[NỘI DUNG](#_Toc68549)  [**GIỚI THIỆU** 5](#_Toc68549)

[Phòng thí nghiệm số 1](#_Toc68550)  [6](#_Toc68550)

[Xác định vòng đời thiết kế hệ thống máy tính](#_Toc68551)  [6](#_Toc68551)

[Phòng thí nghiệm số 2](#_Toc68552)  [25](#_Toc68552)

[Phòng thí nghiệm số 3](#_Toc68553)  [38](#_Toc68553)

[Tạo sơ đồ ca sử dụng](#_Toc68554)  [38](#_Toc68554)

[Phòng thí nghiệm số 4](#_Toc68555)  [52](#_Toc68555)

[Tạo sơ đồ tương tác](#_Toc68556)  [52](#_Toc68556)

[Phòng thí nghiệm số 5](#_Toc68557)  [58](#_Toc68557)

[Tạo sơ đồ lớp](#_Toc68558)  [58](#_Toc68558)

[Phòng thí nghiệm số 6](#_Toc68559)  [72](#_Toc68559)

[Tạo sơ đồ hoạt động](#_Toc68560)  [72](#_Toc68560)

[TÀI LIỆU THAM KHẢO](#_Toc68561)  [76](#_Toc68561)

**GIỚI THIỆU**

Sổ tay phương pháp này chứa mô tả chi tiết về công việc trong phòng thí nghiệm, mỗi hướng dẫn đều nhằm mục đích nắm vững và củng cố một chủ đề cụ thể:

1. Tìm kiếm thông tin trên Internet.
2. Các tính năng làm việc với các nguồn thông tin điện tử.
3. Hình thức trình bày và đặc điểm xử lý thông tin khoa học

# 

# Phòng thí nghiệm số 1

# Xác định vòng đời thiết kế hệ thống máy tính

**Mục đích công việc** : làm quen với các mô hình vòng đời của hệ thống thông tin, xác định ưu điểm và nhược điểm của mô hình, chọn mô hình xây dựng hệ thống thông tin cho từng nhiệm vụ và phần mềm của dự án, lập kế hoạch triển khai của cá nhân nhiệm vụ dự án.

**Thông tin lý thuyết ngắn gọn.**

### 1. Vòng đời của một hệ thống thông tin

Việc phát triển các hệ thống thông tin phức tạp (IS) là không thể nếu không có cách tiếp cận phương pháp luận được xem xét cẩn thận. Khái niệm vòng đời là một trong những khái niệm cơ bản của phương pháp thiết kế hệ thống thông tin.

***Vòng đời của một hệ thống thông tin*** là *một quá trình liên tục kể từ thời điểm đưa ra quyết định về sự cần thiết phải tạo ra nó cho đến khi hoàn thành hoạt động của nó.*

Quá trình thiết kế AIS được quy định bởi các tài liệu sau (tiêu chuẩn, phương pháp, mô hình):

* GOST 34.601-90. Có hiệu lực từ ngày 01/01/1992. Thiết lập các giai đoạn, giai đoạn tạo hệ thống tự động và đưa ra nội dung công việc ở từng giai đoạn. Các giai đoạn và giai đoạn công việc được quy định trong tiêu chuẩn tương ứng với *mô hình* vòng đời theo tầng.
* ISO/IEC 12207:1995. Một tiêu chuẩn quốc tế mô tả các quy trình vòng đời phần mềm. Chứa mô tả của hơn 220 tác phẩm cơ bản có thể được yêu cầu trong quá trình tạo IP. Tất cả các quy trình vòng đời phần mềm được chia thành ba nhóm lớn:
  + Các quy trình chính (mua hàng, cung cấp, phát triển,

vận hành, bảo trì);

* + Các quy trình hỗ trợ (tài liệu, quản lý cấu hình, đảm bảo chất lượng, giải quyết vấn đề, kiểm toán, chứng nhận, đánh giá chung, xác minh); ồ Quy trình tổ chức (tạo cơ sở hạ tầng, quản lý, đào tạo, cải tiến).
  + Để thực hiện các điều khoản của tiêu chuẩn, các công cụ phải được lựa chọn cùng nhau tạo thành một tổ hợp hỗ trợ công nghệ và tự động hóa vòng đời phần mềm được kết nối với nhau và không mâu thuẫn với bộ tài liệu quy định được lắp ráp sẵn. Để tạo thuận lợi cho việc áp dụng tiêu chuẩn vào thực tế, các tài liệu sau đã được tổ chức tiêu chuẩn hóa quốc tế xây dựng và phê duyệt: o ISO/IEC TR 15271:1998 – hướng dẫn áp dụng ISO/IEC 12207; ồ ISO/IEC TR 16326:1999 cung cấp hướng dẫn về quản lý dự án bằng ISO/IEC 12207.
* ISO/IEC 15288:2002. Một tiêu chuẩn quốc tế mô tả các quy trình vòng đời có thể có của các hệ thống do con người tạo ra. Nó được tạo ra dựa trên kinh nghiệm thiết kế hệ thống thông tin tự động, cũng như với sự tham gia của các chuyên gia từ nhiều lĩnh vực khác nhau: kỹ thuật hệ thống, lập trình, quản trị, quản lý chất lượng, bảo mật, v.v. Tiêu chuẩn này nhằm mục đích chứa tập hợp đầy đủ các quy trình có thể xảy ra trong vòng đời của hệ thống. Vì vậy, nhiệm vụ của nhà phát triển IS là tạo ra bộ mà anh ta cần - môi trường quy trình. Việc xem xét tiêu chuẩn lưu ý rằng nó không mô tả các phương pháp và thủ tục cần thiết để đảm bảo rằng các mục tiêu, mục đích và kết quả của các quy trình cụ thể được đáp ứng. Năm 2003, hướng dẫn áp dụng tiêu chuẩn (ISO/IEC TR 19760:2003) được ban hành. Hiện tại, công việc chuẩn bị phiên bản mới của tiêu chuẩn dòng 15288 vẫn tiếp tục.
* Rational Unified Process (RUP) là một khái niệm về phát triển phần mềm lặp (xoắn ốc) được đề xuất bởi Rational Software (hiện là một bộ phận của IBM). Vòng đời của IS bao gồm bốn giai đoạn: khởi động , nghiên cứu ( xây dựng ), thiết kế ( xây dựng ) và thực hiện ( chuyển tiếp ). Mỗi giai đoạn có thể chứa nhiều lần lặp. Ngoài ra, việc hoàn thành cả bốn giai đoạn không phải lúc nào cũng có nghĩa là hoàn thành công việc của dự án - sự phát triển của dự án có thể tiếp tục trong một chu kỳ mới. Là một phần của quá trình lặp lại, các mô hình nhất quán lẫn nhau sẽ được tạo ra, được mô tả bằng UML ( Ngôn ngữ mô hình hóa thống nhất ) được phát triển đặc biệt.
* Khung giải pháp của Microsoft (MSF). Phương pháp phát triển ứng dụng lặp tương tự như RUP. Nó cũng bao gồm bốn giai đoạn: phân tích, thiết kế, phát triển, ổn định và liên quan đến việc sử dụng mô hình hướng đối tượng. So với RUP, nó tập trung hơn vào việc phát triển IP cho doanh nghiệp.
* Lập trình cực đoan (XP). Extreme Programming là phương pháp mới nhất trong số các phương pháp đang được xem xét (những ý tưởng đầu tiên được hình thành vào giữa những năm 1990). Các nguyên tắc cơ bản: làm việc nhóm, tương tác hiệu quả giữa khách hàng và nhà thầu trong toàn bộ quá trình phát triển Khu công nghiệp, sử dụng các nguyên mẫu được cải tiến liên tục, đạt được tính linh hoạt phát triển tối đa

(thích ứng với những yêu cầu thay đổi của khách hàng).

