


Министерство образования Республики Беларусь  
Учебно-методическое объединение высших учебных заведений  
Республики Беларусь по естественнонаучному образованию

УТВЕРЖДАЮ

Первый заместитель Министра образования  
Республики Беларусь

 А.И. Жук

16.06.2010

(дата утверждения)

Регистрационный № ТД- Б. 288 /тип.

ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

Типовая учебная программа  
для высших учебных заведений по специальности

1-31 03 04 Информатика

СОГЛАСОВАНО

Председатель Учебно-методического  
объединения вузов Республики Беларусь  
по естественнонаучному образованию  
 В.В. Самохвал



СОГЛАСОВАНО

Начальник Управления высшего и  
среднего специального образования

 Ю.И. Миксюк

16.06.2010

(дата)

Проректор по учебной и  
воспитательной работе  
Государственного учреждения  
образования «Республиканский  
институт высшей школы»

 В.И. Шупляк

18.05.2010

(дата)

Эксперт-нормоконтролер



18.05.2010

(дата)

Минск 2009

**СОСТАВИТЕЛИ:**

Л.Ф. Дробушевич, доцент кафедры математического обеспечения автоматизированных систем управления Белорусского государственного университета, кандидат физико-математических наук, доцент;

**РЕЦЕНЗЕНТЫ:**

Кафедра электронных вычислительных машин Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»;

Б. А. Железко, заведующий кафедрой «Экономическая информатика» Учреждения образования «Белорусский государственный экономический университет», кандидат технических наук, доцент

**РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ В КАЧЕСТВЕ ТИПОВОЙ:**

Кафедрой математического обеспечения автоматизированных систем управления Белорусского государственного университета (протокол № 3 от «22» сентября 2008 г.);

Научно-методическим советом Белорусского государственного университета (протокол № 1 от «1» декабря 2008 г.);

Научно-методическим советом по прикладной математике и информатике Учебно-методического объединения вузов Республики Беларусь по естественнонаучному образованию (протокол № 3 от «10» марта 2009 г.);

**Ответственный за выпуск:** Л.Ф. Дробушевич

### Пояснительная записка

Дисциплина «Технология программирования» знакомит студентов с методами разработки программных продуктов с использованием различных инструментальных средств, включая интеграцию с CASE-системами. Особое внимание уделяется методам анализа и проектирования программных систем, проектированию интерфейса пользователя, а также вопросам оценки качества программного обеспечения.

Основными целями изучения дисциплины являются: формирование у студентов знаний основ методологии в области разработки программного обеспечения; систематизация и углубление знаний и умений в области информационных технологий; изучение основных приемов и методов разработки программных продуктов; освоение оценки качества программного обеспечения.

Основными задачами изучения дисциплины являются: формирование знаний и умений в области моделирования программных систем; развитие интересов к современным моделям оценки качества процесса разработки программного обеспечения.

В процессе изучения дисциплины рассматривается содержание основных этапов процесса создания программного продукта, в том числе анализ предъявляемых к нему требований, составление технического задания, проектирование, кодирование, включая интеграцию и тестирование программы в целом, а также её сопровождение после установки у заказчика.

Отдельные темы посвящены определению модели процесса разработки и управлению проектом, что особенно важно при работе в команде.

Основой для изучения технологии программирования является курс «Программирование». Методы, излагаемые в курсе, используются при изучении дисциплины «Модели данных и СУБД», а также при изучении ряда дисциплин специализации. Изучение технологии программирования позволяет студентам получить знания, необходимые им в дальнейшем для успешной работы по специальности.

В результате изучения дисциплины студент должен:

знать:

- методы проектирования программных систем;
- методы проектирования интерфейса пользователя;
- методы разработки программных продуктов с использованием различных инструментальных средств, включая интеграцию с CASE-системами;
- основные понятия и методы тестирования программного обеспечения;

уметь:

- анализировать требования, предъявляемые к программным системам, составлять техническое задание, моделировать программное обеспечение, а также проводить его тестирование;
- проводить оценку качества программного обеспечения.

В соответствии с типовым учебным планом специальности 1-31 03 04 «Информатика» учебная программа предусматривает для изучения дисциплины всего 145 учебных часов, в том числе 68 аудиторных часов, из них лекции – 34 часа, лабораторные занятия – 34 часа.

