# Аннотация

Дипломный проект посвящен разработке и реализации программного продукта «Платформа для обучения программированию на Java».

В первом разделе проведен сравнительный анализ аналогичных программных продуктов предназначенных для обучения, описано назначение и сформулированы требования к разрабатываемому программному продукту.

# СОдержание

# Перечень условных обозначений

ПО – программное обеспечение

АСУ – автоматизированная система управление

УМК – учебно-методический комплекс

ЭУМК – электронный учебно-методический комплекс

# введение

В современном мире, по причине быстрого развития информационных технологий, происходит их повсеместное внедрение во все сферы человеческой жизни.

# 1 Описвние предметной области, Целей и назначения проекта

## 1.1 Цели разработки

Платформа предназначена для повышения доступности и качества образовательной деятельности в рамках учебного заведения, а также призвана облегчить работу преподавательской деятельности.

## 1.2 Основание для разработки

Разработка системы осуществляется на основании темы к дипломному проекту в соответствии с документацией деканата факультета «Электросвязи» Белорусской государственной академии связи и регламентируется Уставом Белорусской государственной академии связи.

## 1.3 Назначение

Назначение разрабатываемого программного продукта заключается в передаче студентам теоретических знаний концепции объектно-ориентированного программирования, практических навыков по разработке объектно-ориентированных программ на одном из популярных языков программирования – Java.

## 1.4 Описание предметной области

Опишем понятия, которые входят в предметную область выбранной темы.

**Программное обеспечение (ПО)** представляет собой программу, либо совокупность программ, предназначенных для управления компьютером.

Программное обеспечение в зависимости от назначения разделяется на: системное ПО, прикладное ПО и инструментальное ПО.

Системное программное обеспечения включает в себя программы, обеспечивающие функционирование компьютера и компьютерных сетей. Данное ПО позволяет пользователю осуществлять управление компьютером и компьютерной сетью, а также обеспечивает работоспособность других программ. В качестве примеров системного ПО можно привести операционные системы, архиваторы, файловые менеджеры, утилиты и другие.

Программы, предназначенные для решения задач определенной предметной области, относятся к прикладному программному обеспечению.

Прикладное ПО принято разделять на:

– программы общего назначения (необходимы для работы любого пользователя);

– программы специального назначения (требуются для использования профессионалами в различных сфера деятельности);

– компьютерные игры.

Примерами прикладных программ являются текстовые редакторы, графические редакторы и другие.

Инструментальное программное обеспечение предназначения для проектирования, разработки и сопровождения программ.

Различают такие виды инструментального ПО как: компиляторы, интерпретаторы, отладчики, SDK, интегрированные среды разработки и другие.

**Автоматизированные системы управления (АСУ)** – это комплекс программных и аппаратных средств, предназначенный для управления различными процессами в рамках технологического процесса, предприятия, производства. АСУ применяются в различных отраслях промышленности, энергетике, транспорте и т. п.

Наиболее важной задачей автоматизированных систем управления является повышение эффективности управления объектом на основе роста производительности труда и совершенствования методов планирования процесса управления.

Различают автоматизированные системы управления объектами (технологическими процессами – АСУТП, предприятием – АСУП, отраслью – ОАСУ) и функциональные автоматизированные системы, например, проектирование плановых расчётов, материально-технического снабжения и т.д.

**Java** является универсальным языком программирования, который основан на концепции объектно-ориентированного программирования. Он предназначен для того, чтобы позволить разработчикам приложений «писать один раз, выполнять где угодно» (WORA), что означает, что скомпилированный код Java может работать на всех платформах, поддерживающих Java, без необходимости перекомпиляции.

**Учебная программа** (Учебный план) учреждения образования – созданный в рамках системы обучения документ, определяющий содержание и количество знаний, умений и навыков, предназначенных к обязательному усвоению по той или иной учебной дисциплине, распределение их по темам, разделам и периодам обучения.

