

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учебно-методическое объединение по образованию
в области информатики и радиоэлектроники

УТВЕРЖДАЮ

Первый заместитель Министра образования
Республики Беларусь

_____ И.А. Старовойтова

Регистрационный № ТД-_____ /тип.

ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Типовая учебная программа по учебной дисциплине
для специальностей:

1-40 05 01 Информационные системы и технологии (по направлениям);

1-39 03 02 Программируемые мобильные системы;

1-58 01 01 Инженерно-психологическое обеспечение информационных технологий

СОГЛАСОВАНО

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

_____ А.С. Турцевич

СОГЛАСОВАНО

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

_____ В.А. Богуш

СОГЛАСОВАНО

Начальник Главного управления
профессионального образования
Министерства образования
Республики Беларусь

_____ С.А. Касперович

СОГЛАСОВАНО

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

_____ И.В. Титович

Эксперт-нормоконтролер

Минск 2021

СОСТАВИТЕЛИ:

В.Н. Комличенко, доцент кафедры экономической информатики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат технических наук, доцент;

М.М. Меженная, доцент кафедры инженерной психологии и эргономики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат технических наук, доцент;

Д.А. Сторожев, старший преподаватель кафедры экономической информатики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», магистр экономических наук;

Т.В. Казак, заведующий кафедрой инженерной психологии и эргономики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», доктор психологических наук, профессор;

В.В. Хорошко, заведующий кафедрой проектирования информационно-компьютерных систем учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат технических наук, доцент.

РЕЦЕНЗЕНТЫ:

Кафедра дискретной математики и алгоритмики Белорусского государственного университета (протокол № 19 от 07.05.2021 г.);

В.А. Грушев, заместитель директора по производству Иностранного унитарного научно-производственного предприятия «САМСОЛЮШНС».

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ В КАЧЕСТВЕ ТИПОВОЙ:

Кафедрой экономической информатики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № 12 от 13.04.2021 г.);

Кафедрой инженерной психологии и эргономики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № 16 от 19.04.2021 г.);

Научно-методическим советом учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № 10 от 21.05.2021 г.);

Научно-методическим советом по прикладным информационным системам и технологиям Учебно-методического объединения по образованию в области информатики и радиоэлектроники (протокол № 5 от 05.05.2021 г.);

Научно-методическим советом по разработке программного обеспечения и информационно-коммуникационным технологиям Учебно-методического объединения по образованию в области информатики и радиоэлектроники (протокол № 6 от 18.05.2021 г.);

Ответственный за редакцию: С.С. Шишпаронок

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

ХАРАКТЕРИСТИКА УЧЕБНОЙ ДИСЦИПЛИНЫ

Типовая учебная программа по учебной дисциплине «Технологии разработки программного обеспечения» разработана для студентов учреждений высшего образования, обучающихся по специальностям 1-40 05 01 Информационные системы и технологии (по направлениям), 1-39 03 02 Программируемые мобильные системы, 1-58 01 01 Инженерно-психологическое обеспечение информационных технологий в соответствии с требованиями образовательных стандартов высшего образования первой ступени и типовых учебных планов вышеуказанных специальностей.

Учебная дисциплина «Технологии разработки программного обеспечения» является одной из дисциплин начального цикла подготовки студентов в области информационных технологий. Сущность учебной дисциплины составляют базовые принципы, методы и средства разработки программного обеспечения в части анализа и формализации требований, моделирования бизнес-процессов и алгоритмизации проектных решений, кодирования и отладки приложений.

Учебная дисциплина «Технологии разработки программного обеспечения» используется для освоения базового уровня моделирования, алгоритмизации и программирования решений профессиональных задач, способствует формированию интеллектуального и творческого потенциала личности будущего программиста.

Практическая деятельность инженера требует определенных знаний в области создания условий для обеспечения интеграции и сотрудничества ИТ-специалистов на всех этапах жизненного цикла разработки программного обеспечения, а также навыков применения социально-психологических методов управления, обладания позитивным профессиональным и личностным мышлением. Принципы и технологии создания дружественных пользовательских интерфейсов программного обеспечения, которые изучаются в ходе освоения студентами учебной дисциплины «Технологии разработки программного обеспечения», предполагают воспитание культуры и этики деловых отношений, развитие навыков разрешения конфликтных ситуаций, оптимизации морально-психологического климата в коллективе и поддержания партнерских взаимоотношений, направленных на творческое исполнение обязанностей.

