**Информационная система для организации учебной практики для студентов, обучающихся программированию.**

# Введение

В современном мире спрос на квалифицированных специалистов в области информационных технологий стремительно растёт [1]. Одним из способов получения квалификации в этой сфере является обучение в высшем учебном заведении. Занятия в таких учебных заведениях подразделяются на теоретические и практические. Процесс организации/проведения практических занятий у студентов, обучающихся программированию, представляет собой:

* выдачу заданий студентам;
* выполнение полученных заданий;
* сбор и проверку выполненных заданий преподавателем;
* коммуникацию преподавателя и студента.

Организация этих шагов в единую информационную систему позволит оптимизировать процесс выдачи и сбора заданий, процесс коммуникации, а также автоматизировать проверку загруженных решений.

**Объектом исследования** в данном проекте является процесс организации учебной практики для студентов, изучающих программирование. **Предметом** — автоматизация процесса взаимодействия преподавателя со студентом в рамках выполнения практических заданий.

**Цель проекта** — проектирование программной системы для осуществления выдачи, сбора и автоматизированной проверки практических заданий с реализацией рабочих мест преподавателя и студента, а также модуля автоматического тестирования.

Поставленная цель требует выполнения ряда **задач**:

* провести анализ процесса взаимодействия преподавателя со студентом в рамках выполнения практических заданий;
* провести анализ программ-аналогов, выделить их достоинства и недостатки;
* провести сбор и анализ требований к программному продукту;
* спроектировать архитектуру реализуемого программного комплекса и структуру базы данных;
* реализовать рабочие места преподавателя и студента, и модуль автоматического тестирования;
* произвести тестирование разработанных модулей.

# АНАЛИЗ ТРЕБОВАНИЙ

## 1.1. Описание процесса проведения практических занятий по программированию

Для понимания проблематики взаимодействия преподавателя и студента в рамках практических занятий и обоснования актуальности разработки информационной системы, необходимо проанализировать предметную область и рассмотреть программы-аналоги.

### 1.1.1. Основные сведения и термины

Приведём основные термины этой предметной области:

Практическая работа — вид самостоятельной работы, в процессе которой учащиеся применяют получение знания на практике.

Задание — перечень требований, выдвигаемых преподавателем к результату выполненной студентом работы.

Проверка практической работы — процесс проверки преподавателем правильности решения студента, а также личный разговор преподавателя со студентом для оценки теоретических знаний учащегося.

Правильное решение — решение, удовлетворяющее всем заданным требованиям.

Результат выполнения самостоятельной работы — программный код, написанный студентом, и усвоенный теоретический материал.

### 1.1.2. Процесс взаимодействия преподавателя и студента в рамках практических занятий

Рассмотрим процесс взаимодействия преподавателя и студента в рамках практических занятий. Алгоритм данного взаимодействия представлен на рисунке 1.1.

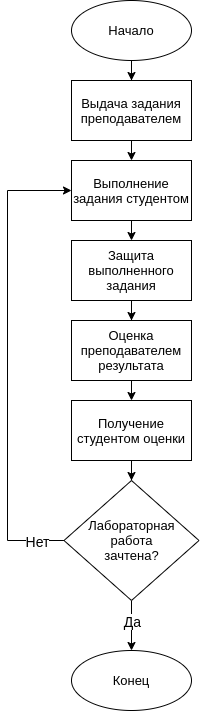


Рис 1.1. «Алгоритм взаимодействия преподавателя и студента в рамках практических занятий»

В рассматриваемом алгоритме будут принимать участие преподаватель и студент.

На первом шаге преподаватель выдаёт задание и методический материал студентам. Затем обучаемые приступают к выполнению выданного практического задания.

По окончании выполнения работы, она сдаётся преподавателю на проверку. Преподаватель проводит тестирование решения студента, а также оценивает качество работы. При необходимости, студенту необходимо защитить свою работу, ответив на блок теоретических вопросов.