**Учебно-методический комплекс** (УМК) дисциплины – стандартное название для совокупности учебно-методической документации, средств обучения и контроля, разрабатываемых в учреждениях образования для каждой дисциплины. УМК должен включать полную информацию, достаточную для прохождения дисциплины. УМК предназначены для обеспечения открытости образовательного процесса и должны быть доступны любому желающему.

**Электронный учебно-методический комплекс** (ЭУМК) дисциплины – продукт, имеющий электронный формат, который используется в учебном процессе учебного заведения в рамках дисциплины.

**Контроль знаний** – оценка знаний учащихся по истечении конкретного периода учебного года (четверти, полугодия), позволяющая определить качество и объем усвоенных знаний.

**Журнал успеваемости** – документ, позволяющий участникам учебного процесса вести учет о текущих и итоговых оценках.

В качестве предметной области выбрано учреждение образования «Белорусская государственная академия связи», которое занимается обучением будущих специалистов в области телекоммуникаций, а точнее кафедра программного обеспечения сетей телекоммуникаций (ПОСТ) [1].

Кафедра является профилирующей и осуществляет подготовку специалистов на уровне высшего образования по направлению специальности 1-45 01 02-03 «Инфокоммуникационные системы (сопровождение программного обеспечения)», (квалификация специалиста – инженер по инфокоммуникационным системам) [1].

Кафедра осуществляет подготовку специалистов на уровне среднего специального образования по специальности 2-45 01 33 «Сети телекоммуникаций», специализации 2-45 01 33 02 «Программное обеспечение сетей телекоммуникаций» (квалификация специалиста – техник по телекоммуникациям) [1].

На уровне высшего образования преподается дисциплина «Конструирование программ и языки программирования», цель которой является приобретение студентами теоретических знаний концепции объектно-ориентированного программирования, практических навыков по разработке объектно-ориентированных программ, а также умений объектно-ориентированных программ [2].

В рамках дисциплины предусматривается изучение языка Java SE, аспекты применения библиотек классов языка Java, включая файлы, коллекции, сетевые и многопоточные приложения, а также взаимодействие с XML [2].

# 2 Выбор жизненного цикла проекта, анализ предметной области, выявление требований к системе

## 2.1 Выбор модели жизненного цикла проекта

Существуют различные схемы классификации проектов, связанных с разработкой программных средств и систем.

В **ГОСТ Р ИСО/МЭК ТО 12182–2002 – Информационная технология. Классификация программных средств** [3] – приведена схема классификации ПС по 16 видам, каждый из которых подразделяется на классы. Данная классификация имеет общий характер и в целом не может использоваться для обоснования выбора модели ЖЦ ПС и систем [4].

Институтом качества программного обеспечения SQI (Software Quality Institute, США) специально для выбора модели ЖЦ предложена схема классификации проектов по разработке ПС и систем [5]. Основу данной классификации составляют четыре категории критериев. По каждому критерию проекты подразделяются на два альтернативных класса [4].

Критерии классификации проектов, предложенные Институтом SQI, объединены в следующие категории.

*1. Характеристики требований к проекту*

Критерии данной категории классифицируют проекты в зависимости от свойств требований пользователя (заказчика) к разрабатываемой системе или программному средству. Например, известны ли требования к началу проекта, сложны ли они, будут ли они изменяться.

*2. Характеристики команды разработчиков*

Чтобы иметь возможность пользоваться критериями данной категории классификации проектов, состав команды разработчиков необходимо сформировать до выбора модели ЖЦ. Характеристики команды разработчиков играют важную роль при выборе модели ЖЦ, поскольку разработчики несут основную ответственность за успешную реализацию проекта. В первую очередь следует учитывать квалификацию разработчиков, их знакомство с предметной областью, инструментальными средствами разработки и т.п.

*3. Характеристики пользователей (заказчиков)*

Чтобы иметь возможность пользоваться критериями данной категории классификации проектов, до выбора модели ЖЦ необходимо определить возможную степень участия пользователей (заказчиков) в процессе разработки и их взаимосвязь с командой разработчиков на протяжении проекта. Это важно, поскольку отдельные модели ЖЦ требуют усиленного участия пользователей в процессе разработки. Выбор таких моделей при отсутствии реальной возможности пользователей участвовать в проекте приведет к существенному снижению качества результатов разработки.