ЦЕЛЬ, ЗАДАЧИ, РОЛЬ УЧЕБНОЙ ДИСЦИПЛИНЫ

Цель учебной дисциплины: формирование систематизированных знаний о жизненном цикле разработки программного обеспечения и технологиях, применяемых на различных его этапах, включая моделирование предметной области, формализацию требований, алгоритмизацию проектных решений, программную реализацию и отладку приложений.

Задачи учебной дисциплины:

приобретение знаний о цели и основных задачах в области разработки программного обеспечения;

приобретение знаний об основах моделей и методологий жизненного цикла разработки программного обеспечения;

приобретение знаний о парадигмах программирования;

овладение базовыми методами анализа предметной области и формализации требований к разработке программного обеспечения;

овладение базовыми методами моделирования и алгоритмизации для анализа и разработки проектных решений;

овладение базовыми методами написания качественного и эффективного кода;

изучение принципов юзабилити для создания дружественных пользовательских интерфейсов;

ознакомление со стандартами разработки программных средств и систем и областью программной инженерии.

Учебная дисциплина «Технологии разработки программного обеспечения» является основой для такой учебной дисциплины, как «Объектно-ориентированное проектирование и программирование», также она может изучаться параллельно или после изучения учебной дисциплины «Основы алгоритмизации и программирования».

ТРЕБОВАНИЯ К УРОВНЮ ОСВОЕНИЯ СОДЕРЖАНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

В результате изучения учебной дисциплины «Технологии разработки программного обеспечения» формируются следующие компетенции:

универсальные:

1) владеть основами исследовательской деятельности, осуществлять поиск, анализ и синтез информации;

2) обладать навыками саморазвития и совершенствования в профессиональной деятельности;

3) проявлять инициативу и адаптироваться к изменениям в профессиональной деятельности;

базовые профессиональные:

применять современные языковые и инструментальные методы и средства визуального моделирования процессов решения задач, представлять программную реализацию моделей в конструкциях изучаемого языка программирования.

В результате изучения учебной дисциплины обучающийся должен:

знать:

базовые понятия информационных технологий, основные и перспективные направления развития информационных систем и технологий;

определение, эволюционное развитие моделей жизненного цикла разра-

ботки программного обеспечения;

парадигмы программирования и существующие подходы к разработке программ;

методы, технологии и средства анализа и моделирования предметной области;

методы, технологии и средства анализа, моделирования и алгоритмизации проектных решений;

принципы, методы и средства структурного программирования;

принципы, методы и средства объектно-ориентированного программирования;

уметь:

выявлять и определять существенные элементы разработки;

выполнять анализ предметной области;

определять и формулировать требования к разработке программного обеспечения;

выполнять графическую интерпретацию проектных решений;

применять современные подходы к программированию и отладке приложений;

владеть:

современными технологиями проектирования и разработки программного обеспечения;

навыками в составе группы специалистов разрабатывать проектную документацию к программному обеспечению;

методами кодирования и отладки программного обеспечения для реализации проектных решений;

владеть современными средствами инфокоммуникаций.

Программа рассчитана на:

1) для специальностей 1-40 05 01 «Информационные системы и технологии (по направлениям)» и 1-58 01 01 «Инженерно-психологическое обеспечение информационных технологий» 110 учебных часов, из них – 66 аудиторных. Примерное распределение аудиторных часов по видам занятий: лекций – 34 часа, лабораторных занятий – 32 часа. Курсовая работа – 30 часов.

2) для специальности 1-39 03 02 «Программируемые мобильные системы» 144 учебных часа, из них – 70 аудиторных. Примерное распределение аудиторных часов по видам занятий: лекций – 34 часа, лабораторных занятий – 36 часов. Курсовая работа – 30 часов.

