении процесса разработки программного обеспечения. Нужно хранить не только записи обо всех проведённых тестах, но и обо всех изменениях исходного кода во всех модулях. С этой целью следует использовать систему контроля версий ПО. Таким образом, если более поздняя версия ПО не проходит тест, который был успешно пройден ранее, будет несложным сверить варианты исходного кода и устранить ошибку. Также необходимо убедиться в неизменном отслеживании и анализе неудачных тестов. Игнорирование этого требования приведёт к лавинообразному увеличению неудачных тестовых результатов.
6. **Проблемы с объектами-заглушками** — за исключением простейших случаев, тестируемый объект должен взаимодействовать с другими объектами. Этих «товарищей по взаимодействию» — объекты-заглушки — делают предельно простыми: либо крайне упрощёнными (память вместо базы данных), либо рассчитанными на конкретный тест и механически повторяющими сессию обмена. Проблемы могут возникать при смене протокола обмена, в таком случае объекты-заглушки должны отвечать новым требованиям протокола.

**5. Инструментарий тестирования**

В целях экономии времени в списке далее будут представлены лишь наиболее популярные инструменты тестирования для наиболее популярных языков программирования.

1. **С**

* CUnit
* Cfix
* Sanity Autotest

1. **Ruby**

* Rspec
* Test::Unit

1. **C++**

* Boost Test
* Google C++ Testing Framework
* Symbian
* Sanity Autotest

1. **C#**

* Nunit
* XUnit.net
* MbUnit

1. **Java**

* Junit
* TestNG
* JavaTESK
* Spock

1. **PHP**

* PHPUnit
* SimpleTest

1. **JavaScript**

* Mocha
* Jest
* Chai
* Sinon.JS
* Karma runner
* QUnit
* JSUnit
* Jasmine

1. **Python**

* PyUnit
* PyTest
* Nose

**6. Заключение**

Тестирование является неотъемлемой частью коммерческой разработки, и любой разработчик должен владеть теоретической базой, а также навыками и инструментарием для выполнения тестирования собственного кода.

В случае, если покрытие кода тестами недостаточное – страдает качество проекта, что приводит к появлению «багов» и накапливанию «технического долга», который со временем приводит к тому, что внедрение новой функциональности становится крайне затруднительным как в плане трудозатрат, так и денежных затрат для заказчика проекта.

На сегодняшний день тестирование является общепринятым средством поддержания должного качества кода, однако зачастую тестированию уделяется недостаточное внимание, из-за чего впоследствии команда разработки и тестирования сталкивается с трудностями в дальнейшем, при развитии системы.

Как уже говорилось выше – модульное тестирование является нижней ступенью в иерархии тестирования и закладывает фундамент уверенности в работоспособности системы, из-за чего является крайне популярным относительно других видов тестирования и имеет большое количество инструментария.

**7.** **Вопросы для контроля знаний**

1. Что такое тестирование ПО?
2. Что такое качество программного обеспечения?
3. Каковы цели тестирования?
4. Что такое модульное тестирование?
5. Какие преимущества модульного тестирования?
6. Какие недостатки модульного тестирования?
7. Назовите инструменты тестирования для одного из языков программирования.

**Рекомундуемый список литературы:**

[1] Куликов, С.С. Тестирование программного обеспечения / С.С. Куликов. Минск, 2017.

[2] Kaner, С. Lesson Learning in Software Testing / Сem Kaner, James Bach, Bret Pettichord. Toronto, 2001.

[3] Модульное тестирование // Википедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Модульное\_тестирование#  
Приложения\_модульного\_тестирования – Дата доступа: 28.05.2020.

Преподаватель Н.А. Новосёлов

Приложение Ж

**ПЛАН-КОНСПЕКТ лекционного занятия**

**«Интеграционное тестирование» по дисциплине   
«Тестирование и отладка программного обеспечения»**

**направления специальности 2-40 01 01 «программное**

**обеспечение информационных технологий»**

Дисциплина: Тестирование и отладка программного обеспечения

Тема учебного занятия: «**Интеграционное тестирование» (2 часа)**

**Цели лекции**: Сформировать понятие об особенностях модульного тестирования. Дать понятие об организации модульного тестирования.