*4. Характеристики типов проектов и рисков*

В некоторых моделях в достаточно высокой степени предусмотрено управление рисками. В других моделях управление рисками вообще не предусматривается. Поэтому при выборе модели ЖЦ следует учитывать реальные риски проекта, критичность и сложность продуктов разработки.

Для выбора подходящей к условиям конкретного проекта модели ЖЦ ПС и систем Институтом качества программного обеспечения SQI рекомендуется использовать специальную процедуру [5]. Данная процедура базируется на применении четырех таблиц вопросов. Примеры вопросов приведены в таблицах A.1-A.4 приложения А.

Для моего проекта наиболее важными вопросами из таблиц А.1-А.4 являются:1, 2, 3, 6, 7 – вопросы из таблицы А.1; все вопросы из таблицы А.2; 3, 4 – вопросы из таблицы А.3; 3, 5, 6, 7, 8, 10 – вопросы из таблицы А.4.

Таблица 2.1 – Итоги выбора жизненного цикла

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Количество отмеченных вопросов | Каскадная | Vобразная | RAD | Инкре-  ментная | Быстрого  прототипи-  рования | Эволюци-  онная |
| Важные | 14 | 13 | 14 | 8 | 7 | 5 |
| Не важные | 2 | 2 | 5 | 1 | 6 | 4 |
| Итого | 16 | 15 | 19 | 9 | 13 | 9 |

Исходя из итогов таблицы 2.1 лидирует модель жизненного цикла RAD по количеству важных вопросов и суммарному количеству вопросов. Так как к процессу разработки нет ограничений, выбираем **модель жизненного цикла RAD**.

## 1.5 Анализ предметной области

В качестве стратегии

## 1.4 Требования к системе

Как было указано выше ­– платформа предназначена для обучения студентов, это говорит о том, что необходимо учитывать план обучения студентов и учебно-методический комплекс, по которому проходит процесс обучения в учреждении образования. В процесс

# заключение

# список использовыных источников

1 УО Белорусская государственная академия связи [Электронный ресурс]. – Режим доступа: <http://bsac.by/> – Дата доступа: 16.05.2019.

2 Рябычина, О. П. Конструирование программ и языки программирования. Учебная программа учреждения высшего образования по учебной дисциплине для специальности: 1-45 01 02 Инфокоммуникационные системы (по направлениям) по направлению: 1-45 01 02-03 Инфокоммуникационные системы (сопровождение программного обеспечения) / О. П. Рябычина. – Минск : УО БГАС, 2015. – 13 с.

3 ГОСТ Р ИСО/МЭК ТО 12182 – 2002. Информационная технология. Классификация программных средств.

4 Бахтизин В. В. Технология разработки программного обеспечения / В. В. Бахтизин, Л. А. Глухова. – Минск : БГУИР, 2010. – 267 с.

5 Фатрелл, Р. Управление программными проектами: достижение оптимального качества при минимуме затрат / Р. Фатрелл, Д. Шафер, Л. Шафер. – М. : Вильямс, 2003.

# приложение а

Таблица А.1 – Выбор модели жизненного цикла на основе характеристик требований

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии  категории  требований | Каскадная | V-образная | RAD | Инкре-  ментная | Быстрого  прототипи-  рования | Эволюци-  онная |
| 1 | Являются ли требования к  проекту легко определи-  мыми и реализуемыми? | Да | Да | Да | Нет | Нет | Нет |
| 2 | Могут ли требования быть  сформулированы в начале  ЖЦ? | Да | Да | Да | Да | Нет | Нет |
| 3 | Часто ли будут изменяться  требования на протяжении  ЖЦ? | Нет | Нет | Нет | Нет | Да | Да |
| 4 | Нужно ли демонстрировать  требования с целью их оп-  ределения? | Нет | Нет | Да | Нет | Да | Да |
| 5 | Требуется ли проверка  концепции программного  средства или системы? | Нет | Нет | Да | Нет | Да | Да |
| 6 | Будут ли требования изме-  няться или уточняться с  ростом сложности системы  (программного средства) в  ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 7 | Нужно ли реализовать ос-  новные требования на ран-  них этапах разработки? | Нет | Нет | Да | Да | Да | Да |