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

ПРИМЕРНЫЙ ТЕМАТИЧЕСКИЙ ПЛАН

Наименование раздела, темы	Всего аудиторных, часы	Лекции, часы	Лабораторные занятия, часы
Раздел 1. Введение в терминологию и методологию разработки программного обеспечения	12	12	-
Тема 1. Жизненный цикл разработки программного обеспечения: этапы, модели и методологии	4	4	-
Тема 2. Понятие программы. Парадигмы программирования	4	4	-
Тема 3. Развитие представлений о разработке программ: от спагетти-кода к методологии структурного программирования. Объектно-ориентированное программирование	4	4	-
Раздел 2. Технологии проектирования программного обеспечения	28	12	16
Тема 4. Анализ предметной области. Формализация и управление требованиями к программному обеспечению	8	4	4
Тема 5. Технологии проектирования проектных решений: общие принципы, методы, стандарты	8	4	4
Тема 6. Моделирование и алгоритмизация как средства проектирования программного обеспечения	12	4	8
Раздел 3. Технологии разработки и отладки программного обеспечения	26/30*	10	16/20*
Тема 7. Инструменты программирования и отладки: интегрированная среда разработки, система контроля версий, системы управления проектами	8/12*	4	4/8*
Тема 8. Принципы и технологии создания качественного кода	12	4	8
Тема 9. Принципы и технологии создания дружелюбных пользовательских интерфейсов программного обеспечения	6	2	4
Итого:	66/70*	34	32/36*

Примечание: * - количество часов для специальности 1-39 03 02 «Программируемые мобильные системы».

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. ВВЕДЕНИЕ В ТЕРМИНОЛОГИЮ И МЕТОДОЛОГИЮ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Тема 1. ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ: ЭТАПЫ, МОДЕЛИ И МЕТОДОЛОГИИ

Понятие жизненного цикла разработки программного обеспечения. Этапы жизненного цикла и их характеристика. Эволюция моделей жизненного цикла: каскадная, V-образная, итерационно-инкрементальная, спиральная. Гибкие методологии разработки: принципы и особенности на примерах Scrum, Kanban.

Тема 2. ПОНЯТИЕ ПРОГРАММЫ. ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ

Определение и характеристики программ. Стили программирования: императивный и декларативный. Парадигмы программирования: структурная и объектно-ориентированная, функциональная и логическая.

Тема 3. РАЗВИТИЕ ПРЕДСТАВЛЕНИЙ О РАЗРАБОТКЕ ПРОГРАММ: ОТ СПАГЕТТИ-КОДА К МЕТОДОЛОГИИ СТРУКТУРНОГО ПРОГРАММИРОВАНИЯ. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Характеристика и проблемы неструктурного программирования. Принципы структурного программирования и преимущества модульных программ. Базовые программные конструкции структурного программирования: линейная, разветвляющаяся, циклическая. Объектно-ориентированное программирование как ответ на возрастающую сложность программного обеспечения: базовые понятия, общая концепция, сложности подхода.

Раздел 2. ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Тема 4. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ. ФОРМАЛИЗАЦИЯ И УПРАВЛЕНИЕ ТРЕБОВАНИЯМИ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Предпроектные исследования и анализ предметной и проблемной областей.

Разработка требований к программному обеспечению: подходы и инструменты. Понятие спецификации и ее наполнение.

Тема 5. ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ ПРОЕКТНЫХ РЕШЕНИЙ: ОБЩИЕ ПРИНЦИПЫ, МЕТОДЫ, СТАНДАРТЫ

Общие принципы проектирования программного обеспечения как детерминированного, эвристического, итерационного процесса. Модульное проектирование. Нисходящий и восходящий подходы к проектированию.

Тема 6. МОДЕЛИРОВАНИЕ И АЛГОРИТМИЗАЦИЯ КАК СРЕДСТВА ПРОЕКТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Структурное моделирование и проектирование программных средств.

Функциональное моделирование: методы, технологии, стандарты и средства.

Информационное моделирование: методологии, стандартные методы и средства разработки информационных моделей.

Понятие, значение и правила оформления алгоритма. Линейные, разветвляющиеся и циклические конструкции в алгоритмах.

Раздел 3. ТЕХНОЛОГИИ РАЗРАБОТКИ И ОТЛАДКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Тема 7. ИНСТРУМЕНТЫ ПРОГРАММИРОВАНИЯ И ОТЛАДКИ: ИНТЕГРИРОВАННАЯ СРЕДА РАЗРАБОТКИ, СИСТЕМА КОНТРОЛЯ ВЕРСИЙ, СИСТЕМЫ УПРАВЛЕНИЯ ПРОЕКТАМИ

Особенности работы в интегрированной среде разработки: структура проекта, запуск и отладка. Значение и описание работы с системой контроля версий. Особенности командной разработки с использованием систем управления проектами. Примеры инструментов и их сравнительный анализ.

Тема 8. ПРИНЦИПЫ И ТЕХНОЛОГИИ СОЗДАНИЯ КАЧЕСТВЕННОГО КОДА

Соглашение о коде. Рефакторинг как средство совершенствования кода. Обработка исключительных ситуаций. Валидация вводимых данных. Принципы информационной безопасности.