**Форма проведения учебного занятия:** лекция

**Задачи лекции**:

1. обучающая:
2. сформировать представление об особенностях интеграционного тестирования;
3. сформировать понятие об организации интеграционного тестирования;
4. сформировать понятие о сущности и способах организации интеграционного тестирования;
5. воспитательная:
6. способствовать воспитанию усидчивости, организованности, дисциплинированности;
7. развивающая:
8. способствовать развитию умений учащихся обобщать полученные знания, проводить анализ, синтез, сравнения, делать необходимые выводы.

**План лекции:**

1. Понятие тестирования, виды тестирования.
2. Понятие интеграционного тестирования.
3. Системы непрерывной интеграции.
4. Инструментарий тестирования.
5. Заключение.
6. Вопросы для контроля знаний.

**Используемые средства обучения:** мультимедийная презентация.

**Конспект лекции:**

**1. Понятие тестирования, виды тестирования**

**Тестирование программного обеспечения** — проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом. В более широком смысле, тестирование — это одна из техник контроля качества, включающая в себя активности по планированию работ (Test Management), проектированию тестов (Test Design), выполнению тестирования (Test Execution) и анализу полученных результатов (Test Analysis).

**Качество программного обеспечения (Software Quality)** — это совокупность характеристик программного обеспечения, относящихся к его способности удовлетворять установленные и предполагаемые потребности.

**Цели тестирования:**

* Повысить вероятность того, что приложение, предназначенное для тестирования, будет работать правильно при любых обстоятельствах.
* Повысить вероятность того, что приложение, предназначенное для тестирования, будет соответствовать всем описанным требованиям.
* Предоставление актуальной информации о состоянии продукта на данный момент.

**Этапы тестирования:**

1. Анализ продукта;
2. Работа с требованиями;
3. Разработка стратегии тестирования и планирование процедур контроля качества;
4. Создание тестовой документации;
5. Тестирование прототипа;
6. Основное тестирование;
7. Стабилизация;
8. Эксплуатация.

**Уровни тестирования:**

1. **Модульное тестирование (Unit Testing)**
2. **Интеграционное тестирование (Integration Testing)**
3. **Системное тестирование (System Testing)**
4. **Операционное тестирование (Release Testing)**
5. **Приемочное тестирование (Acceptance Testing)**

**2. Понятие интеграционного тестирования**

**Интеграционное тестирование (Integration testing)** — одна из фаз тестирования программного обеспечения, при которой отдельные программные модули объединяются и тестируются в группе. Обычно интеграционное тестирование проводится после модульного тестирования и предшествует системному тестированию.

Интеграционное тестирование в качестве входных данных использует модули, над которыми было проведено модульное тестирование, группирует их в более крупные множества, выполняет тесты, определённые в плане тестирования для этих множеств, и представляет их в качестве выходных данных и входных для последующего системного тестирования.

Целью интеграционного тестирования является проверка соответствия проектируемых единиц функциональным, приёмным и требованиям надежности. Тестирование этих проектируемых единиц — объединения, множества или группы модулей — выполняется через их интерфейс, с использованием тестирования «чёрного ящика».

**3. Системы непрерывной интеграции**

В целях автоматизации интеграционного тестирования применяются системы непрерывной интеграции (**Continuous Integration System**, CIS). Принцип действия подобных систем состоит в следующем:

1. CIS производит мониторинг системы контроля версий;
2. При изменении исходных кодов в репозитории производится обновление локального хранилища;
3. Выполняются необходимые проверки и модульные тесты;
4. Исходные коды компилируются в готовые выполняемые модули;
5. Выполняются тесты интеграционного уровня;
6. Генерируется отчет о тестировании.

Таким образом, автоматические интеграционные тесты выполняются сразу же после внесения изменений, что позволяет обнаруживать и устранять ошибки в короткие сроки.

**4. Инструментарий тестирования**

На рынке ПО существует множество систем непрерывной интеграции. Определить какой именно инструмент использовать – не самая простая задача. Стоит исходить из знания конкретного инструмента командой, а также технических особенностей каждой из систем.

Для наиболее понятного отображения ключевых характеристик и особенностей систем непрерывной интеграции зачастую применяют метод составления таблиц.

Одна из таких таблиц представлена на рисунке 1. Каждая система обозначена в крайнем левом столбце, далее же перечисляются характеристики систем, которые можно сравнить с другими, чтобы определить наиболее подходящий вариант.