Таблица А.2 – Выбор модели жизненного цикла на основе характеристик команды разработчиков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии  категории  требований | Каскадная | V-образная | RAD | Инкре-  ментная | Быстрого  прототипи-  рования | Эволюци-  онная |
| 1 | Являются ли проблемы  предметной области проек-  та новыми для большинст-  ва разработчиков? | Нет | Нет | Нет | Нет | Да | Да |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии  категории  требований | Каскадная | V-образная | RAD | Инкре-  ментная | Быстрого  прототипи-  рования | Эволюци-  онная |
| 2 | Являются ли инструмен-  тальные средства, исполь-  зуемые в проекте, новыми  для большинства разработ-  чиков? | Да | Да | Нет | Нет | Нет | Да |
| 3 | Изменяются ли роли участ-  ников проекта на протяже-  нии ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 4 | Является ли структура про-  цесса разработки более  значимой для разработчи-  ков, чем гибкость? | Да | Да | Нет | Да | Нет | Нет |
| 5 | Важна ли легкость распре-  деления человеческих ре-  сурсов проекта? | Да | Да | Да | Да | Нет | Нет |
| 6 | Приемлет ли команда раз-  работчиков оценки, про-  верки, стадии разработки? | Да | Да | Нет | Да | Да | Да |

Таблица А.3 – Выбор модели жизненного цикла на основе характеристик коллектива пользователей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии  категории  требований | Каскадная | V-образная | RAD | Инкре-  ментная | Быстрого  прототипи-  рования | Эволюци-  онная |
| 1 | Будет ли присутствие поль-  зователей ограничено в ЖЦ  разработки? | Да | Да | Нет | Да | Нет | Нет |
| 2 | Будут ли пользователи оце-  нивать текущее состояние  программного продукта  (системы) в процессе раз-  работки? | Нет | Нет | Нет | Да | Да | Да |
| 3 | Будут ли пользователи во-  влечены во все фазы ЖЦ  разработки? | Нет | Нет | Да | Нет | Да | Нет |
| 4 | Будет ли заказчик отслежи-  вать ход выполнения про-  екта? | Нет | Нет | Нет | Нет | Да | Да |

Таблица А.4 – Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии  категории  требований | Каскадная | V-образная | RAD | Инкре-  ментная | Быстрого  прототипи-  рования | Эволюци-  онная |
| 1 | Разрабатывается ли в про-  екте продукт нового для ор-  ганизации направления? | Нет | Нет | Нет | Да | Да | Да |
| 2 | Будет ли проект являться  расширением существую-  щей системы? | Да | Да | Да | Да | Нет | Нет |
| 3 | Будет ли проект крупно-  или среднемасштабным? | Нет | Нет | Нет | Да | Да | Да |
| 4 | Ожидается ли длительная  эксплуатация продукта? | Да | Да | Нет | Да | Нет | Да |
| 5 | Необходим ли высокий  уровень надежности про-  дукта проекта? | Нет | Да | Нет | Да | Нет | Да |
| 6 | Предполагается ли эволю-  ция продукта проекта в те-  чение ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 7 | Велика ли вероятность из-  менения системы (продук-  та) на этапе сопровожде-  ния? | Нет | Нет | Нет | Да | Да | Да |
| 8 | Является ли график сжа-  тым? | Нет | Нет | Да | Да | Да | Да |
| 9 | Предполагается ли повтор-  ное использование компо-  нентов? | Нет | Нет | Да | Да | Да | Да |
| 10 | Являются ли достаточными  ресурсы (время, деньги, ин-  струменты, персонал)? | Нет | Нет | Нет | Нет | Да | Да |