**2. Mô hình vòng đời IS.**

Mô hình vòng đời IS được hiểu là một cấu trúc xác định trình tự thực hiện và mối quan hệ giữa các tiến trình hành động, nhiệm vụ trong suốt vòng đời.

***Mô hình vòng đời IS*** là *cấu trúc mô tả các quy trình, hành động và nhiệm vụ được thực hiện trong quá trình phát triển, vận hành và bảo trì trong toàn bộ vòng đời của hệ thống.*

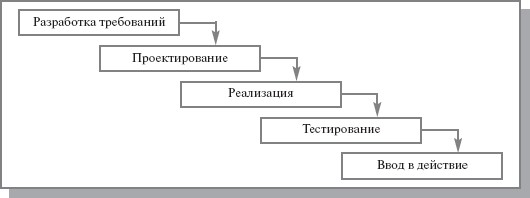
Việc lựa chọn mô hình vòng đời phụ thuộc vào tính chất cụ thể, quy mô, độ phức tạp của dự án và tập hợp các điều kiện mà AIS được tạo ra và vận hành.

Phù hợp với các mô hình vòng đời phần mềm đã biết, các mô hình vòng đời IS được xác định - *xếp tầng, lặp, xoắn ốc.*

***I. Mô hình xếp tầng*** mô tả cách tiếp cận cổ điển để phát triển hệ thống trong bất kỳ lĩnh vực chủ đề nào; được sử dụng rộng rãi vào những năm 1970 và 80.

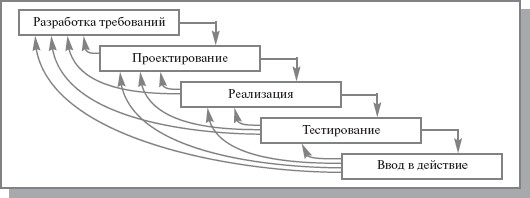
Mô hình xếp tầng cung cấp khả năng tổ chức công việc tuần tự và đặc điểm chính của mô hình là phân chia tất cả công việc thành các giai đoạn. Việc chuyển từ giai đoạn trước sang giai đoạn tiếp theo chỉ xảy ra sau khi hoàn thành xong tất cả công việc của giai đoạn trước.

Có ***năm***các giai đoạn phát triển bền vững, thực tế độc lập với lĩnh vực chủ đề (Hình 1.1):



**Hình.1.1** . Giai đoạn phát triển

* Ở giai đoạn đầu tiên, việc nghiên cứu vấn đề được thực hiện và các yêu cầu của khách hàng được hình thành. Kết quả của giai đoạn này là một đặc tả kỹ thuật (nhiệm vụ phát triển) đã được thống nhất với tất cả các bên quan tâm.
* Trong giai đoạn thứ hai, theo yêu cầu của thông số kỹ thuật, một số giải pháp thiết kế nhất định sẽ được phát triển. Kết quả là một bộ tài liệu thiết kế xuất hiện.
* Giai đoạn thứ ba là thực hiện dự án; Về cơ bản, phát triển phần mềm (viết mã) theo đúng quyết định thiết kế của giai đoạn trước. Phương pháp thực hiện không có tầm quan trọng cơ bản trong trường hợp này. Kết quả của giai đoạn này là một sản phẩm phần mềm hoàn chỉnh.
* Ở giai đoạn thứ tư, phần mềm thu được sẽ được kiểm tra xem có tuân thủ các yêu cầu nêu trong thông số kỹ thuật hay không. Hoạt động thử nghiệm giúp xác định nhiều loại thiếu sót tiềm ẩn khác nhau xuất hiện trong điều kiện vận hành thực tế của IS.
* Giai đoạn cuối cùng là bàn giao dự án đã hoàn thành.
* Ở mỗi giai đoạn, một bộ tài liệu thiết kế hoàn chỉnh được tạo ra đáp ứng các tiêu chí về tính đầy đủ và nhất quán. Ở giai đoạn cuối, tài liệu dành cho người dùng được phát triển, bao gồm tất cả các loại hỗ trợ AIS được cung cấp theo tiêu chuẩn (tổ chức, thông tin, phần mềm, kỹ thuật, v.v.). Việc thực hiện nhất quán các giai đoạn công việc cho phép bạn lập kế hoạch về ngày hoàn thành và các chi phí liên quan.
* Đồng thời, đối với mô hình xếp tầng, cần lưu ý sự chậm trễ đáng kể trong việc đạt được kết quả, sự phức tạp của công việc song song trong dự án và sự phức tạp của quản lý dự án, và do đó, mức độ rủi ro cao và không đáng tin cậy của đầu tư vào IP. Ngoài ra, theo quy luật, những sai sót và thiếu sót ở bất kỳ giai đoạn nào cũng xuất hiện ở các giai đoạn công việc tiếp theo, dẫn đến cần phải hoàn trả.
* II. Mô hình lặp lại bao gồm một loạt các chu trình (bước) ngắn từ lập kế hoạch, thực hiện, nghiên cứu, hành động (Hình 1.2). Quá trình phát triển IS được thực hiện lặp đi lặp lại với các vòng phản hồi giữa các giai đoạn. Những điều chỉnh giữa các giai đoạn giúp có thể tính đến ảnh hưởng thực tế lẫn nhau của kết quả phát triển ở các giai đoạn khác nhau; Thời gian tồn tại của mỗi giai đoạn kéo dài trong toàn bộ thời kỳ phát triển.



**Hình.1.2.** Mô hình lặp

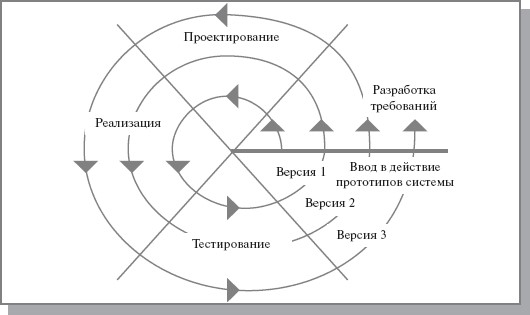
Việc tạo ra các hệ thống thông tin phức tạp liên quan đến việc phối hợp các giải pháp thiết kế thu được trong quá trình thực hiện các nhiệm vụ riêng lẻ. Cách tiếp cận từ dưới lên để thiết kế đòi hỏi sự lặp lại như vậy khi các giải pháp thiết kế cho các nhiệm vụ riêng lẻ được kết hợp thành các giải pháp hệ thống chung. Đồng thời, cần phải rà soát lại những yêu cầu đã hình thành trước đó.