## Примерный тематический план

№	Название раздела, темы	Количество аудиторных часов		
		Всего	В том числе	
			Лекции	Лабораторные занятия
	<b>Раздел I. Организация процесса разработки программного обеспечения</b>	4	4	-
1.	Основные понятия и определения	1	1	-
2.	Модели процесса разработки программного обеспечения	2	2	-
3.	Управление проектом	1	1	-
	<b>Раздел II. Классические методы анализа и проектирования программных систем</b>	8	4	4
4.	Методы структурного анализа	4	2	2
5.	Методы структурного проектирования	4	2	2
	<b>Раздел III. Объектно-ориентированные методы разработки программных систем</b>	38	14	24
6.	Основные принципы представления программных систем	4	2	2
7.	Унифицированный язык моделирования UML	6	2	4
8.	Модели требований	6	2	4
9.	Статические модели	6	2	4
10.	Динамические модели	6	2	4
11.	Модели реализации	6	2	4
12.	Основы компонентной объектной модели	4	2	2
	<b>Раздел IV. Методы тестирования программного обеспечения</b>	14	8	6
13.	Основные методы тестирования	6	4	2
14.	Методика тестирования	4	2	2
15.	Особенности тестирования объектно-ориентированных программ	4	2	2
	<b>Раздел V. Вопросы оценки качества программного обеспечения</b>	4	4	-
16.	Методы оценки сложности программного обеспечения	2	2	-
17.	Вопросы оценки сцепления и связности	2	2	-
	<b>Всего</b>	<b>68</b>	<b>34</b>	<b>34</b>

## Содержание

### *Раздел I. Организация процесса разработки программного обеспечения*

#### *1. Основные понятия и определения*

Метод, методология, программный продукт, программное изделие, модуль и др. Жизненный цикл программного обеспечения. Фазы жизненного цикла. Организация процесса разработки программных систем (методы, средства, процедуры).

#### *2. Модели процесса разработки программного обеспечения*

Классические модели процесса разработки программного обеспечения (каскадная, спиральная, компонентно-ориентированная, инкрементная, RAD-модель). Тяжёловесные и облегчённые процессы разработки программных систем (унифицированный процесс, XP-процесс). Содержание этапов разработки. Особенности жизненного цикла объектно-ориентированных программных систем.

#### *3. Управление проектом*

Составляющие управления проектом. Управление и планирование. Масштаб проекта и риск. Управление персоналом проекта. Варианты организации персонала. Управление релизами. Инструментальные средства поддержки.

Характеристика современных CASE-средств. Классификация, перспективы развития, опыт использования. Роль CASE-инструментов в проектировании объектно-ориентированных систем. Связь CASE-технологий с методами быстрой разработки приложений (RAD).

### *Раздел II. Классические методы анализа и проектирования программных систем*

#### *4. Методы структурного анализа*

Диаграммы потоков данных – DFD, описание потоков данных и процессов, расширения для систем реального времени. Расширение возможностей управления. Методы анализа, ориентированные на структуры данных. Метод анализа и проектирования Джексона.

#### *5. Методы структурного проектирования*

Типы информационных потоков. Проектирование для потоков данных типа "преобразование" и "запрос". Структурные карты Константайна. Методология SADT (диаграммы Росса). Доопределение функций. Учёт системного времени.

### *Раздел III. Объектно-ориентированные методы разработки программных систем*

#### *6. Основные принципы представления программных систем*

Абстрагирование, инкапсуляция, модульность, иерархия. Общая характеристика объектов. Виды отношений между объектами. Связи. Агрегация. Общая характеристика классов. Виды отношений между классами. Роль декомпозиции в объектном анализе и проектировании. Роль декомпозиции в

объектном анализе и проектировании. Различные подходы проведения объектно-ориентированного анализа.

#### **7. Унифицированный язык моделирования UML**

Предметы в UML. Отношения в UML. Диаграммы в UML. Механизмы расширения в UML. Различные точки зрения на диаграммы (концептуальная, спецификации, реализации).

#### **8. Модели требований**

Диаграммы Use Case. Актёры, прецеденты, и отношения в диаграммах Use Case. Спецификация элементов Use Case (основной и альтернативный потоки). Кооперации.

#### **9. Статические модели**

Диаграммы классов. Отношения в диаграммах классов. Организация свойств и операций. Множественность. Архитектура программного обеспечения. Каркасы и образцы проектирования. Классификация архитектур. Типы архитектур и их модели. Качество и выбор архитектуры.

#### **10. Динамические модели**

Моделирование поведения ПС. Диаграммы взаимодействия и сотрудничества. Диаграммы последовательности. Диаграммы состояний. Диаграммы деятельности.