Рисунок 1 – Сравнительная характеристика систем непрерывной интеграции

**5. Заключение**

Интеграционное тестирование, как связующее звено, между модульным тестированием и системным тестированием является ключевым аспектом качественного программного продукта, т.к. именно на этапе интеграционного тестирования становится понятно на сколько хорошо отдельно взятые модули могут работать в единой системе.

Также интеграционное тестирование позволяет определить соответствует ли проект системы тому состоянию, в котором она находится на момент проведения тестирования.

Зачастую именно на этапе интеграционного тестирования выявляется большинство проблем, связанных с интерфейсом пользователя и безопасностью системы.

**6. Вопросы для контроля знаний**

1. Что такое тестирование ПО?
2. Что такое качество программного обеспечения?
3. Каковы цели тестирования?
4. Что такое интеграционное тестирование?
5. Что такое CIS?
6. Для чего нужны CIS?

**Рекомундуемый список литературы:**

[1] Интеграционное тестирование // Википедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Интеграционное\_тестирование – Дата доступа: 28.05.2020.

[2] Comparison of continuous integration software // Wikipedia [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Comparison\_of\_  
continuous\_integration\_software – Дата доступа: 28.05.2020.

[3] Куликов, С.С. Тестирование программного обеспечения / С.С. Куликов. Минск, 2017.

[4] Kaner, С. Lesson Learning in Software Testing / Сem Kaner, James Bach, Bret Pettichord. Toronto, 2001.

Преподаватель Н.А. Новосёлов

Приложение И

**ПЛАН-КОНСПЕКТ лекционного занятия**

**«Системное тестирование» по дисциплине   
«тестирование и отладка программного обеспечения»**

**направления специальности 2-40 01 01 «программное**

**обеспечение информационных технологий»**

Дисциплина: Тестирование и отладка программного обеспечения

Тема учебного занятия: «**Системное тестирование» (2 часа)**

**Цель лекции**: Сформировать понятие об особенностях системного тестирования. Дать понятие об организации системного тестирования.

**Форма проведения учебного занятия:** лекция

**Задачи лекции**:

1. обучающая:
2. сформировать представление об особенностях системного тестирования;
3. сформировать понятие об организации системного тестирования;
4. сформировать понятие о сущности и способах организации системного тестирования;
5. воспитательная:
6. способствовать воспитанию усидчивости, организованности, дисциплинированности;
7. развивающая:
8. способствовать развитию умений учащихся обобщать полученные знания, проводить анализ, синтез, сравнения, делать необходимые выводы.

**План лекции:**

1. Понятие тестирования, виды тестирования.
2. Понятие системного тестирования.
3. Типы дефектов, выявляемых системным тестированием.
4. Подходы к системному тестированию.
5. Заключение.
6. Вопросы для контроля знаний.

**Используемые средства обучения:** мультимедийная презентация.

**Конспект лекции:**

**1. Понятие тестирования, виды тестирования**

**Тестирование программного обеспечения** — проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом. В более широком смысле, тестирование — это одна из техник контроля качества, включающая в себя активности по планированию работ (Test Management), проектированию тестов (Test Design), выполнению тестирования (Test Execution) и анализу полученных результатов (Test Analysis).

**Качество программного обеспечения (Software Quality)** — это совокупность характеристик программного обеспечения, относящихся к его способности удовлетворять установленные и предполагаемые потребности.

**Цели тестирования:**

* Повысить вероятность того, что приложение, предназначенное для тестирования, будет работать правильно при любых обстоятельствах.
* Повысить вероятность того, что приложение, предназначенное для тестирования, будет соответствовать всем описанным требованиям.
* Предоставление актуальной информации о состоянии продукта на данный момент.

**Этапы тестирования:**

1. Анализ продукта;
2. Работа с требованиями;
3. Разработка стратегии тестирования и планирование процедур контроля качества;
4. Создание тестовой документации;
5. Тестирование прототипа;
6. Основное тестирование;
7. Стабилизация;
8. Эксплуатация.