После этого преподаватель подводит итог. При успешном выполнении задание считается принятым. В противном случае студент должен исправить ошибки и недочёты, и повторно предоставить решение на проверку.

### 1.1.3. Перечень проблем

Выделим главные проблемы, которые встречаются в процессе практических занятий и рассмотрим их отдельно.

Одной из основных проблем является децентрализованная выдача преподавателем заданий студентам: в настоящее время не существует единого центра для хранения и выдачи практических заданий. В силу этого каждый преподаватель подбирает собственные способы и информационно-технические решения.

Подобное значительно снижает эффективность взаимодействия.

Следующая ключевая проблема — трудоемкость сдачи студентом выполненной работы и дальнейшей проверки преподавателем результата. На данный момент студент вынужден лично демонстрировать результат выполненной работы, после чего преподаватель вручную проверяет реализацию программы и проводит ряд тестовых запусков.

Выполнение тестирования преподавателем программы вручную — трудоёмко и требует большого количества времени. Также в силу ограниченности по времени, преподаватель не может провести тщательное тестирование программы на полном перечне входных данных.

Заключительной проблемой является необходимость личного взаимодействия для обратной связи преподавателя со студентом в рамках сдачи практического задания.

## 1.2. Обзор программ-аналогов

При организации процесса обучения студента, обучающегося программированию, возможно использование комплекса различного программного обеспечения. Он нацелен на решение ряда вышерассмотренных проблем.

### 1.2.1. Взаимодействие преподавателя и студента

Взаимодействие преподавателя и студента включает в себя:

* выдачу заданий студентам;
* сбор выполненных заданий;
* обратную связь.

Примером программного обеспечения, позволяющего выдавать задания, получать решения на проверку и отмечать прогресс выполнения, является Moodle.

Moodle (модульная объектно-ориентированная динамическая учебная среда) — это свободная система управления обучением, ориентированная прежде всего на организацию взаимодействия между преподавателем и студентами.

Используя Moodle преподаватель, по результатам выполнения студентами заданий, может выставлять оценки и оставлять комментарии. Таким образом, Moodle является центром обеспечения взаимодействия между участниками учебного процесса.