Trong mô hình lặp, các điều chỉnh giữa các giai đoạn đảm bảo sự phát triển ít sử dụng nhiều lao động hơn so với mô hình thác nước.

Đồng thời, thời gian tồn tại của từng giai đoạn kéo dài trong toàn bộ thời gian làm việc; do số lần lặp lại lớn, sự khác biệt trong việc thực hiện các giải pháp thiết kế và tài liệu có thể nảy sinh ở giai đoạn thực hiện ; .

***III*** *.* ***Mô hình xoắn ốc*** , không giống như mô hình xếp tầng, nhưng tương tự như mô hình trước đó, giả định một quá trình phát triển IS lặp đi lặp lại (Hình 1.3). Đồng thời, tầm quan trọng của các giai đoạn ban đầu, chẳng hạn như phân tích và thiết kế, tăng lên, tại đó tính khả thi của các giải pháp kỹ thuật được kiểm tra và chứng minh bằng cách tạo ra các nguyên mẫu.

Mỗi lần lặp lại thể hiện một chu trình phát triển hoàn chỉnh, dẫn đến việc phát hành phiên bản bên trong hoặc bên ngoài của sản phẩm (hoặc một tập hợp con của sản phẩm cuối cùng), được cải tiến từ lần lặp này sang lần lặp khác để trở thành một hệ thống hoàn chỉnh:



**Hình.1.3.** Mô hình xoắn ốc

Do đó, mỗi vòng xoắn ốc tương ứng với việc tạo ra một đoạn hoặc phiên bản của sản phẩm phần mềm, các mục tiêu và đặc điểm của dự án được làm rõ, chất lượng của dự án được xác định và công việc được lên kế hoạch cho vòng xoắn ốc tiếp theo. Mỗi lần lặp lại nhằm mục đích xác định sâu hơn và nhất quán các chi tiết của dự án, từ đó chọn được phương án hợp lý cho việc triển khai cuối cùng.

Sử dụng mô hình xoắn ốc cho phép bạn chuyển sang giai đoạn tiếp theo của dự án mà không cần đợi giai đoạn hiện tại hoàn thành - công việc chưa hoàn thành có thể được hoàn thành ở lần lặp tiếp theo. Mục tiêu chính của mỗi lần lặp lại là tạo ra một sản phẩm hoạt động được để trình diễn cho người dùng càng nhanh càng tốt. Do đó, quá trình làm rõ và bổ sung dự án được đơn giản hóa đáng kể.

Cách tiếp cận xoắn ốc trong phát triển phần mềm khắc phục hầu hết nhược điểm của mô hình thác nước, đồng thời cung cấp một số tính năng bổ sung, giúp quá trình phát triển trở nên linh hoạt hơn. Khi sử dụng mô hình xoắn ốc, việc thực hiện các thay đổi đối với dự án khi yêu cầu của khách hàng thay đổi được đơn giản hóa đáng kể, mức độ rủi ro giảm (mức độ rủi ro tối đa khi bắt đầu phát triển dự án, khi tiến triển phát triển nó giảm dần), tính linh hoạt cao hơn trong quản lý dự án được cung cấp do khả năng thực hiện các thay đổi chiến thuật đối với sản phẩm đã phát triển, khả năng cải thiện quá trình phát triển - do kết quả phân tích ở cuối mỗi lần lặp lại, những thay đổi trong tổ chức phát triển được đánh giá; trong lần lặp tiếp theo, nó sẽ cải thiện, việc sử dụng lại các thành phần trở nên dễ dàng hơn, vì việc xác định các phần chung của dự án khi chúng đã được phát triển một phần sẽ dễ dàng hơn nhiều so với việc cố gắng tách chúng ra ngay từ đầu dự án. Mô hình xoắn ốc cho phép hệ thống ổn định và đáng tin cậy hơn. Điều này là do khi hệ thống phát triển, các lỗi và điểm yếu sẽ được phát hiện và sửa chữa ở mỗi lần lặp. Đồng thời, các thông số hiệu quả tới hạn được điều chỉnh, trong trường hợp mô hình tầng chỉ có sẵn trước khi hệ thống được triển khai. Việc thu hút người dùng tham gia vào quá trình thiết kế và sao chép ứng dụng cho phép bạn nhận được nhận xét và bổ sung trực tiếp cho các yêu cầu trong quá trình thiết kế ứng dụng, giảm thời gian phát triển. Đại diện khách hàng có cơ hội kiểm soát quá trình tạo ra hệ thống và tác động đến nội dung chức năng của nó. Kết quả là việc vận hành một hệ thống có tính đến phần lớn nhu cầu của khách hàng.

Tuy nhiên, mô hình xoắn ốc của thiết kế vi mạch thường có chi phí cao (vì vậy sẽ hợp lý khi sử dụng nó cho các hệ thống phức tạp và đắt tiền). Mô hình có cấu trúc phức tạp, có thể gây khó khăn cho việc sử dụng trong thực tế đối với các chuyên gia và khách hàng chưa qua đào tạo. Vòng xoáy có thể tiếp tục vô thời hạn, vì mỗi phản hồi của khách hàng đối với phiên bản đã tạo có thể tạo ra một chu kỳ làm việc mới. Một số lượng lớn các giai đoạn trung gian làm phức tạp việc duy trì tài liệu dự án. Có thể khó xác định thời điểm chuyển sang lần lặp tiếp theo của chu kỳ. Thông thường, các hạn chế về thời gian được áp đặt đối với việc thực hiện vòng lặp và từng giai đoạn của nó.

Trong một số trường hợp, việc sử dụng mô hình xoắn ốc là không thể hoặc bị hạn chế vì không thể sử dụng/thử nghiệm một sản phẩm có chức năng chưa hoàn chỉnh (ví dụ: phát triển quân sự, năng lượng hạt nhân, v.v.). Việc triển khai lặp lại theo từng giai đoạn của hệ thống thông tin doanh nghiệp thường liên quan đến những khó khăn về mặt tổ chức (truyền dữ liệu, tích hợp hệ thống, thay đổi quy trình kinh doanh, chính sách kế toán, đào tạo người dùng).

Chi phí nhân công trong quá trình thực hiện lặp lại theo từng giai đoạn hóa ra cao hơn nhiều và việc quản lý quá trình thực hiện một cách thiếu hiểu biết có thể phủ nhận tất cả các kết quả thu được. Vì lý do này, ở giai đoạn triển khai, các mô hình lặp lại thường được loại bỏ và giới thiệu hệ thống “một lần và mãi mãi”.

### 3. CÔNG NGHỆ THIẾT KẾ

***Công nghệ thiết kế vi mạch*** *là tập hợp các phương pháp và phương tiện để thiết kế hệ thống thông tin tự động cũng như các phương pháp và phương tiện để tổ chức thiết kế (quản lý quá trình tạo và hiện đại hóa một dự án hệ thống thông tin).*

Chủ đề của công nghệ thiết kế được lựa chọn phải phản ánh các quá trình thiết kế có liên quan lẫn nhau ở tất cả các giai đoạn của vòng đời IS.