**Уровни тестирования:**

1. **Модульное тестирование (Unit Testing)**
2. **Интеграционное тестирование (Integration Testing)**
3. **Системное тестирование (System Testing)**
4. **Операционное тестирование (Release Testing)**
5. **Приемочное тестирование (Acceptance Testing)**

**2. Понятие системного тестирования**

**Системное тестирование – это тестирование программного обеспечения, выполняемое на полной, интегрированной системе, с целью проверки соответствия системы исходным требованиям, как функциональным, так и не функциональным.**

**Системное тестирование выполняется методом «Черного ящика», т.к. проверяемое множество является «внешними» сущностями, которые не требуют взаимодействия с внутренним устройством программы. Также выполнять его рекомендуется в окружении, максимально приближенном к окружению конечного пользователя.**

**3. Типы дефектов, выявляемых системным тестированием**

**Выполняя системное тестирование, можно обнаружить следующие типы дефектов:**

1. **Неправильное использование системных ресурсов.**
2. **Непредусмотренные комбинации пользовательских данных.**
3. **Проблемы с совместимостью окружения.**
4. **Непредусмотренные сценарии использования.**
5. **Несоответствие с функциональными требованиями.**
6. **Плохое удобство использования.**

Как можно увидеть по перечню дефектов доступных для обнаружения – системное тестирование является комплексным инструментов, который позволяет выявить как проблемы «внутренней» части системы, так и ее интерфейса.

Этот вид тестирования, в силу своей специфики является достаточно трудозатратным, т.к. не может быть автоматизирован в достаточной мере, из-за чего системное тестирование зачастую не используется в проектах с малым бюджетом.

**4. Подходы к системному тестированию**

Можно выделить следующие подходы к системному тестированию:

•  **На базе требований** **–** тестирование проводится в соответствии с функциональными или нефункциональными требованиями, для каждого из которых пишется testcase (тестовые прецеденты).

•  **На базе случаев использования** **– т**естирование происходит в соответствии с вариантами использования продукта, на основе которых создаются usercases (пользовательские прецеденты). Для каждого из данных пользовательских прецедентов создаются свои тестовые прецеденты.

Также к системному тестированию можно отнести **альфа-тестирование** и **бета-тестирование**.

**Альфа-тестирование** *(alpha testing) –* это вид приемочного тестирования, которое обычно проводится на поздней стадии разработки продукта и включает имитацию реального использования продукта штатными разработчиками либо командой тестировщиков. Обычно альфа тестирование заключается в систематической проверке всех функций программы с использованием техник тестирования «белого ящика» и «черного ящика».

**Бета-тестирование** *(beta testing)* – интенсивное использование почти готовой версии продукта с целью выявления максимального числа ошибок в его работе для их последующего устранения перед окончательным выходом (релизом) продукта на рынок, к массовому потребителю. Бета-тестирование представляет собой реально работающую версию программы с полным функционалом. И задача бета-тестов – оценить возможности и стабильность работы программы с точки зрения ее будущих пользователей.

**5. Заключение**

Системное тестирование, как наиболее комплексное, всеобъемлющее – имеет ряд уникальных особенностей, которые делают его крайне важным в общем процессе тестирования программного обеспечения.

Исходя из своих особенностей системное тестирование может выявить наибольшее количество проблем на различных уровнях системы, что является большим преимуществом в сравнении с другим видами тестирования.

Также, как и другие виды тестирования – системное тестирование имеет различные подходы, которые с различных сторон позволяют взглянуть на проблемы конечного продукта и выявить недостатки.

## 6. Вопросы для контроля знаний

1. Что такое тестирование ПО?
2. Что такое качество программного обеспечения?
3. Каковы цели тестирования?
4. Что такое системное тестирование?
5. Какие типы дефектов, выявляемых системным тестированием вы знаете?
6. Какие существуют подходы к системному тестированию?

**Рекомундуемый список литературы:**

[1] Системное тестирование // Википедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Системное\_тестирование – Дата доступа: 28.05.2020.

[2] Альфа- и Бета-тестирование // QATestLab [Электронный ресурс]. – Режим доступа: https://training.qatestlab.com/blog/technical-articles/alpha-beta-testing/ – Дата доступа: 28.05.2020.

[3] Куликов, С.С. Тестирование программного обеспечения / С.С. Куликов. Минск, 2017.

[4] Kaner, С. Lesson Learning in Software Testing / Сem Kaner, James Bach, Bret Pettichord. Toronto, 2001.