Тема 9. ПРИНЦИПЫ И ТЕХНОЛОГИИ СОЗДАНИЯ ДРУЖЕСТВЕННЫХ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Понятие, значимость и принципы юзабилити в создании эргономичных пользовательских интерфейсов. Существующие подходы к реализации принципов юзабилити и практические примеры дружественных интерфейсов.

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ**ЛИТЕРАТУРА****ОСНОВНАЯ**

1. Гниденко, И. Г. Технология разработки программного обеспечения / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. – Москва: Юрайт, 2019.
2. Мартин, Р. Чистая архитектура. Искусство разработки программного обеспечения / Р. Мартин. – Санкт-Петербург : Питер, 2019.
3. Гагарина, Л. Г. Технология разработки программного обеспечения / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Сидорова-Виснадул; под ред. Л. Г. Гагариной. – Москва: ИД «ФОРУМ»: ИНФРА-М, 2021.
4. Вигерс, К. Разработка требований к программному обеспечению. / К. Вигерс, Дж. Битти. / 3-е изд., доп. ; пер. с англ. – М. : Русская редакция; СПб. : БХВ-Петербург, 2019.
5. Мартин, Р. Чистый код : создание, анализ и рефакторинг / Р. Мартин. – Санкт-Петербург: Питер, 2021.
6. Макконнелл, С. Совершенный код. Мастер-класс / С. Макконнелл. – М.: Русская редакция, 2019.
7. Купер, А. Интерфейс. Основы проектирования взаимодействия. – / А. Купер, Р. Рейман ; пер. с англ. – СПб. : Питер, 2019.
8. Федоров, И. Г. Моделирование бизнес-процессов в нотации BPMN 2.0: научно-практическое издание / И. Г. Федоров. М-во образования и науки Российской Федерации, Московский гос. ун-т экономики, статистики и информатики (МЭСИ). – Москва : МЭСИ, 2013.
9. Репин, В. В. Процессный подход к управлению. Моделирование бизнес-процессов / В. В. Репин, В. Г. Елиферов. – М. : Стандарты и качество, 2004.
10. Ларман, К. Применение UML 2.0 и шаблонов проектирования. Введение в объектно-ориентированный анализ, проектирование и итеративную разработку / К. Ларман. – М. : Вильямс, 2013.
11. Буч, Г. Язык UML. Руководство пользователя / Г. Буч, Дж. Рамбо, А. Джекобсон. – М. : ДМК, 2015.
12. Буч, Г. Введение в UML от создателей языка / Г. Буч, Д. Рамбо, И. Якобсон. – М. : ДМК Пресс, 2015.
13. Гамма, Э. Приёмы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон [и др.]. – СПб. : Питер, 2015.
14. Чакон С. Git для профессионального программиста / С. Чакон, Б. Страуб – СПб. : Питер, 2016.
15. Маклаков, С. В. Создание информационных систем с AllFusion Modeling Suite / С. В. Маклаков. – М. : ДИАЛОГ – МИФИ, 2003.
16. ИСО/МЭК 12207 – 95 «Информационная технология. Процессы жизненного цикла программных средств» или ISO/IEC 12207(ISO – International Organization of Standardization)

ДОПОЛНИТЕЛЬНАЯ

17. Боггс, М. UML и Rational Rose / М. Боггс. – Москва : РГГУ, 2010.
18. Фаулер, М. UML. Основы. Краткое руководство по стандартному языку объектного моделирования / М. Фаулер. – Москва : Символ-Плюс, 2011.
19. Розенберг, Д. Применение объектного моделирования с использованием UML и анализ прецедентов. / Д. Розенберг, К. Скотт ; пер. с англ. – М. : ДМК Пресс, 2002.
20. ГОСТ 34.601–90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы, стадии создания.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

При изучении учебной дисциплины рекомендуется использовать следующие формы самостоятельной работы:

- написание рефератов и создание материалов презентаций;
- составление тестов и эталонов ответов к ним;
- подготовка к контрольной работе или коллоквиуму;
- участие студентов в научно-исследовательской и методической работе, проводимой на кафедре;
- участие в конкурсах студенческих работ и студенческих конференциях.

ПЕРЕЧЕНЬ РЕКОМЕНДУЕМЫХ СРЕДСТВ ДИАГНОСТИКИ КОМПЕТЕНЦИЙ СТУДЕНТА

Типовыми учебными планами по специальностям 1-40 05 01 «Информационные системы и технологии (по направлениям)», 1-39 03 02 «Программируемые мобильные системы», 1-58 01 01 «Инженерно-психологическое обеспечение информационных технологий» в качестве формы текущей аттестации по учебной дисциплине «Технологии разработки программного обеспечения» рекомендуется экзамен и курсовая работа.