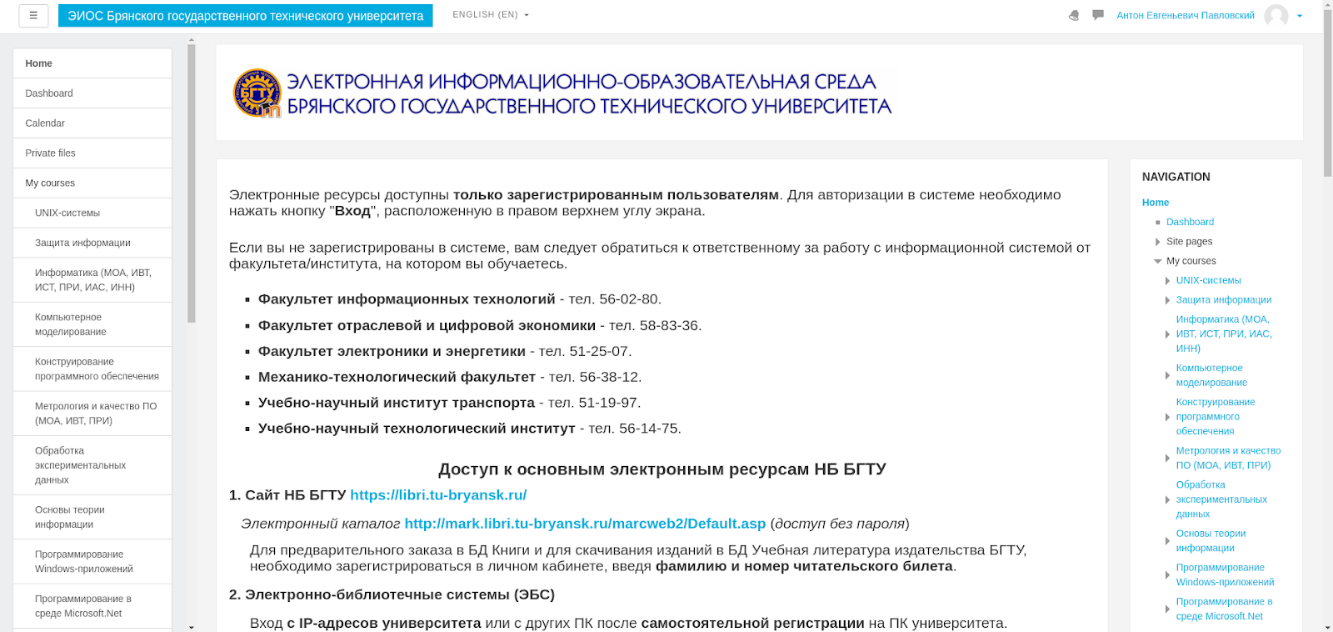


Рис. 2. Главная страница в системе Moodle БГТУ

Впрочем, можно выделить следующие недостатки:

* высокая сложность работы с системой;
* материалы передаются вручную, притом под каждого студента осуществляется ручная настройка заданий, нет никакой автоматизации;
* ограниченный размер принимаемых данных, ограниченные форматы;
* отсутствие автоматических оповещений.

Для упрощения обратной связи существуют электронные журналы. Вариантом такого журнала является АРМ БГТУ (Автоматизированное рабочее место преподавателя информатики Брянского государственного технического университета). Программный комплекс имеет возможность отображения выданных лабораторных работ по дисциплине, отметок об их выполнении, выставленных оценок и комментариев к работе. Также имеется возможность учета посещаемости и бонусных баллов.

На начальном экране веб-сервиса предлагается выбрать учебную группу, и изучаемую в ней дисциплину (рис. 3).