Преподаватель Н.А. Новосёлов

Приложение К

**ПЛАН-КОНСПЕКТ лабораторного занятия**

**Дисциплина:** Тестирование и отладка программного обеспечения

**Тема занятия:** разработка плана и проведение модульного тестирования ПО

**Цели занятия:** сформировать умение разрабатывать план и проводить модульное тестирование ПО.

**Задачи:** сформировать знания учащихся о модульном тестировании программного обеспечения.

**Вид занятия:** лабораторное занятие.

**Предварительная подготовка:** обучающимся рассказывается теоретическая часть занятия, выдаются задания для проведения лабораторной работы, вспомогательная литература.

**Этапы занятия:**

1. Организационный момент (приветствие, проверка присутствующих);
2. Представление темы и цели занятия;
3. Изложение материала о модульном тестировании, особенностях, преимуществах и недостатках;
4. Краткая демонстрация материала, связанного с темой;
5. Ответы на вопросы учащихся (при их наличии);
6. Выдача заданий;
7. Выполнение заданий обучающимися;
8. Подведение итогов работы: сдача отчетов, демонстрация выполненного задания учащимися, ответы на появившиеся вопросы.

**Конспект:**

**Модульное тестирование (Unit Testing)** — **проверяет функциональность и ищет дефекты в частях приложения, которые доступны и могут быть протестированы по-отдельности (модули программ, объекты, классы, функции и т.д.).**

**Преимущества модульного тестирования**

1. **Поощрение изменений** — **модульное тестирование позже позволяет программистам проводить**[**рефакторинг**](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3)**, будучи уверенными, что модуль по-прежнему работает корректно (**[**регрессионное тестирование**](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D1%80%D0%B5%D1%81%D1%81%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)**). Это поощряет программистов к изменениям кода, поскольку достаточно легко проверить, что код работает и после изменений;**
2. **Упрощение интеграции** — **модульное тестирование помогает устранить сомнения по поводу отдельных модулей и может быть использовано для подхода к тестированию «снизу-вверх»: сначала тестируя отдельные части программы, а затем программу в целом;**
3. **Документирование кода** — **модульные тесты можно рассматривать как «живой документ» для тестируемого**[**класса**](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5))**. Клиенты, которые не знают, как использовать данный класс, могут использовать юнит-тест в качестве примера;**
4. **Отделение интерфейса от реализации** — **поскольку некоторые классы могут использовать другие классы, тестирование отдельного класса часто распространяется на связанные с ним. Например, класс пользуется базой данных; в ходе написания теста программист обнаруживает, что тесту приходится взаимодействовать с базой. Это ошибка, поскольку тест не должен выходить за границу класса. В результате разработчик абстрагируется от соединения с базой данных и реализует этот интерфейс, используя свой собственный mock-объект. Это приводит к менее связанному коду, минимизируя зависимости в системе.**

**Недостатки модульного тестирования**

1. **Сложный код** — тестирование программного обеспечения — комбинаторная задача. Например, каждое возможное значение булевской переменной потребует двух тестов: один на вариант TRUE, другой — на вариант FALSE. В результате на каждую строку исходного кода потребуется 3−5 строк тестового кода.
2. **Результат известен лишь приблизительно** — бизнес-приложения зачастую работают с конечными и счётными множествами, научные — с континуальными. Поэтому сложно подобрать тесты для каждой из ветвей программы, сложно сказать, верен ли результат, выдерживается ли точность, и т. д. А во многих случаях качество моделирования определяется «на глаз», и последний результат записывается как «опорный». Если найдено расхождение, новый результат проверяют вручную и выясняют, какой качественнее: старый или новый.
3. **Код, взаимодействующий с системой** — код, взаимодействующий с портами, таймерами и прочими «нестабильными» частями системы, крайне сложно проверить в изолированном окружении.
4. **Ошибки интеграции и производительности** — при выполнении юнит-тестов происходит тестирование каждого из модулей по отдельности. Это означает, что ошибки интеграции, системного уровня, функций, исполняемых в нескольких модулях, не будут определены. Кроме того, данная технология бесполезна для проведения тестов на производительность. Таким образом, модульное тестирование более эффективно при использовании в сочетании с другими методиками тестирования.
5. **При общей низкой культуре программирования** — для получения выгоды от модульного тестирования требуется строго следовать технологии тестирования на всём протяжении процесса разработки программного обеспечения. Нужно хранить не только записи обо всех проведённых тестах, но и обо всех изменениях исходного кода во всех модулях. С этой целью следует использовать систему контроля версий ПО. Таким образом, если более поздняя версия ПО не проходит тест, который был успешно пройден ранее, будет несложным сверить варианты исходного кода и устранить ошибку. Также необходимо убедиться в неизменном отслеживании и анализе неудачных тестов. Игнорирование этого требования приведёт к лавинообразному увеличению неудачных тестовых результатов.
6. **Проблемы с объектами-заглушками** — за исключением простейших случаев, тестируемый объект должен взаимодействовать с другими объектами. Этих «товарищей по взаимодействию» — объекты-заглушки — делают предельно простыми: либо крайне упрощёнными (память вместо базы данных), либо рассчитанными на конкретный тест и механически повторяющими сессию обмена. Проблемы могут возникать при смене протокола обмена, в таком случае объекты-заглушки должны отвечать новым требованиям протокола.