Оценка учебных достижений студента производится по десятибалльной шкале.

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

- устный опрос;
- защита лабораторной работы;
- собеседование;
- защита индивидуальной практической работы;
- коллоквиум.

РЕКОМЕНДУЕМЫЕ МЕТОДЫ (ТЕХНОЛОГИИ) ОБУЧЕНИЯ

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

устное изложение учебного материала на лекциях, сопровождаемое показом и демонстрацией слайдов, видеоматериалов;

компьютерное обучение;

элементы проблемного обучения (проблемное изложение, вариативное изложение, частично-поисковый метод), реализуемые на лекционных занятиях;

элементы учебно-исследовательской деятельности, реализуемые на лабораторных занятиях и в процессе выполнения курсовой работы.

КУРСОВОЕ ПРОЕКТИРОВАНИЕ

Целью курсовой работы является получение студентами теоретических знаний и практических навыков по конструированию и реализации программ на языке высокого уровня в рамках процедурной парадигмы. На уровне конструирования задачей курсовой работы является освоение технологии нисходящего проектирования и принципов методологии структурного программирования. Результат этапа конструирования представляется в форме графических схем алгоритмов. На уровне программирования задачей курсовой работы является реализация графических схем алгоритмов на языке высокого уровня с соблюдением правил соглашения о коде (Code Convention). Результат этапа программирования представляется в форме проекта в среде разработки приложений на языке высокого уровня (например, Microsoft Visual Studio при использовании языка программирования C/C++/C#).

Исходные данные.

- 1) Язык программирования высокого уровня.
- 2) Среда разработки (IDE) для соответствующего языка программирования.
- 3) Вид приложения – консольное.
- 4) Парадигма программирования – процедурная (по согласованию с преподавателем допускается реализация программы в рамках объектно-ориентированной парадигмы программирования).
- 5) Способ организации данных – структуры (struct) (либо поля классов в случае объектно-ориентированного программирования).
- 6) Способ хранения данных – файлы (по согласованию с преподавателем допускается подключение баз данных).
- 7) Каждая логически завершенная подзадача программы должна быть реализована в виде отдельной функции (метода в случае объектно-ориентированного программирования).
- 8) Построение программного кода должно соответствовать соглашению о коде «Code Convention».
- 9) К защите курсовой работы представляются: консольное приложение, пояснительная записка, графический материал (плакаты и чертежи).

10) Текст пояснительной записки оформляется в соответствии со стандартом предприятия.

Функциональные требования.

Первым этапом работы программы является авторизация – предоставление прав доступа.

В рамках данного этапа необходимо считать данные из файла с учетными записями пользователей следующего вида:

login;

password;

role (данное поле служит для разделения в правах администраторов и пользователей).

После ввода пользователем своих персональных данных (логина и пароля) и сверки со считанной из файла информацией необходимо предусмотреть возможность входа:

в качестве администратора;

в качестве пользователя.

Регистрация новых пользователей осуществляется администратором в режиме работы с учетными записями пользователей (т. е. администратор сам создает для пользователей аккаунты).

Вторым этапом работы программы является собственно работа с данными, которая становится доступной только после прохождения авторизации. Данные хранятся в отдельном файле и имеют вид, описанный подробно в каждом варианте к курсовой работе. Для работы с данными должны быть предусмотрены два функциональных модуля: модуль администратора и модуль пользователя.

Модуль администратора включает следующие подмодули (с указанием функциональных возможностей):

1. Управление учетными записями пользователей:

- просмотр всех учетных записей;
- добавление новой учетной записи;
- редактирование учетной записи;
- удаление учетной записи.

2. Работа с данными:

а) режим редактирования:

- просмотр всех данных;
- добавление новой записи;
- удаление записи;
- редактирование записи;

б) режим обработки данных:

- выполнение индивидуального задания;
- поиск данных;
- сортировка.

Модуль пользователя включает подмодуль работы с данными со следующими функциональными возможностями:

просмотр всех данных;
 выполнение индивидуального задания;
 поиск данных;
 сортировка.

Для реализации перечисленных модулей/подмодулей необходимо создавать меню с соответствующими пунктами.