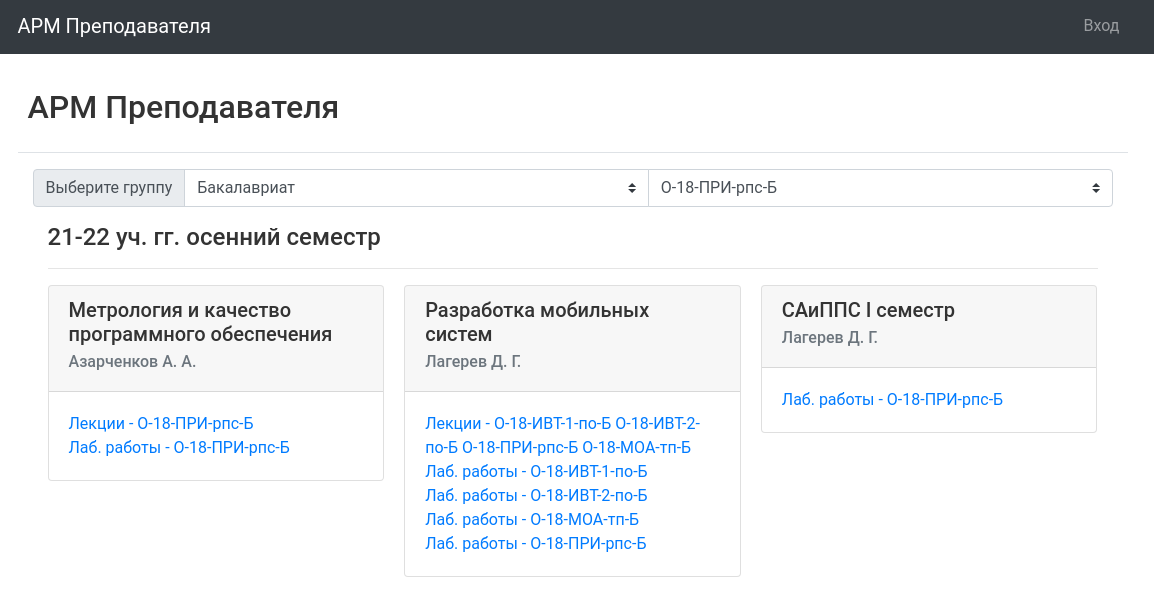
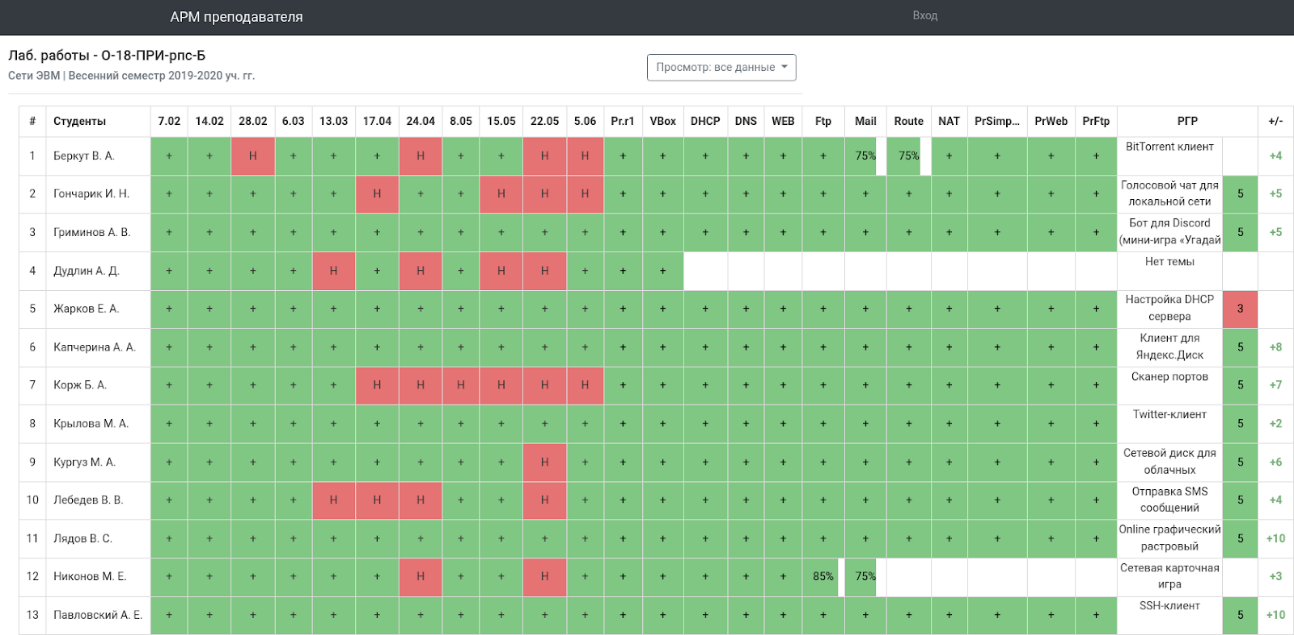


Рис. 3. Начальное окно АРМ БГТУ

Визуализация учета посещаемости и прогресса выполнения лабораторных работ представлена на рис. 4.

Главным достоинством системы ARM является быстрый доступ к статистике сдачи работ, оценкам, посещаемости. Недостаток же заключается в том, что выставление отметок приходится выполнять вручную, а данный процесс при большом количестве проверяемых работ может занять много времени, и большие объемы данных о выполнении работ вносить трудно.



### Рис. 4. Учёт посещаемости и выполнения лабораторных работ в АРМ БГТУ

Для передачи материалов нередко используется службы обмена файлами, именуемые файловыми хостингами.

Файловый хостинг – это служба, предоставляющая пользователю место под его файлы, с целью их публикации, хранения и скачивания. Практически все современные файловые хостинги в настоящее время эволюционировали в облачные хранилища данных или же сразу создавались в таком формате.