**Инструментарий**

В качестве инструмента для выполнения работы необходимо использовать PHPUnit.

PHPUnit – утилита для модульного тестирования PHP скриптов. Является наиболее распространённой, простой в понимании и хорошей документацией (https://phpunit.readthedocs.io/ru/latest/).

**Задание к работе:**

1. Установить PHPUnit на рабочую машину (если его нет в наличии);
2. Запустить тестируемый код на PHP, для проверки работоспособности. Создать экземпляр класса и вызвать методы для тестирования корректности работы.
3. Написать полный набор модульных тестов с использованием PHPUnit.

**Вспомогательная литература:**

[1] Куликов, С.С. Тестирование программного обеспечения / С.С. Куликов. Минск, 2017.

[2] Kaner, С. Lesson Learning in Software Testing / Сem Kaner, James Bach, Bret Pettichord. Toronto, 2001.

**Рекомундуемый список литературы:**

[1] Руководство по PHPUnit // PHPUnit [Электронный ресурс]. – Режим доступа: https://phpunit.readthedocs.io/ru/latest/ – Дата доступа: 28.05.2020.

[2] Модульное тестирование // Википедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Модульное\_тестирование#  
Приложения\_модульного\_тестирования – Дата доступа: 28.05.2020.

Преподаватель Н.А. Новосёлов

Приложение Л

**Инструкция для лабораторного занятия**

Министерство образования Республики Беларусь

Учреждение образования

Филиал Белорусского государственного университета

информатики и радиоэлектроники

«Минский радиотехнический колледж»

Дисциплина «Тестирование и отладка программного обеспечения»

**Инструкция**

по выполнению лабораторной работы

«Разработка плана и проведение модульного тестирования ПО»

Минск

2020

Составитель: Новосёлов Н.А.\_\_

Обсуждена и одобрена на заседании методической комиссии.

Протокол от \_\_\_\_\_\_\_\_\_\_\_\_\_ № \_\_\_\_\_\_

**Лабораторная работа № 5**

**Тема работы: «Разработка плана и проведение модульного тестирования ПО»**

**1. Цель работы:** изучить модульное тестирование на примере работы с PHP и PHPUnit.

**2. Задачи:**

1. обучающая:
2. сформировать знания учащихся о модульном тестировании;
3. воспитательная:
4. способствовать воспитанию усидчивости, организованности, дисциплинированности;
5. развивающая:
6. способствовать развитию умений учащихся обобщать полученные знания, проводить анализ, синтез, сравнения, делать необходимые выводы;
7. способствовать развитию коммуникативных навыков, логического мышления, самостоятельности, ответственности.

**3. Задание для самостоятельной работы:**

3.1. Установить PHPUnit на рабочую машину (если его нет в наличии).

3.2 Запустить тестируемый код на PHP, для проверки работоспособности. Создать экземпляр класса и вызвать методы для тестирования корректности работы.

3.3. Написать полный набор модульных тестов с использованием PHPUnit.

**4. Оснащение работы:** персональный компьютер, установленный интерпретатор PHP исполняемый Apache или Nginx, PHPUnit, веб-браузер, редактор кода, исходный тестируемый код.