В курсовой работе необходимо предусмотреть:

1. Обработку исключительных ситуаций:
 - введенные пользователем данные не соответствуют формату поля (например, символы в числовом поле);
 - введенные пользователем данные нелогичны (например, отрицательная цена товара);
 - файл с данными для чтения не существует;
 - ничего не найдено по результатам поиска;
 - номер удаляемой записи выходит за пределы массива/вектора.
2. Возможность возврата назад (навигация).
3. Запрос на выполнение необратимых действий, а именно подтверждение удаления вида «Вы действительно хотите удалить файл (запись)?».
4. Обратную связь с пользователем, например, вывод сообщения об успешности удаления/редактирования записи и т. д.

Требования к программной реализации.

1. Все переменные и константы должны иметь осмысленные имена в рамках тематики варианта курсовой работы. Переменным рекомендуется присваивать имена, состоящие из букв нижнего регистра; для формирования составного имени используется нижнее подчеркивание (например, `number_of_students`) или «верблюжья нотация» (например, `flagExit`).
2. Имена функций должны быть осмысленными, начинаться с буквы нижнего регистра, строиться по принципу глагол + существительное (например, `addAccount`, `findStudentBySurname`). Если функция выполняет проверку и возвращает результат типа `bool`, то ее название должно начинаться с глагола `is` (например, `isNumberNumeric`, `isLoginUnique`).
3. Не допускается использование оператора прерывания `goto`.
4. Код не должен содержать неименованных числовых констант («магических» чисел), неименованных строковых констант (например, имен файлов и др.). Подобного рода информацию следует представлять, как глобальные константы. По правилам качественного стиля программирования тексты всех информационных сообщений, выводимых пользователю в ответ на его действия, также оформляются как константы.
5. Код необходимо комментировать (как минимум в части объявления структур, массивов/векторов, прототипов функций, нетривиальной логики).
6. Код не должен дублироваться – для этого существуют функции.
7. Одна функция решает только одну задачу (например, не допускается в одной функции считывать данные из файла и выводить их на консоль – это две разные функции). При этом внутри функции возможен вызов других функций.

8. Выполнение операций чтения/записи в файл должно быть сведено к минимуму (т. е. после однократной выгрузки данных из файла в массив/вектор дальнейшая работа ведется с этим массивом/вектором, а не происходит многократное считывание данных из файла в каждой функции).

9. Следует избегать глубокой вложенности условных и циклических конструкций: вложенность блоков должна быть не более трех.

10. Следует избегать длинных функций: текст функции должен уместиться на один экран (размер текста не должен превышать 25–50 строк).

Выбор способов организации данных.

1. В качестве выбора способа описания входных данных приводится описание следующих типов `struct` (с указанием конкретных полей):

для учетных записей пользователей;

для данных.

В случае объектно-ориентированного программирования приводятся названия предполагаемых классов и содержащихся в них полей. При работе с базой данных дополнительно приводится структура таблиц.

2. В качестве способа объединения входных данных указывается:

использование массивов (статически или динамически создаваемых) или векторов;

их выбранная область видимости (локальные или глобальные).

Разработка перечня пользовательских функций программы подразумевает перечисление прототипов функций, необходимых для реализации программы, и краткие комментарии к ним.

В случае использования объектно-ориентированного программирования приводятся методы для классов.

Разработка алгоритмов работы программы включает схемы алгоритмов (с кратким словесным описанием их работы в тексте пояснительной записки) для главной функции и двух любых других пользовательских функций. В случае использования объектно-ориентированного программирования разрабатываются алгоритмы для двух любых методов классов, а также алгоритм главной функции.

Алгоритм должен быть оформлен согласно ГОСТ 19.701-90 Схемы алгоритмов, программ, данных и систем.

При разработке алгоритмов рекомендуется использовать профессиональные графические редакторы диаграмм и блок-схем, например, Microsoft Visio.

Описание работы программы подразумевает краткое словесное описание работы программы со скриншотами консоли.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ КУРСОВЫХ РАБОТ

Общее для всех вариантов задание: реализовать авторизацию для входа в систему, функционал администратора и функционал пользователя.

1. Разработка программы учета сведений о музыкальном конкурсе. Информация о конкурсе включает: ФИО участника, год рождения, название страны, наименование музыкального инструмента (гитара, фортепиано, скрипка, виолончель и др.), занятое место по результатам конкурса. Индивидуальное задание: по каждому классу музыкальных инструментов вывести список первых трех мест с указанием возраста победителей. Вывести список самых молодых (до 12 лет) победителей конкурса в порядке увеличения возраста.