Các yêu cầu chính đối với công nghệ thiết kế được lựa chọn như sau:

· dự án được tạo ra bằng công nghệ này phải đáp ứng yêu cầu của khách hàng;

· Công nghệ phải phản ánh càng nhiều càng tốt tất cả các giai đoạn của vòng đời dự án;

· Công nghệ phải đảm bảo tối thiểu chi phí lao động và chi phí cho việc thiết kế và hỗ trợ dự án;

· Công nghệ sẽ góp phần nâng cao năng suất của các nhà thiết kế;

Công nghệ phải đảm bảo độ tin cậy của quá trình

thiết kế và vận hành dự án;

· Công nghệ sẽ tạo điều kiện dễ dàng cho việc bảo trì tài liệu dự án.

Các phương pháp thiết kế AIS có thể được phân loại theo mức độ sử dụng các công cụ tự động hóa, giải pháp thiết kế tiêu chuẩn và khả năng thích ứng với những thay đổi dự kiến. ***Theo mức độ tự động hóa, chúng được phân biệt:*** *thiết kế thủ công* , trong đó việc thiết kế các thành phần AIS được thực hiện mà không sử dụng các công cụ phần mềm đặc biệt; lập trình được thực hiện bằng ngôn ngữ thuật toán;

*thiết kế máy tính* , trong đó việc tạo hoặc cấu hình (cài đặt) các giải pháp thiết kế được thực hiện bằng các công cụ phần mềm đặc biệt.

***Dựa trên mức độ sử dụng các giải pháp thiết kế tiêu chuẩn, chúng được phân biệt:***

*ban đầu (riêng lẻ)* , khi thiết kế

các giải pháp được phát triển từ đầu phù hợp với yêu cầu của AIS; *thiết kế tiêu chuẩn* , giả sử cấu hình AIS từ

giải pháp thiết kế tiêu chuẩn làm sẵn (mô-đun phần mềm).

***về mức độ thích ứng của các giải pháp thiết kế*** :

*tái thiết* - điều chỉnh các giải pháp thiết kế được thực hiện bằng cách xử lý các thành phần có liên quan (lập trình lại các mô-đun phần mềm); *tham số hóa* - giải pháp thiết kế được cấu hình theo các tham số đã chỉ định và có thể thay đổi;

*tái cấu trúc mô hình* - mô hình của lĩnh vực chủ đề được thay đổi,

dẫn đến việc tự động cải cách các giải pháp thiết kế.

Sự kết hợp của các đặc điểm khác nhau của việc phân loại các phương pháp thiết kế quyết định bản chất của công nghệ thiết kế AIS được sử dụng. Có hai loại công nghệ thiết kế chính: *chuẩn* và *công nghiệp* . Công nghệ thiết kế công nghiệp được chia thành hai lớp con: thiết kế *tự động* (sử dụng công nghệ CASE) và thiết kế *tiêu chuẩn* (hướng tham số hoặc hướng mô hình).

Công nghệ CASE là một phương pháp thiết kế IS, đồng thời là một bộ công cụ cho phép bạn lập mô hình trực quan một lĩnh vực chủ đề, phân tích mô hình này ở tất cả các giai đoạn phát triển và bảo trì IS cũng như phát triển ứng dụng phù hợp với nhu cầu thông tin của người dùng. Hầu hết các công cụ CASE hiện có đều dựa trên các phương pháp phân tích và thiết kế hướng đối tượng (hầu hết) hoặc cấu trúc, sử dụng các đặc tả dưới dạng sơ đồ hoặc văn bản để mô tả các yêu cầu bên ngoài, mối quan hệ giữa các mô hình hệ thống, động lực hành vi hệ thống và kiến trúc phần mềm.

Các công cụ CASE cho phép bạn tạo ra không chỉ một sản phẩm thực tế đã sẵn sàng để sử dụng mà còn đảm bảo quá trình phát triển “chính xác”. Mục tiêu chính của công nghệ là tách thiết kế phần mềm khỏi mã hóa, lắp ráp, thử nghiệm và “giấu” càng nhiều càng tốt với người dùng tương lai tất cả các chi tiết về phát triển và vận hành phần mềm. Đồng thời, hiệu quả làm việc của nhà thiết kế tăng lên đáng kể: thời gian phát triển giảm, số lỗi phần mềm giảm và các mô-đun phần mềm có thể được sử dụng trong các quá trình phát triển trong tương lai.

Là ví dụ về các công cụ CASE phổ biến, chúng tôi chỉ ra phần mềm của Computer Associates, IBM- Rational Software và Oracle:

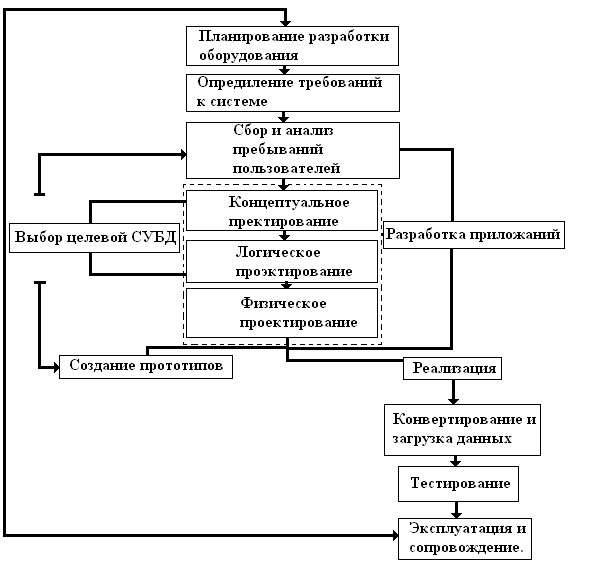
* BPwin – mô hình hóa quy trình kinh doanh;
* ERwin – mô hình hóa cơ sở dữ liệu và kho dữ liệu;
* ERwin Examiner – kiểm tra cấu trúc và mô hình DBMS được tạo trong Erwin ;
* Người mẫuMart – Môi trường làm việc nhóm

nhà thiết kế;

* Paradigm Plus – mô hình hóa ứng dụng và tạo mã đối tượng;
* Rational Rose – mô hình hóa các quy trình kinh doanh và các thành phần ứng dụng;
* Rational Suite AnalystStudio – gói dành cho nhà phân tích dữ liệu;
* Oracle Designer (một phần của Oracle9i Developer Suite) là một công cụ có tính ứng dụng cao để thiết kế hệ thống phần mềm và cơ sở dữ liệu, triển khai công nghệ CASE và phương pháp riêng của Oracle - CDM. Là một công cụ CASE phức tạp, thật hợp lý khi sử dụng nó khi nhắm mục tiêu vào dòng sản phẩm Oracle.