Одним из активно используемых облачных файловых хостингов с программой-клиентом является Dropbox. Dropbox позволяет пользователям создать специальную папку на своих компьютерах, которая синхронизируется таким образом, что она имеет одинаковое содержимое независимо от того, какое устройство используется для просмотра (рис. 5). Данная служба отлично подходит для выполнения студентами резервного копирования выполненных заданий и синхронизации собственных данных.

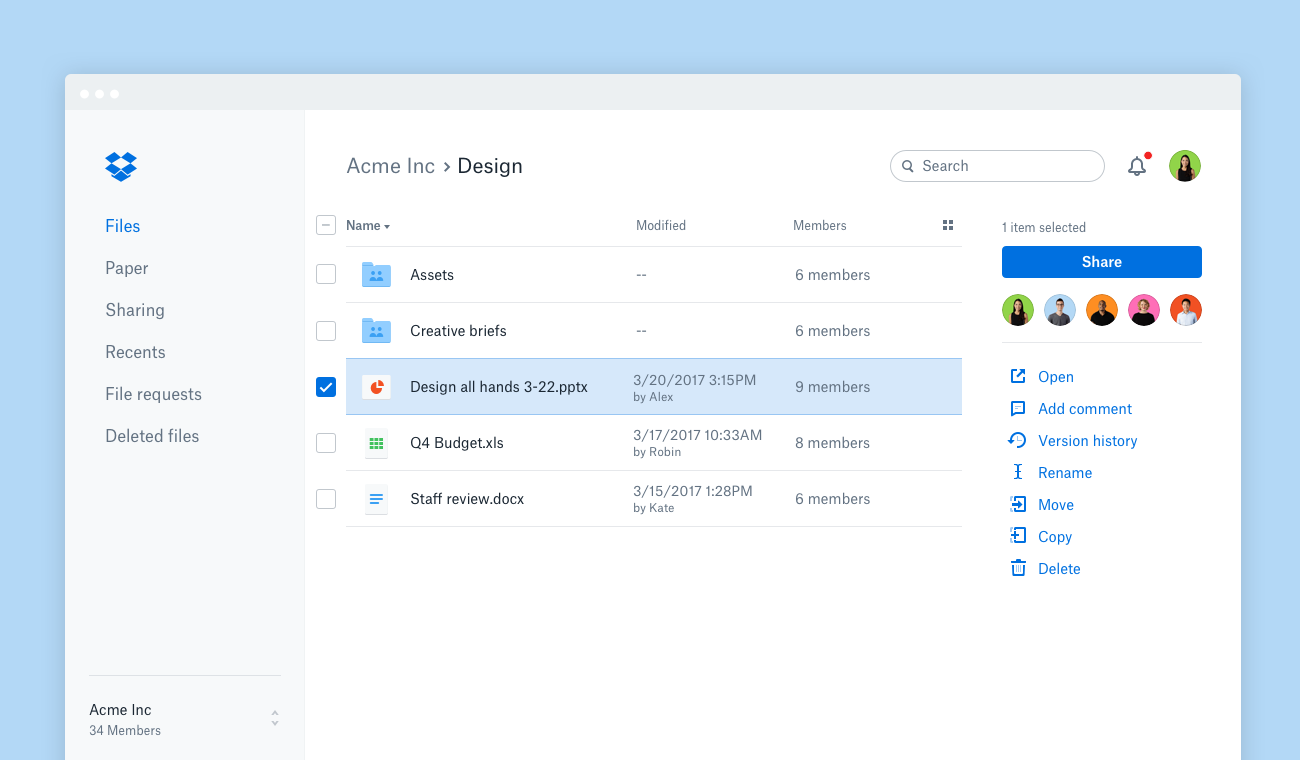


Рис. 5. Встроенная папка файлообменника Dropbox

### 1.2.2. Автоматическая проверка

Системы обнаружения ошибок в программном коде используются для контроля за качеством выполнения заданий студентами, позволяют обнаружить не только синтаксические, но и логические ошибки, а также ошибки, связанные с выделением памяти, контролем диапазонов и граничных значений и многие другие. Кроме того, данный вид программного обеспечения позволяет оценить качество программного кода в целом, его оформление и структуру. Примером такой системы является PVS-Studio. При всех её достоинствах, главным её недостатком является высокая стоимость.