**5. Основные теоретические сведения:**

**Модульное тестирование (Unit Testing)** — **проверяет функциональность и ищет дефекты в частях приложения, которые доступны и могут быть протестированы по-отдельности (модули программ, объекты, классы, функции и т.д.).**

**Преимущества модульного тестирования**

1. **Поощрение изменений** — **модульное тестирование позже позволяет программистам проводить**[**рефакторинг**](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3)**, будучи уверенными, что модуль по-прежнему работает корректно (**[**регрессионное тестирование**](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D1%80%D0%B5%D1%81%D1%81%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)**). Это поощряет программистов к изменениям кода, поскольку достаточно легко проверить, что код работает и после изменений;**
2. **Упрощение интеграции** — **модульное тестирование помогает устранить сомнения по поводу отдельных модулей и может быть использовано для подхода к тестированию «снизу-вверх»: сначала тестируя отдельные части программы, а затем программу в целом;**
3. **Документирование кода** — **модульные тесты можно рассматривать как «живой документ» для тестируемого**[**класса**](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5))**. Клиенты, которые не знают, как использовать данный класс, могут использовать юнит-тест в качестве примера;**
4. **Отделение интерфейса от реализации** — **поскольку некоторые классы могут использовать другие классы, тестирование отдельного класса часто распространяется на связанные с ним. Например, класс пользуется базой данных; в ходе написания теста программист обнаруживает, что тесту приходится взаимодействовать с базой. Это ошибка, поскольку тест не должен выходить за границу класса. В результате разработчик абстрагируется от соединения с базой данных и реализует этот интерфейс, используя свой собственный mock-объект. Это приводит к менее связанному коду, минимизируя зависимости в системе.**

**Недостатки модульного тестирования**

1. **Сложный код** — тестирование программного обеспечения — комбинаторная задача. Например, каждое возможное значение булевской переменной потребует двух тестов: один на вариант TRUE, другой — на вариант FALSE. В результате на каждую строку исходного кода потребуется 3−5 строк тестового кода.
2. **Результат известен лишь приблизительно** — бизнес-приложения зачастую работают с конечными и счётными множествами, научные — с континуальными. Поэтому сложно подобрать тесты для каждой из ветвей программы, сложно сказать, верен ли результат, выдерживается ли точность, и т. д. А во многих случаях качество моделирования определяется «на глаз», и последний результат записывается как «опорный». Если найдено расхождение, новый результат проверяют вручную и выясняют, какой качественнее: старый или новый.
3. **Код, взаимодействующий с системой** — код, взаимодействующий с портами, таймерами и прочими «нестабильными» частями системы, крайне сложно проверить в изолированном окружении.
4. **Ошибки интеграции и производительности** — при выполнении юнит-тестов происходит тестирование каждого из модулей по отдельности. Это означает, что ошибки интеграции, системного уровня, функций, исполняемых в нескольких модулях, не будут определены. Кроме того, данная технология бесполезна для проведения тестов на производительность. Таким образом, модульное тестирование более эффективно при использовании в сочетании с другими методиками тестирования.
5. **При общей низкой культуре программирования** — для получения выгоды от модульного тестирования требуется строго следовать технологии тестирования на всём протяжении процесса разработки программного обеспечения. Нужно хранить не только записи обо всех проведённых тестах, но и обо всех изменениях исходного кода во всех модулях. С этой целью следует использовать систему контроля версий ПО. Таким образом, если более поздняя версия ПО не проходит тест, который был успешно пройден ранее, будет несложным сверить варианты исходного кода и устранить ошибку. Также необходимо убедиться в неизменном отслеживании и анализе неудачных тестов. Игнорирование этого требования приведёт к лавинообразному увеличению неудачных тестовых результатов.
6. **Проблемы с объектами-заглушками** — за исключением простейших случаев, тестируемый объект должен взаимодействовать с другими объектами. Этих «товарищей по взаимодействию» — объекты-заглушки — делают предельно простыми: либо крайне упрощёнными (память вместо базы данных), либо рассчитанными на конкретный тест и механически повторяющими сессию обмена. Проблемы могут возникать при смене протокола обмена, в таком случае объекты-заглушки должны отвечать новым требованиям протокола.

**Инструментарий**

В качестве инструмента для выполнения работы необходимо использовать PHPUnit.

PHPUnit – утилита для модульного тестирования PHP скриптов. Является наиболее распространённой, простой в понимании и хорошей документацией (https://phpunit.readthedocs.io/ru/latest/).

**6. Порядок выполнения работы:**

6.1. Запускаем локальный сервер Apache или Nginx

6.2. Создаем файл Email.php и помещаем туда следующий код:

<?php declare(strict\_types=1);

final class Email

{

private $email;

private function \_\_construct(string $email)

{

$this->ensureIsValidEmail($email);

$this->email = $email;

}

public static function fromString(string $email): self

{

return new self($email);

}

public function \_\_toString(): string

{

return $this->email;

}

private function ensureIsValidEmail(string $email): void

{

if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

throw new InvalidArgumentException(

sprintf(

'"%s" is not a valid email address',

$email

)

);

}

}

}

6.3. Создаем файл index.php. Создаем экземпляр класса Email и используем все его методы.