### 4. CÁC GIAI ĐOẠN TẠO IP

Tiêu chuẩn ISO/IEC 12207 xác định cấu trúc vòng đời bao gồm các quy trình, hoạt động và nhiệm vụ phải được thực hiện trong quá trình tạo hệ thống thông tin (Hình 1.4):



**Hình.1.4.** Cấu trúc vòng đời

Các giai đoạn tạo AIS (ISO/IEC 15288)

|  |  |
| --- | --- |
| Sân khấu | *Sự miêu tả* |
| Sự hình thành khái niệm | * nhu cầu phân tích, lựa chọn ý tưởng và giải pháp thiết kế |
| Phát triển | * thiết kế hệ thống |
| Thực hiện | * hệ thống sản xuất |
| Hoạt động | * vận hành và sử dụng hệ thống |
| Ủng hộ | * đảm bảo hoạt động của hệ thống |
| Ngừng hoạt động | * ngừng sử dụng, lưu trữ hệ thống |

Các giai đoạn tạo hệ thống theo yêu cầu của GOST 34.601-90 “Hệ thống tự động. Các giai đoạn sáng tạo.":

|  |  |
| --- | --- |
| ***Sân khấu*** | ***Sự miêu tả*** |
| Hình thành các yêu cầu đối với AIS | * kiểm tra cơ sở và chứng minh sự cần thiết phải tạo AIS; * hình thành các yêu cầu của người dùng đối với AIS; * chuẩn bị báo cáo về công việc đã thực hiện và các thông số kỹ thuật và chiến thuật để phát triển. |
| Phát triển khái niệm AIS: | * nghiên cứu đối tượng tự động hóa; * thực hiện các hoạt động khoa học cần thiết * công trình nghiên cứu; * phát triển các tùy chọn khái niệm AIS đáp ứng yêu cầu của người dùng; * chuẩn bị báo cáo và phê duyệt ý tưởng. |
| Thông số kỹ thuật: | * phát triển và phê duyệt các thông số kỹ thuật để tạo ra AIS |
| Thiết kế dự thảo: | * phát triển các giải pháp thiết kế sơ bộ cho hệ thống và các bộ phận của nó; * phát triển tài liệu dự thảo cho AIS và các bộ phận của nó. |
| Dự án kỹ thuật | * phát triển tài liệu cho AIS và các bộ phận của nó; · phát triển và thực hiện tài liệu về việc cung cấp linh kiện; * phát triển nhiệm vụ thiết kế ở các phần lân cận của dự án. * phát triển các giải pháp thiết kế cho hệ thống và các bộ phận của nó; |
| Tài liệu làm việc: | * phát triển tài liệu làm việc cho AIS và các bộ phận của nó; * phát triển và điều chỉnh các chương trình. |
| Vận hành : | * chuẩn bị một đối tượng tự động hóa; * đào tạo nhân sự; * trang bị cho AIS các sản phẩm được cung cấp (phần mềm và phần cứng, hệ thống phần mềm và phần cứng, sản phẩm thông tin); * xây dựng và lắp đặt ; * công tác nghiệm thu; * tiến hành các thử nghiệm sơ bộ; * tiến hành vận hành thử nghiệm; * thực hiện các thử nghiệm chấp nhận. |
| Hỗ trợ AIS: | * thực hiện công việc theo đúng nghĩa vụ bảo hành; * dịch vụ sau bảo hành. |

**Chi tiết một số công việc theo từng giai đoạn.**

|  |  |
| --- | --- |
| ***Tên sân khấu*** | ***Nội dung công việc*** |
| Quy hoạch phát triển IS | * tìm kiếm và biện minh cho các phương pháp (phương pháp) tối ưu của tổ chức trong điều kiện của một doanh nghiệp cụ thể, * xác định ranh giới ứng dụng IP |
| Xác định các yêu cầu về thành phần và phân phối thông tin (bao gồm cả việc tổ chức cơ sở dữ liệu) | * xác định thành phần người dùng và thiết lập nhiệm vụ cho từng người trong quá trình thiết kế, vận hành hệ thống thông tin ; * xác định các quy trình và tài liệu công việc hiện có |
| Phát triển một mô tả thống nhất về các đặc điểm của đối tượng | * thu thập và phân tích các yêu cầu mô tả đối tượng thông tin từ tất cả những người sử dụng thông tin tiềm năng |
| Phát triển và nghiên cứu các mô hình dự án IS dưới dạng DBMS | * mô hình hóa cơ sở dữ liệu khái niệm, logic và vật lý, bao gồm: * xác định các đối tượng cơ sở dữ liệu; * xác định các kết nối, năng lực của họ và * nghĩa vụ, thuộc tính; * Phân tích lại thực thể: mối quan hệ giữa thực thể và miền, quy trình công việc ảnh hưởng đến thực thể, tương tác giữa các thực thể, thuộc tính, phân tích miền, lựa chọn loại ràng buộc; * chuẩn hóa dữ liệu; * lựa chọn kiến trúc cơ sở dữ liệu hệ thống; * bảng; thông tin liên lạc; chỉ số; * yêu cầu; * bảo vệ dữ liệu: mức độ bảo vệ; * theo dõi và ghi lại các sự kiện trong IS. |
| Thiết kế giao diện người dùng | * lựa chọn mô hình giao diện; * kiến trúc giao diện người dùng: * hỗ trợ quy trình làm việc; * các lớp giao diện người dùng; * tiêu chí giao diện người dùng; * mối quan hệ giữa các thực thể và hình thức; * lựa chọn điều khiển; * hỗ trợ tính toàn vẹn dữ liệu; * báo cáo; * hỗ trợ người dùng |
| Biện minh và lựa chọn hệ thống phần mềm để phát triển  IP | * đánh giá chi phí dự kiến của việc phát triển và vận hành IS trong môi trường doanh nghiệp |
| Phát triển dự án  (nguyên mẫu) là | * tạo nguyên mẫu ( mô hình IP ) bằng các công cụ thiết kế trực quan (ví dụ: truy cập ) |
| Phát triển và  Triển khai IP | * phát triển các phần máy chủ và máy khách của IS dưới dạng cơ sở dữ liệu và chương trình ứng dụng |
| Đang tải dữ liệu | * điền thông tin vào cơ sở dữ liệu |
| Kiểm tra | * kiểm tra hoạt động của hệ thống và loại bỏ * lỗi ứng dụng xảy ra |
| Vận hành và  hộ tống | * phát triển các biện pháp tổ chức để triển khai hệ thống thông tin , giám sát hoạt động của hệ thống và nếu cần thiết, thực hiện các thay đổi đối với các ứng dụng phần mềm. |

**Bài tập.**

1. Làm quen với thông tin lý thuyết về

công việc trong phòng thí nghiệm

1. Xác định ưu điểm và nhược điểm của các mô hình vòng đời IS
2. Chọn và biện minh cho việc lựa chọn mô hình vòng đời IS để hoàn thành nhiệm vụ thiết kế riêng lẻ.
3. Tạo kế hoạch xây dựng IS cho một nhiệm vụ dự án riêng lẻ bằng cách sử dụng phần mềm.