#### **11. Модели реализации**

Диаграммы компонентов. Интерфейсы. Компоновка системы. Разновидности компонентов.

#### **12. Основы компонентной объектной модели**

COM-объекты, COM-интерфейсы. Технологии OLE и ActiveX. Повторное использование COM-объектов. Связывание объектов различного происхождения. Технология COM и объектно-ориентированное программирование. Реализация интерфейсов.

### **Раздел IV. Методы тестирования программного обеспечения**

#### **13. Основные методы тестирования**

Цели и задачи тестирования программного обеспечения. Возможности и ограниченность тестирования. Стратегии и критерии тестирования. Основные методы структурного тестирования (стратегия "белого ящика"). Тестирование циклов и потоков данных. Основные методы функционального тестирования (стратегия "чёрного ящика").

#### **14. Методика тестирования**

Тестирование элементов. Тестирование интеграции (нисходящее и восходящее тестирование). Тестирование правильности. Системное тестирование.

#### **15. Особенности тестирования объектно-ориентированных программ**

Методика объектно-ориентированного тестирования. Способы тестирования содержания и взаимодействия классов. Тестирование на основе состояний. Тестирование систем и подсистем.

### **Раздел V. Вопросы оценки качества программного обеспечения**

#### **16. Методы оценки сложности программного обеспечения**

Проблемы оценки качества и надежности программного обеспечения, пути решения проблем, перспективы. Стил программирования. Сложность программного обеспечения. Влияние сложности программного обеспечения на его

надежность. Вопросы оценки сложности. Влияние методов разработки на качество и надежность программного обеспечения.

### **17. Вопросы оценки сцепления и связности**

Оценка схем иерархии в структурных методах разработки (критерии Майерса). Сцепление и связность. Особенности сцепления и связности в объектно-ориентированном программном обеспечении (закон Деметера, видимость на уровне классов и объектов).

## **Литература**

### **Основная**

1. Брауде Э. Технология разработки программного обеспечения. – СПб.: Питер, 2004.
2. Г. Буч. Объектно-ориентированный анализ и проектирование, 2-е изд. Пер. с англ. – М.: Изд-во Бином, 1998.
3. Шлеер С., Меллор С. Объектно-ориентированный анализ: моделирование мира в состояниях. – Киев: ДИАЛЕКТИКА, 1994.
4. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. Пер. с англ. – М.: ДМК, 2000.
5. Зиглер К. Методы проектирования программных систем. – М.: Мир, 1985.
6. Дж. Фокс. Программное обеспечение и его разработка. – М.: Мир, 1985.
7. Д. Кинг. Создание эффективного ПО. – М.: Мир, 1991.
8. Л. Константайн. Конструирование программ. – М.: Мир, 1991.
9. Майерс Г. Искусство тестирования программ. М.: Финансы и статистика, 1982.
10. Канер Сэм. Тестирование программного обеспечения. Пер. с англ. – К.: Изд-во "ДиаСофт", 2000.
11. Бейзер Б. Тестирование чёрного ящика. Технология функционального тестирования. – СПб.: Питер, 2004.
12. Кантор М. Управление программными проектами. Пер. с англ. – М.: Издательский дом "Вильямс", 2002.
13. Дэвид Чепел. Технологии ActiveX и OLE. Пер. с англ. – М.: "Русская редакция", 1997.
14. Л. Харрис. Программирование OLE. – М.: БИНОМ, 1996.
15. Г.Н.Калынов. CASE структурный системный анализ. – М.: Изд-во «ЛОРИ», 1996.

### **Дополнительная**

16. Л. Ф. Дробушевич. Учебно-методическое пособие к общему курсу "Методы программирования" для студентов специальности "Информатика", Части I, II. –Мн.: Ротапринт Белгосуниверситета, 1999.
17. Л. Ф. Дробушевич. Технология программирования: Практ. пособие. – Мн.: БГУ, 2000.

18. *Л. Ф. Дробушевич*. Технология программирования: Моделирование программных систем в Rational Rose: Учеб. - метод. пособие. – Мн.: БГУ, 2002.
19. *Л. Ф. Дробушевич*. Технология программирования: Моделирование программных систем: Метод. указания и задания к лабораторным работам. – Мн.: БГУ, 2003.
20. *Ларман Крэг*. Применение UML и шаблонов проектирования. 2-е издание. Пер. с англ. – М.: Издательский дом "Вильямс", 2002.