Для проверки правильности решения может быть использована одна из систем автоматического тестирования. Преподаватель добавляет в подобную систему наборы тестовых данных, а студент потом может загрузить своё решение задачи и узнать результаты проверки. Timus — одна из таких систем.

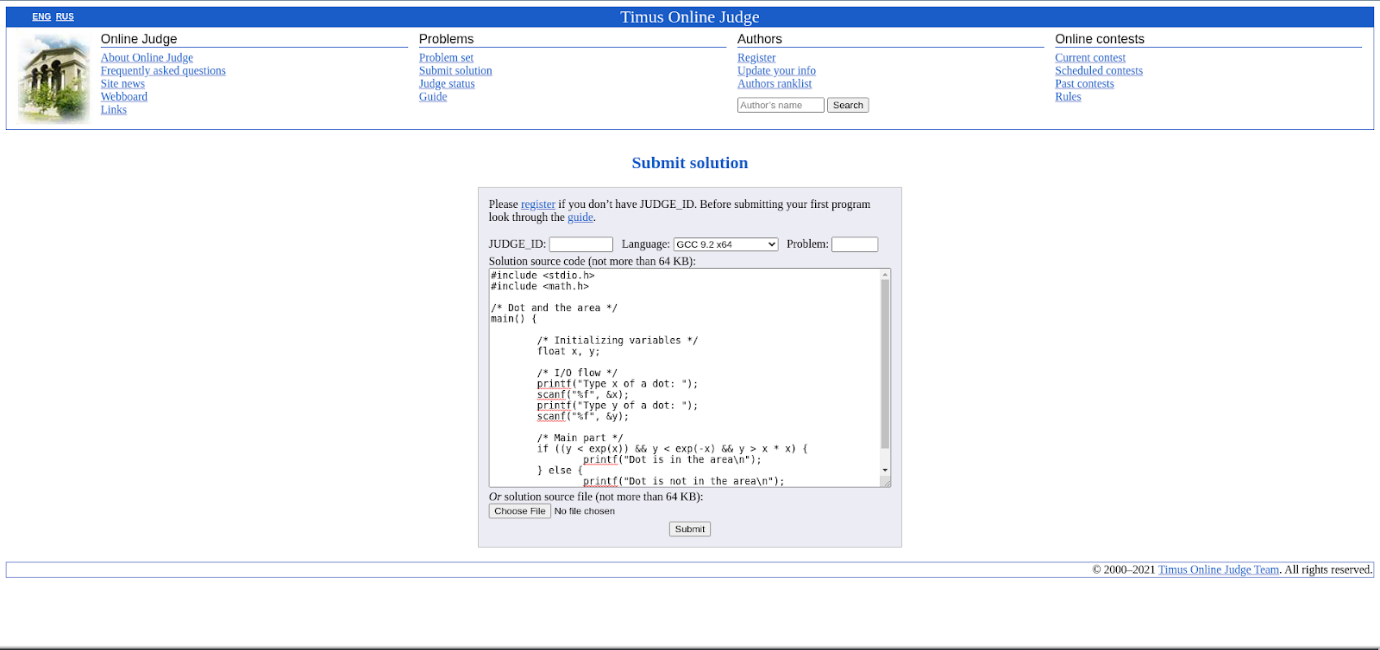


Рис 6. Страница загрузки решения в Timus

Подобные системы проверки заданий не могут исключить вмешательство преподавателя в учебный процесс, но позволяют сэкономить время на проверку решений и учесть такие параметры, как время выполнения и затраченный объем оперативной памяти.

Тем не менее, системы являются изолированными, то есть отсутствует интеграция с вышеупомянутыми системами управления обучением и обмена файлами.

### 1.2.3. Вывод

Рассмотренные программные комплексы не решают всех перечисленных проблем. Они лишь частично позволяют организовать и автоматизировать учебный процесс.

