

Государственное образовательное учреждение  
высшего профессионального образования

**Петрозаводский государственный университет**  
**кафедра информатики и математического обеспечения**

Методические указания для студента по дисциплине  
*Технология производства программного обеспечения*

## **1. Требования и условия сдачи зачета**

Аттестация по дисциплине включает индивидуальную аттестацию студента по его работе в минипроекте и теоретический зачет.

Для аттестации по минипроекту студенту необходимо регулярно и успешно работать в рамках выбранного минипроекта по разработке ПО. Основным критерием является комплекс оценок, полученных студентом в течении всего семестра в результате отчетов на лабораторных работах.

Аттестация по минипроекту является допуском к сдаче теоретического зачета. Список вопросов предоставляется. Основным критерием является умение студента интерпретировать и использовать (по крайней мере, потенциально) полученные теоретические знания на конкретных примерах, в первую очередь, на примере своего минипроекта.

## **2. Рекомендации и требования по выполнению лабораторных работ**

Лабораторное занятие предназначено для отчета студентов по разработке минипроекта — небольшого простого программного проекта.

Студенты выполняют минипроект группой разработчиков из 3-5 человек. Разбиение выполняет инструктор (преподаватель лабораторных работ) на 1-м занятии, учитывая, возможно, пожелания студентов. Основные работы по минипроекту выполняются студентами в рамках отведенных часов на самостоятельную работу (3ч/нед). Часть работ может выполняться во время лабораторного занятия (1ч/нед).

В качестве тематики рекомендуется разработка интерфейса пользователя к некоторому внешнему ПО. Последним может, например, выступать игра, вычислительная услуга или библиотека алгоритмов. Рекомендуются на выбор тип варианта (в порядке убывания предпочтения).

1. Свободно-распространяемое ПО других авторов, желательно, с открытым кодом.
2. Программы, разработанные студентами ранее в рамках других учебных дисциплин.
3. Внешнее ПО будет создаваться совместно с интерфейсом.

Окончательное определение внешнего ПО для проекта выполняет инструктор на 1-м занятии, учитывая, возможно, пожелания студентов. Возмо-

жен выбор темы, отличной от разработки интерфейса пользователя, например, по инициативе студентов, но при обязательном согласовании с инструктором. Ключевым требованием является простота для реализации, чтобы сложности кодирования не препятствовали освоению других аспектов производства ПО в условиях учебного проекта при существенных ограничениях на время и опыт.

Во время лабораторной работы проектная группа проводит собрание по минипроекту. Инструктор ведет часть этого собрания, посвященную отчету разработчиков, о выполнении ранее назначенных заданий. По результатам инструктор выставляет индивидуальные оценки каждому разработчику и проекту в целом (отсутствует, 0 - 3, опоздание, групповое опоздание и ряд других критериев), также используется система штрафных баллов за несданные в срок задачи. Эти оценки будут использованы при принятии решения об аттестации работы студента по минипроекту.

Задача студенческой группы состоит в выполнении всех стандартных стадий производства ПО в рамках учебного минипроекта. Особое внимание уделяется развитию навыков регулярной работы и отчетности, создания документации, нахождению и использованию адекватных задаче CASE-средств и умению абстрагирования от кодирования.

Используемые стадии проекта и ключевые документы представлены в графике лабораторных занятий. Каждая стадия завершается созданием соответствующего документа. План проекта создается параллельно со знакомством с системой и регулярно обновляется в течение всего процесса разработки.

Рекомендуемая структура документации представлена в отдельном документе. Для ряда документов предоставляются шаблоны.

### **3. Рекомендации по организации самостоятельной работы**

Выделенное для самостоятельной работы время студент тратит отчеты, где фиксирует краткую информацию о выполненных работах и потраченных ресурсах.

Необходимо сочетать как групповую (например, собрания), так и индивидуальную работу. Отдельно отслеживается эффективность работ.

### **4. Учебно-методические материалы по дисциплине**

#### **Список основной литературы**

1. Соммервилл И. *Инженерия программного обеспечения*. 6-е издание. : Пер. с англ. М.: Издательский дом «Вильямс», 2002. 624 с. ISBN 5-8459-0330-0.
2. Roger S. Pressman. *Software Engineering. A Practitioner's Approach*. European adapt., 5th ed. McGraw-Hill, 2000. 915 p. ISBN 0-07-709677-0.
3. Орлов С. А. *Технологии разработки программного обеспечения*. Учебное пос. 2-е изд. / С. А. Орлов. — СПб.: Питер, 2003. — 480 с.
4. Брауде Э. *Технология разработки программного обеспечения*. — СПб.: Питер, 2004. — 655 с.

## Список дополнительной литературы

4. Ларман К. *Применение UML и шаблонов проектирования*. : Пер. с англ. : Уч. пос. М.: Издательский дом «Вильямс», 2001. 496 с. ISBN 5-8459-0125-1.
5. Коналлен Д. *Разработка Web-приложений с использованием UML*. : Пер. с англ. М.: Издательский дом «Вильямс», 2001. 288 с. ISBN 5-8459-0203-7.
6. Аджиев В. *Мифы о безопасном ПО: уроки знаменитых катастроф* // Открытые системы, 1998. № 06. <http://www.osp.ru/os/1998/06/21.htm>
7. Фатрелл Р., Шафер Д., Шафер Л. *Управление программными проектами: достижение оптимального качества при минимуме затрат*. : Пер. с англ. — М.: Издательский дом «Вильямс», 2003. — 1136 с.