2. Разработка программы учета сведений о пациентах медицинского центра. Сведения о пациентах медицинского центра содержат: ФИО пациента, пол, дату рождения, место проживания (город), контактный телефон, диагноз. Индивидуальное задание: вывести иногородних пациентов. Вывести список пациентов старше X лет, у которых диагноз Y (X, y вводятся с клавиатуры).

3. Разработка программы продажи железнодорожных билетов. Железнодорожный вокзал осуществляет продажу билетов на поезда дальнего следования. Для каждого поезда указаны: номер поезда, пункт назначения, дату отправления, время отправления, время прибытия на конечный пункт, стоимость одного билета, количество оставшихся для продажи билетов, количество проданных билетов. Индивидуальное задание: обеспечить функциональную возможность покупки билетов на конкретный рейс в требуемом количестве (при этом количество оставшихся для продажи билетов и количество проданных билетов обновляется). Вывести номер, время отправления и наличие билетов для поездов, прибывающих в город X в интервале от A до B (X, A, B вводятся с клавиатуры).

4. Разработка программы планирования факультативных учебных дисциплин для студентов. Для формирования факультативных занятий необходимо обработать информацию следующего вида: ФИО студента, номер группы, средний балл успеваемости, пять возможных для факультативного посещения дисциплин. Выбираемая дисциплина помечается 1, невыбираемая – 0. Перечень предлагаемых дисциплин: «Математика», «Физика», «Программирование», «Английский язык», «Базы данных». Индивидуальное задание: вывести список и общее количество студентов, желающих прослушать дисциплину X. Если число желающих больше 15, то отобрать 15 студентов с более высоким баллом успеваемости. Вывести предлагаемые дисциплины в порядке убывания популярности с указанием общего числа записавшихся на каждую из них.

5. Разработка программы учета сведений об игроках хоккейной команды. Сведения об игроках хоккейной команды включают: ФИО игрока, дату рождения, количество сыгранных матчей, число заброшенных шайб, количество голевых передач, количество штрафных минут. Индивидуальное задание: вывести шесть лучших игроков (голы + передачи) с указанием их результативности.

6. Разработка программы продажи авиабилетов. Авиакомпания осуществляет продажу билетов на самолеты с указанием: номера рейса, типа самолета, пункта назначения, даты вылета, времени вылета, времени прилета, вместимости самолета, количества оставшихся билетов бизнес-класса, стоимости билета бизнес-класса, количества оставшихся билетов эконом-класса, стоимости билета эконом-класса. Индивидуальное задание: обеспечить функциональную воз-

возможность покупки билетов на конкретный рейс в требуемом количестве (при этом количество оставшихся для продажи билетов обновляется). Если на интересующий рейс нет билетов требуемого класса, то при наличии билетов другого класса на этот рейс – вывести соответствующее информационное сообщение (например, «Билетов эконом-класса на данный рейс нет в наличии, но имеются билеты бизнес-класса в количестве 10»).

7. Разработка программы учета автомобилей таксопарка. Таксопарк содержит информацию об имеющихся автомобилях: вид автомобиля (такси, микроавтобус, лимузин), вместимость, расход топлива, стоимость автомобиля, количество автомобилей данного вида. Индивидуальное задание: подсчитать общую стоимость таксопарка. Подсчитать общую стоимость автомобилей каждого вида. Подобрать автомобили, по вместимости, соответствующие заданному с клавиатуры диапазону.

8. Разработка программы расчета стипендии. Для расчета стипендии необходимо обработать информацию следующего вида: номер группы, ФИО студента, форма обучения (платная/бюджетная), зачеты по пяти предметам (зачет/незачет), отметки по четырем предметам, признак участия в общественной работе: 1 – активное участие, 0 – неучастие. Индивидуальное задание: рассчитать стипендию для студентов-бюджетников. При этом отличники (отметки 9, 10) и общественники получают 50 % надбавку, а просто отличники – 25 %. Студенты со средним баллом, равным или ниже 5, стипендию не получают. Базовый размер стипендии вводится с клавиатуры.

9. Разработка программы начисления пособий по уходу за ребенком. Сведения о детях сотрудниц компании содержат следующую информацию: ФИО сотрудницы, дата рождения ребенка, ФИО ребенка. Для каждого ребенка сотрудницы создается отдельная запись указанного вида. Индивидуальное задание: для каждой сотрудницы вывести количество детей, а также рассчитать общую сумму полагающихся пособий. Пособие начисляется для детей в возрасте до трех лет. Для первого ребенка в семье базовое пособие составляет X (вводится с клавиатуры), для второго – $1,25X$, для третьего и последующих – $1,5X$.