Из всего вышеперечисленного можно сделать вывод о необходимости разработки единого программного комплекса управления процессом образования студентов по дисциплинам, связанных с программированием.

## 1.3. Функциональная модель разрабатываемой системы

<Lab04>

# 2. ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Наименование программы: «Информационная система для организации учебной практики для студентов, обучающихся программированию».

## 1.1. Основание для разработки

Основанием для разработки программного комплекса управления процессом обучения студентов является задание на дипломную работу, выданное <TODO> на основании приказа по БГТУ № <TODO> от <TODO>.

## 1.2. Назначение и область применения

Программный комплекс предназначен для:

* анализа успешности выполнения практических заданий студентами при изучении дисциплин, связанных с программированием;
* интеграции средств выполнения лабораторных работ в единую систему с присутствием информационных связей между отдельными ее составляющими.

Данный программный комплекс применим для таких учебных курсов, как «Программирование» и «Объектно-ориентированное программирование».

## 1.3. Требование к программе

### 1.3.1. Требования к функциональным характеристикам

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

1. Осуществлять авторизацию пользователей при входе в систему.
2. При входе в веб-сервис под учетной записью, соответствующей роли **«Студент»**, пользователь должен иметь возможность:
   1. видеть список доступных курсов и иметь возможность просмотреть для конкретного курса прогресс, задания, дополнительные темы и дополнительные задания;
   2. видеть прогресс по темам, относящимся к курсу, значения индикаторов для тем, а также значение прогресса по курсу;
   3. видеть темы, к которым принадлежат задания, формулировку заданий, их статус и значения индикаторов при наличии;
   4. отмечать дополнительные задания как принятые на исполнение;
   5. отмечать дополнительные темы как принятые на исполнение.
3. При входе в веб-сервис под учетной записью, соответствующей роли **«Преподаватель»**, пользователь должен иметь возможность:
   1. видеть список доступных курсов;
   2. изменять структуру учебного курса (добавлять, изменять и удалять темы курса);
   3. вносить в систему наборы заданий для выполнения студентами при изучении курса, определять для них вычисляемые и задаваемые вручную индикаторы, определять обязательность задания;
   4. получать информацию об прогрессе студентов по курсу;
   5. участвовать в процессе оценивания работ студента (изменять значения автоматически рассчитанных индикаторов, задавать значения индикаторов, вносимые в систему вручную);
   6. устанавливать целевой коэффициент изученности темы для студента.
4. При входе в веб-сервис под учетной записью, соответствующей роли **«Администратор»**, пользователь должен иметь возможность:
   1. управлять списками студентов (добавлять и удалять студентов, переводить их в другую группу, изменять их личные данные);
   2. управлять списком преподавателей (добавлять и удалять преподавателей, изменять их личные данные);
   3. управлять учебными курсами (создавать, удалять, переименовывать, назначать группы студентов для изучения курса, назначать преподавателя, ответственного за курс);
   4. изменять пароли пользователей;
   5. добавлять собственные метрики.
5. База данных должна хранить сведения о пользователях, параметры курсов, задания и их параметры, параметры прохождения учебных курсов студентами, индикаторы выполнения. </FIXME>
6. Требования к модулю среды программирования:
   1. отображение принятых на выполнение заданий;
   2. возможность выбрать задание как исполняемое в данный момент;
   3. возможность отменить необязательное задание как не принятое на исполнение;
   4. контроль времени выполнения задания;
   5. контроль количества компиляций (успешных и завершенных с ошибкой);
   6. осуществление передачи кода на обработку на сервер.
7. Модуль изолированной среды выполнения должен осуществлять автоматическую проверку соответствия тестовым значениям.
8. Модуль выгрузки результатов в АРМ должен выполнять выгрузку результатов работы из системы в АРМ БГТУ (выданные задания, выполненные задания, выставленные оценки). </FIXME>

### 1.3.2. Требования к надёжности

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением Заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

* 1. организацией бесперебойного питания технических средств;
  2. использованием лицензионного программного обеспечения;

1. регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;
2. регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов.