**Nội dung của báo cáo.**

**Lĩnh vực chủ thể là một lĩnh vực hoạt động, một thực thể kinh tế.   
Phát triển hệ thống thông tin cho: (lĩnh vực chủ đề cụ thể)**

**Vòng đời là khoảng thời gian kể từ thời điểm vấn đề được đặt ra cho đến khi hệ thống thông tin được triển khai.**

Báo cáo được lập trên tờ A4 và bao gồm

1. Tiêu đề công việc
2. Mục đích của công việc
3. Bảng hoàn chỉnh “Ưu điểm và nhược điểm của các mô hình vòng đời IS”

|  |  |  |
| --- | --- | --- |
| *Mô hình vòng đời IS* | *Thuận lợi* | *sai sót* |
| Thác nước |  |  |
| Lặp đi lặp lại |  |  |
| xoắn ốc |  |  |

1. Mô hình vòng đời IS đã chọn và lý do lựa chọn nhiệm vụ dự án cá nhân để phát triển IS.
2. Kế hoạch xây dựng IS của một dự án cá nhân được giao theo mẫu:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *p/p* | *Tên sân khấu*  *(giai đoạn) công việc* | *Nội dung công việc* | *Kết quả công việc* | *Phần mềm đã qua sử dụng* |
|  |  |  |  |  |

# 

# Phòng thí nghiệm số 2

### Phát triển mô hình phân rã khái niệm và quy trình (tiêu chuẩn IDEF0)

**Mục đích công việc:** rèn luyện kỹ năng xây dựng sơ đồ tình huống.

**Bài tập:**

1. Tạo một mô hình khái niệm
2. Tạo sơ đồ phân rã

### Tài liệu lý thuyết

Phương pháp **IDEF0** được thiết kế để mô hình hóa các hoạt động của một tổ chức. Ở giai đoạn đầu phát triển dự án, một mô hình được thiết kế để mô tả các quy trình kinh doanh và quy trình công nghệ hiện có trong doanh nghiệp theo nguyên tắc “AS - IS” (“As Is”) và quan trọng là nó đại diện cho doanh nghiệp từ góc nhìn của những nhân viên làm việc ở đó và biết tường tận mọi sắc thái, kể cả những sắc thái không chính thức. AS-IS – “những gì chúng ta đang làm hôm nay” trước khi chuyển sang “những gì chúng ta sẽ làm vào ngày mai”.

**Một mô hình hoạt động** hay nói cách khác là **một mô hình chức năng. Mô hình chức năng** xem hệ thống như một tập hợp *các hành động,* trong đó mỗi hành động biến đổi một số *đối tượng* hoặc *tập hợp các đối tượng.* Công nghệ **IDEF 0** sử dụng nguyên tắc ***phân rã chức năng*** của hệ thống (chia hệ thống thành các mảnh). **Nguyên tắc phân rã** có nghĩa là một mô hình chức năng cần được xây dựng theo *quy tắc “từ trên xuống dưới”,* từ dạng *chung* của mô hình đến các mô hình *cụ thể* . Vì vậy, thông thường một mô hình hàm để giải một bài toán là một tập hợp *các mô hình hàm từng phần.*

Các mô hình chức năng làm nổi bật các hành động bằng cách biểu diễn chúng dưới dạng một *khối phần tử đặc biệt* . **Khối -** yếu tố cấu trúc chính

chức năng , được biểu diễn bằng đồ họa là **sơ đồ.** Tên của hành động **là một danh từ** hoặc **động từ.** Kết quả của việc phân rã mô hình là một tập hợp các sơ đồ được sắp xếp theo thứ bậc và liên kết với nhau được tạo ra. Mỗi sơ đồ là một đơn vị mô tả hệ thống và được đặt trên một trang riêng.

Phương pháp **IDEF 0** dựa trên bốn khái niệm chính:

khối chức năng (nút), cung giao diện, phân rã, bảng chú giải.

### Khối chức năng

cơ bản của công nghệ **IDEF 0** là

khái niệm về **khối chức** năng Nó nhằm mục đích đại diện cho một ***hoạt động cụ thể ( Hoạt động )*** , đại diện cho một số ***chức năng cụ thể*** trong hệ thống được đề cập. Ngược lại, chức năng này có nghĩa là một số hành động (tập hợp các hành động) có mục tiêu cố định và dẫn đến một số kết quả cuối cùng.

**Một khối chức năng** được biểu diễn bằng một hình chữ nhật, các cạnh của nó có các giá trị sau:

* Mặt trên là quản lý.
* Mặt dưới là cơ chế.
* Bên phải là lối ra.
* Bên trái là lối vào.

Một khối chức năng có một tên, được xác định bằng tâm trạng bằng lời nói hoặc một danh từ bằng lời nói. Sự tương tác giữa một hành động và thế giới xung quanh nó, bao gồm cả các hành động khác, được hiển thị bằng các cung giao diện (mũi tên).

### Cung giao diện

**Một cung giao diện** thể hiện một thành phần hệ thống được **xử lý bởi** một khối chức năng hoặc **có tác động** đến hoạt động (chức năng) được đại diện bởi khối chức năng đó. **Cung giao diện** được mô tả dưới dạng một mũi tên, biểu thị ***một phương tiện*** cung cấp việc truyền dữ liệu hoặc đối tượng từ hoạt động này sang hoạt động khác. Mũi tên mô tả sự tương tác của tác phẩm với thế giới bên ngoài và với nhau, thể hiện một số thông tin và được gọi là ***danh từ.***

Tên của mũi tên cho biết ***vai trò của nó*** (tập hợp các vai trò có thể được chỉ định - ***ICOM*** ):

Đầu vào của khối chức năng là ***I*** *nput* .

Kiểm soát *–* ***Kiểm soát*** .

Đầu ra của khối chức năng là *đầu ra* ***O.***

Cơ chế *–* ***Cơ chế*** *.*

**Đầu vào** là vật liệu hoặc thông tin được một công việc sử dụng hoặc biến đổi để tạo ra kết quả (đầu ra **) .** Có thể không có mũi tên vào.

**Kiểm soát** - Các quy tắc, chính sách, thủ tục hoặc tiêu chuẩn chi phối công việc. Nó ảnh hưởng đến công việc nhưng không bị công việc biến đổi. Mũi tên được vẽ khi đi vào mép trên của tác phẩm. Mỗi khối chức năng phải có ít nhất một mũi tên điều khiển.

Thường rất khó để xác định xem dữ liệu là đầu vào hay điều khiển. Nếu dữ liệu trong tác phẩm bị thay đổi hoặc xử lý thì rất có thể đây là dữ liệu đầu vào, còn nếu không thì đó là dữ liệu kiểm soát. Nếu khó xác định trạng thái của mũi tên, nên vẽ mũi tên điều khiển.

**Đầu ra –** vật liệu hoặc thông tin được tạo ra bởi một công việc **.** Cần có ít nhất một mũi tên thoát, xuất phát từ phía bên phải của tác phẩm.

**Cơ chế thực thi ( Mechanism )** – nguồn lực thực hiện công việc (ví dụ: nhân sự, máy móc, thiết bị, v.v.). Mũi tên được vẽ khi đi vào cạnh dưới của tác phẩm. Mũi tên cơ chế có thể không được hiển thị.

Nói chung, khối chức năng được hiển thị trong Hình. 2.1.



**Cơm. 2.1** Khối chức năng

Trong hình. 2.1 tất cả các cung giao diện được hiển thị dưới dạng mũi tên được đặt tên. Theo yêu cầu của tiêu chuẩn, mỗi khối chức năng phải có ít nhất một đầu ra và một điều khiển, vì mỗi nhiệm vụ (quy trình) phải có ít nhất một đầu ra và ít nhất một quy tắc để giải quyết nó. Cung giao diện “Cơ chế” có thể không được mô tả.

**Một mô hình chức năng** được xây dựng từ một số khối chức năng được kết nối bằng các cung giao diện theo cách thức được yêu cầu .