10. Разработка программы планирования бюджета проекта. В компании по разработке программного обеспечения для планирования работ по проекту необходимо обработать информацию следующего вида: наименование проекта, вид работ (работа над требованиями, разработка архитектуры, реализация, тестирование), ФИО сотрудника, предполагаемое количество часов, стоимость одного часа. Индивидуальное задание: для каждого проекта вывести его итоговую стоимость, перечень видов работ с указанием общего количества задействованных специалистов и стоимости данного этапа.

11. Разработка программы расчета выплат по больничным листам. Сведения о больничных сотрудников компании имеют следующий вид: ФИО сотрудника, год, месяц, количество дней, пропущенных по болезни, оплата за один день. Индивидуальное задание: для месяца X года Y вывести список сотрудников с указанием выплат по больничным листам для каждого из них. Вывести

общую сумму выплат по больничным листам за интересующий месяц. X, Y вводятся с клавиатуры.

12. Разработка программы подбора туристической путевки. Туристическая компания содержит сведения о предлагаемых путевках: тип путевки (отдых, экскурсионный тур, лечение, шопинг, круиз), страна пребывания, вид транспорта, количество дней, питание (завтраки/все включено), стоимость. Индивидуальное задание: вывести все путевки требуемого типа (вводится с клавиатуры) стоимостью менее X в порядке убывания стоимости.

13. Разработка программы учета продаж театральных билетов. Сведения о продажах билетной кассы содержат следующую информацию: дата, наименование театра, наименование спектакля, количество проданных билетов. Индивидуальное задание: вывести список самых популярных театров в порядке убывания в указанный месяц. Вывести список самых популярных спектаклей в порядке убывания в указанный месяц. Месяц вводится с клавиатуры.

14. Разработка программы учета продаж проездных билетов. Сведения о продажах проездных билетов содержат следующую информацию: год, месяц, наименование транспорта, количество проданных проездных билетов, стоимость одного проездного билета на данный вид транспорта. Индивидуальное задание: определить общий доход от продажи проездных билетов за определенный месяц. Вывести список самых востребованных видов транспорта в порядке убывания в указанный месяц. Месяц вводится с клавиатуры.

15. Разработка программы учета командировок сотрудников предприятия. Сведения о командировках сотрудников компании содержат следующую информацию: ФИО сотрудника, год, месяц, длительность командировки в днях, город, в который осуществлялся выезд, сумма командировочных расходов на один день. Индивидуальное задание: определить общие выплаты командировочных за указанный месяц. Вывести список наиболее часто посещаемых городов в порядке убывания за указанный период (с месяца X по месяц Y).

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

1. Разработка схем алгоритмов программ.
2. Разработка программного кода и отладка схемы проекта программы.
3. Требования, виды требований, разработка прототипа технического задания в соответствии со стандартами.
4. Разработка процессов или бизнес-процессов «as is – как есть» и «to be – как должно быть».
5. Описание бизнес-процессов на языке BPMN.
6. Применение объектно-ориентированного подхода в анализе и проектировании программного обеспечения (UML).
7. Модели и генерирование кода на основе UML – диаграмм.
8. Разработка проекта решения задачи на основе «сквозного» моделирования и оценка затрат ресурсов.
9. Моделирование проектных схем на основе отечественных стандартов разработки программных приложений.

10. Отладка и рефакторинг программного кода на основе выполненных ранее архитектурных проектных решений.
11. Оптимизация программного кода.
12. Формирование объектно-ориентированных моделей и разработка программных решений задач.
13. Работа с системой контроля версий.
14. Создание и конфигурация автоматической сборки.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ КОМПЬЮТЕРНЫХ ПРОГРАММ

1. Операционная система (например, Windows 7 и выше).
2. Среда программирования (например, Microsoft Visual Studio и т.п.).
3. MS Office.
4. AllFusion Process Modeler 7.
5. AllFusion Data Modeler 7.
6. Sparx Enterprise Architect.
7. StarUML.
8. Rational Rose.
9. Microsoft Visio 2016 (и выше).
10. MAVEN.
11. Bizagi Process Modeler.
12. ARIS Express.
13. IBM Rational RequisitePro.
14. Redmine.
15. Trello.
16. GitHub.