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 30-ти минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Отказы программы вследствие некорректных действий пользователя при взаимодействии с программой недопустимы.

## 1.4. Условия эксплуатации

### 1.4.1. Климатические условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

### 1.4.2. Требования к квалификации и численности персонала

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 3 единиц – администратор, преподаватель, студент.

Администратор должен обладать навыками работы с ЭВМ на уровне уверенного пользователя.  В перечень задач, выполняемых администратором, входят:

* управление учебными курсами;
* управление списком преподавателей;
* управление списками студентов.

Преподаватель должен являться сотрудником учебного заведения, осуществляющим учебный процесс, и обладать навыками работы с ЭВМ на уровне уверенного пользователя. В перечень задач, выполняемых преподавателем, входят:

* формирование структуры учебного курса;
* формирование базы заданий для выполнения в рамках учебного курса;
* контроль за результатами работы студента на протяжении всего процесса обучения.

Студент должен обладать навыками работы с ЭВМ на уровне уверенного пользователя.

### 1.4.3. Требования к составу и параметрам технических средств

1. IBM PC-совместимый персональный компьютер для работы с системой должен соответствовать следующим параметрам.
   1. Процессор с тактовой частотой не ниже 1,8 ГГц. Рекомендуется использовать как минимум двухъядерный процессор.
   2. 2 ГБ ОЗУ; рекомендуется 4 ГБ ОЗУ (минимум 2,5 ГБ при выполнении на виртуальной машине).
   3. 50 ГБ свободного места на жестком диске.
   4. Видеоадаптер с минимальным разрешением 720p (1280 на 720 пикселей).
2. IBM PC-совместимый персональный компьютер, выполняющий роль сервера, должен соответствовать следующим параметрам.
   1. Процессор совместимый с Intel, имеющий частоту не ниже 2 ГГц.
   2. Оперативная память объемом не менее 4 ГБ.
   3. Свободное место на жёстком диске не менее 80 ГБ.
   4. Операционная система Microsoft Windows 8.1 или новее.
3. Для корректной работы требуется наличие интернет-соединения со скоростью не менее 5 Мбит/с.

### 1.4.4. Требования к информационной и программной совместимости

Программные средства, используемые на серверной стороне программного комплекса, должны быть представлены лицензионными версиями операционной системы «Microsoft Windows 8.1» или новее и системы управления базами данных «PostgreSQL 10» или новее.

Программные средства, используемые на клиентской стороне программного комплекса, должны быть представлены:

* + - * лицензионной версией операционной системы «Microsoft Windows 8.1» или новее;
      * интегрированной средой разработки Visual Studio 2017;
      * web-браузером Google Chrome версии 55 и выше или Mozilla Firefox версии 57 и выше;

Требования к защите информации и программ, и специальные требования не предъявляются.

## 1.5. Стадии и этапы разработки

### 1.5.1. Стадии разработки

Разработка должна быть проведена в три стадии:

* + - 1. разработка технического задания;
      2. рабочее проектирование;
      3. внедрение. </FIXME>

### 1.5.2. Этапы разработки

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

* + - 1. разработка программы;
      2. разработка программной документации;
      3. испытания программы.

На стадии внедрения должен быть выполнен этап подготовки и передачи программы. </FIXME>

### 1.5.3. Содержание работ по этапам

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

* + - 1. постановка задачи;
      2. определение и уточнение требований к техническим средствам;
      3. определение требований к программе;
      4. определение стадий, этапов и сроков разработки программы;
      5. согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию и отладке программы.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

* + - 1. разработка, согласование и утверждение методики испытаний;
      2. проведение испытаний;
      3. корректировка программы и программной документации по результатам испытаний.

Представленное техническое задание описывает не только функциональные требования к программному комплексу, но и требования к надежности и условия эксплуатации, а также требования к материально-аппаратному и программному обеспечению.