Xin lưu ý rằng các mũi tên có thể phân nhánh để tạo ra các kết nối cần thiết giữa các khối chức năng.

*Đầu vào của* một khối chức năng không chỉ có thể là *đầu ra* của một khối chức năng khác mà còn có thể là *bộ điều khiển* hoặc thậm chí là *một cơ chế của khối chức năng đó.* Kết quả là, mô hình chức năng có thể chứa nhiều quy trình khác nhau, khá phức tạp và khác thường để giải quyết các vấn đề trong hệ thống thông tin.

### Tạo sơ đồ bằng công nghệ IDEF0

Khi phát triển mô hình kinh doanh của một tổ chức, nên sử dụng ba loại sơ đồ:

* ***Sơ đồ loại I – Sơ đồ bối cảnh*** (chỉ có thể có một) – phần trên cùng của cấu trúc cây, thể hiện mức độ mô tả trừu tượng nhất của hệ thống và sự tương tác của nó với môi trường bên ngoài.

Nó định nghĩa ***một hàm ngữ cảnh* ;**

* ***Sơ đồ loại II – Sơ đồ phân rã*** .

Sơ đồ phân tích được thiết kế để trình bày chi tiết công việc và bao gồm công việc *anh em* , tức là công việc *con* có chung công việc *cha* . Công việc cấp dưới cũng giống như công việc cấp trên nhưng chi tiết hơn. Các sơ đồ được nhà phân tích tạo ra để tiến hành một buổi kiểm tra, tức là thảo luận về sơ đồ với một chuyên gia về chủ đề đó.

Sau mỗi phiên phân tích, các phiên kiểm tra được tiến hành - các chuyên gia về chủ đề này chỉ ra sự tuân thủ của các quy trình kinh doanh thực tế với các sơ đồ đã tạo. Mọi mâu thuẫn được tìm thấy đều được sửa chữa và chỉ sau khi vượt qua bài kiểm tra *mà không có bất kỳ nhận xét nào* thì phiên phân tích tiếp theo mới có thể bắt đầu. Điều này đảm bảo rằng mô hình phù hợp với quy trình kinh doanh thực tế ở mọi cấp độ của mô hình.

Cần đặt số lượng tác phẩm không quá sáu (3–6), nếu không sẽ khó đọc sơ đồ (quá bão hòa). Giới hạn trên (sáu) buộc người thiết kế phải sử dụng hệ thống phân cấp khi mô tả các đối tượng phức tạp và giới hạn dưới (ba) đảm bảo rằng sơ đồ tương ứng có đủ chi tiết để chứng minh cho việc tạo ra nó.

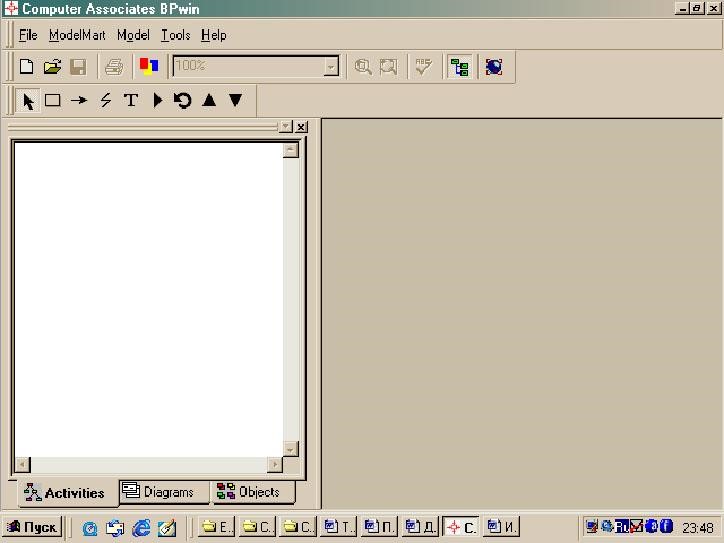
Trong sơ đồ phân tích, công việc quan trọng nhất và hoàn thành đầu tiên nằm ở phía trên bên trái. Công việc ít quan trọng hơn hoặc hoàn thành muộn hơn sẽ lần lượt được thực hiện.

 ***Loại sơ đồ III - Sơ đồ cây nút*** hiển thị sự phụ thuộc theo thứ bậc của các công việc, nhưng không thể hiện mối quan hệ giữa các công việc (có thể có bất kỳ số lượng sơ đồ nào trong số này, vì cây có thể được xây dựng ở bất kỳ độ sâu nào và không nhất thiết phải từ gốc).

***Quy trình công nghệ mô hình hóa IDEF0:***

BPWin CASE có giao diện người dùng đơn giản và trực quan để xây dựng các mô hình và kịch bản chức năng cần thiết.

Nó phụ thuộc vào công nghệ được sử dụng. Hình 2.2 thể hiện cửa sổ BPWin ( **Computer Associates BPWin** ).



**Cơm. 2.2** Cửa sổ BPWin

Thanh công cụ chính của cửa sổ ***Computer Associates BPwin*** chứa các nút sau:

 - tạo ra một mô hình mới,

 – mở một mô hình hiện có,

 – lưu mô hình đã xây dựng,

 – in mẫu,

- lựa chọn quy mô,

 – chia tỷ lệ,

 - kiểm tra chính tả,

 – bật/tắt trình điều hướng mô hình,

 – bật/tắt Model Mart.

Trình điều hướng mô hình hiển thị thành phần của mô hình theo cấp độ phát triển. Với sự trợ giúp của nó, bạn có thể dễ dàng và nhanh chóng chuyển từ cấp này sang cấp khác. Làm việc với trình điều hướng mô hình cũng tương tự như làm việc với Windows Explorer.

Thanh công cụ đặc biệt chứa các nút chính sau:

– chỉnh sửa các khối chức năng và mũi tên,

– thêm một khối chức năng vào mô hình,

- tạo mũi tên,

– chuyển sang cấp độ cao hơn của mô hình,

– phân rã mô hình.

Cửa sổ mô hình là nơi tạo ra mô hình chức năng của hệ thống đang nghiên cứu. Trong đó, các khối chức năng được xây dựng và chỉnh sửa, các mũi tên được vẽ và chỉnh sửa, quá trình phân rã được thực hiện.

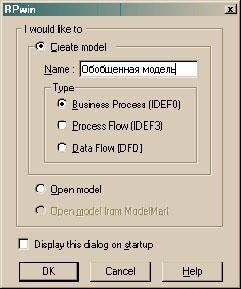
### Chuẩn bị mô hình

1. Bấm vào nút tạo mô hình để mở hộp thoại **BPWin** (Hình 2.3):

hộp thoại ***BPWin*** thực hiện các hành động sau:

###  chọn Quy trình kinh doanh (IDEF0) ;

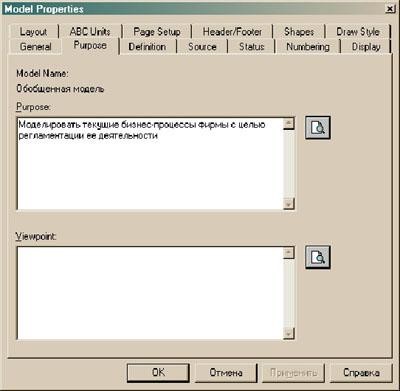
* đặt tên model và nhấn **OK** ;
* trong cửa sổ ***Thuộc tính cho Mô hình Mới*** , ghi họ của tác giả mô hình;
* bấm vào nút **OK** .