6.4. Запускаем файл index.php и проверяем работоспособность.

6.5. Создаем папку test внутри директории проекта, в ней создаем файл EmailTest.php.

6.6. Изучаем исходный код Email.php и планируем какие тест-кейсы должны покрывать этот код.

6.7. Реализуем тесты в коде файла EmailTest.php

6.8. Запускаем тесты (зависит от способа установки PHPUnit) командой phpunit tests

6.9. Тесты не должны вывести ошибок. Должен отобразится таймер времени и затраченная память. Ниже на строку выводится статистика с количеством тестов и их результатами.

**7. Форма отчета о работе:**

*Лабораторная работа № \_\_\_*

*Номер учебной группы \_\_\_\_\_\_\_\_*

*Фамилия, инициалы обучающегося \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Дата выполнения работы \_\_\_\_\_\_\_\_\_\_\_\_*

*Тема работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Цель работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Задание: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Оснащение работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Результаты выполнения работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Отчет по лабораторной работе оформляется в программной оболочке Microsoft Word (других редакторах) и предоставляется преподавателю в отпечатанном виде на листах формата А4.

Отчет должен содержать:

1. Название и цель лабораторной работы;
2. Скриншоты о проделанной работе;
3. Заключения и выводы.

**8. Контрольные вопросы и задания:**

1. Что такое модульное тестирование?
2. Какие преимущества у модульного тестирования?
3. Какие недостатки у модульного тестирования?
4. Что такое PHPUnit?

**Рекомундуемый список литературы:**

[1] Куликов, С.С. Тестирование программного обеспечения / С.С. Куликов. Минск, 2017.

[2] Kaner, С. Lesson Learning in Software Testing / Сem Kaner, James Bach, Bret Pettichord. Toronto, 2001.

Приложение М

**Презентации лекционных занятий**

**Ведомость документов**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Обозначение | | | | | Наименование | | | | | Дополнительные сведения | | |
|  | | | | | Текстовые документы | | | | |  | | |
|  | | | | |  | | | | |  | | |
| БГУИР КР 1-08 01 01-07 13 ПЗ | | | | | Пояснительная записка | | | | | 84 с. | | |
|  | | | | |  | | | | |  | | |
|  | | | | |  | | | | |  | | |
|  | | | | | Электронные документы | | | | |  | | |
|  | | | | |  | | | | |  | | |
| БГУИР КР 1-08 01 01-07 13 Э | | | | | Мультимедиапрезентации для прове- | | | | | CD-R 1 шт. | | |
|  | | | | | дения лекционных занятий | | | | |  | | |
|  | | | | |  | | | | |  | | |
|  | | | | |  | | | | |  | | |
|  | | | | |  | | | | |  | | |
|  | | | | |  | | | | |  | | |
|  | | | | |  | | | | |  | | |
|  | | | | |  | | | | |  | | |
|  | | | | |  | | | | |  | | |
|  | | | | |  | | | | |  | | |
|  | | | | |  | | | | |  | | |
|  | | | | |  | | | | |  | | |
|  | | | | |  | | | | |  | | |
|  | | | | |  | | | | |  | | |
|  | | | | |  | | | | |  | | |
|  | | | | |  | | | | |  | | |
|  |  |  |  |  | | БГУИР КР 1-08 01 01-07 13 Д1 | | | | | | |
|  |  |  |  |  | |
|  |  |  |  |  | |
| Изм. | Л. | № докум. | Подп. | Дата | | Методическая разработка раздела «Организация тестирования программного обеспечения» по дисциплине «Тестирование и отладка программного обеспечения» уровня среднего специального образования |  | | | | Лист | Листов |
| Разраб. | | Новосёлов Н.А. |  | 11.05.20 | |  |  |  | | 83 | 83 |
| Пров. | | Парафиянович Т.А. |  | 11.05.20 | | Кафедра ИРТ  гр. 844691 | | | | | |
|  | |  |  |  | |
|  | |  |  |  | |
|  | |  |  |  | |