**Cơm. 2.3** Hộp thoại **BPWin**

2. Sử dụng lệnh Model/Model Properties để mở hộp thoại

cửa sổ Thuộc tính mô hình (Hình 2.4), trong đó bạn có thể định cấu hình các thuộc tính mô hình theo các khuyến nghị về phương pháp luận.



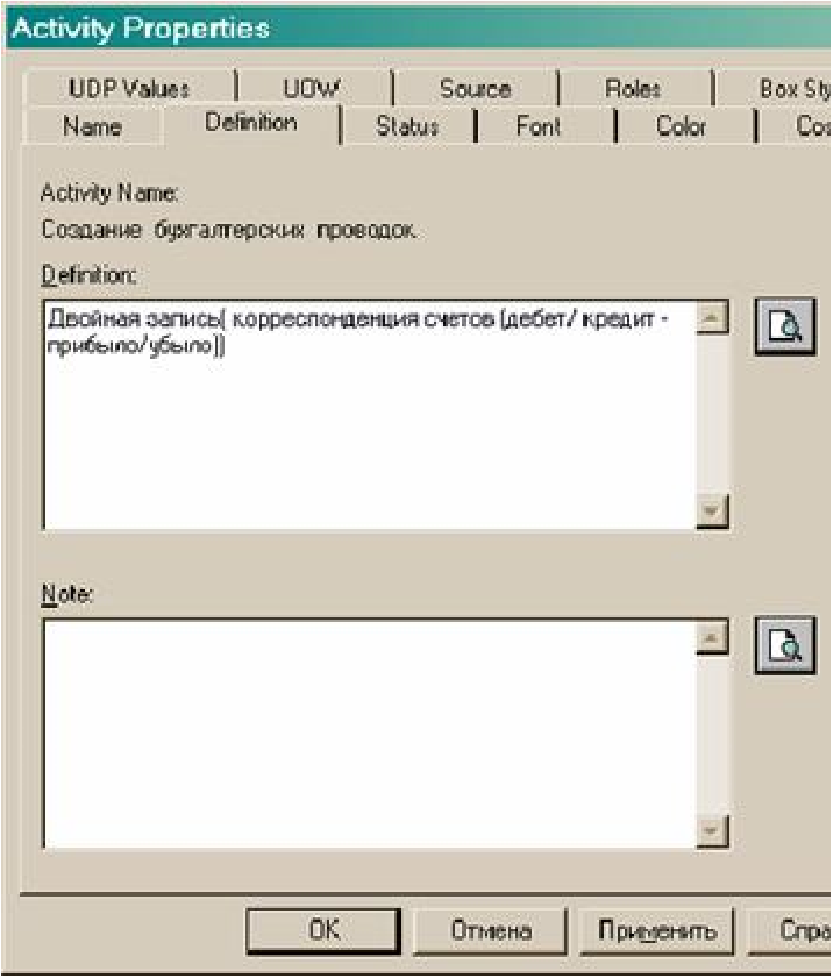
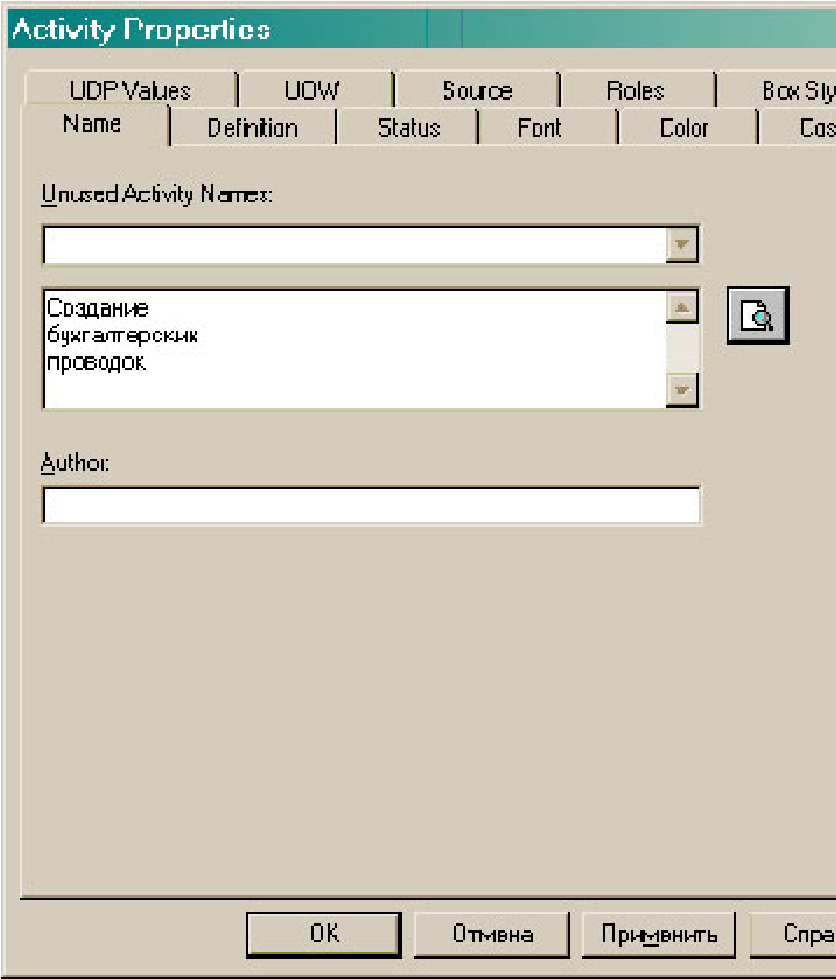
**Cơm. 2.4** Hộp thoại Thuộc tính Mô hình

### Thực hiện công việc phòng thí nghiệm Cấp độ đầu tiên

1. Thiết kế khối chức năng trong cửa sổ mô hình bằng cách thực hiện các bước sau:

ồ trong menu ngữ cảnh của khối chức năng chọn

lệnh **...** ; ồ trong hộp thoại Thuộc ***tính hoạt động*** (Hình 2.5), trong tab **Tên ,** đặt **tên của** hoạt động (viết tắt) sẽ được đặt trong khối chức năng này và trong tab **Định nghĩa ,** trong trường **Định nghĩa ,** nhập **mô tả chi tiết đầy đủ của** hoạt động; ồ trong tab **Phông chữ** đặt phông chữ ***Arial Cyr*** và đánh dấu vào các ô để cho phép sử dụng phông chữ này trong tất cả các khối chức năng của sơ đồ ( ***Tất cả các hoạt động TRONG cái này sơ đồ , Tất cả các hoạt động TRONG cái này người mẫu*** và ***Thay đổi tất cả lần xuất hiện của cái này phông chữ TRONG cái mô hình*** ), sau đó nhấp vào **OK** .



**Cơm . 2,5** Đối thoại cửa sổ ***Thuộc tính hoạt động***

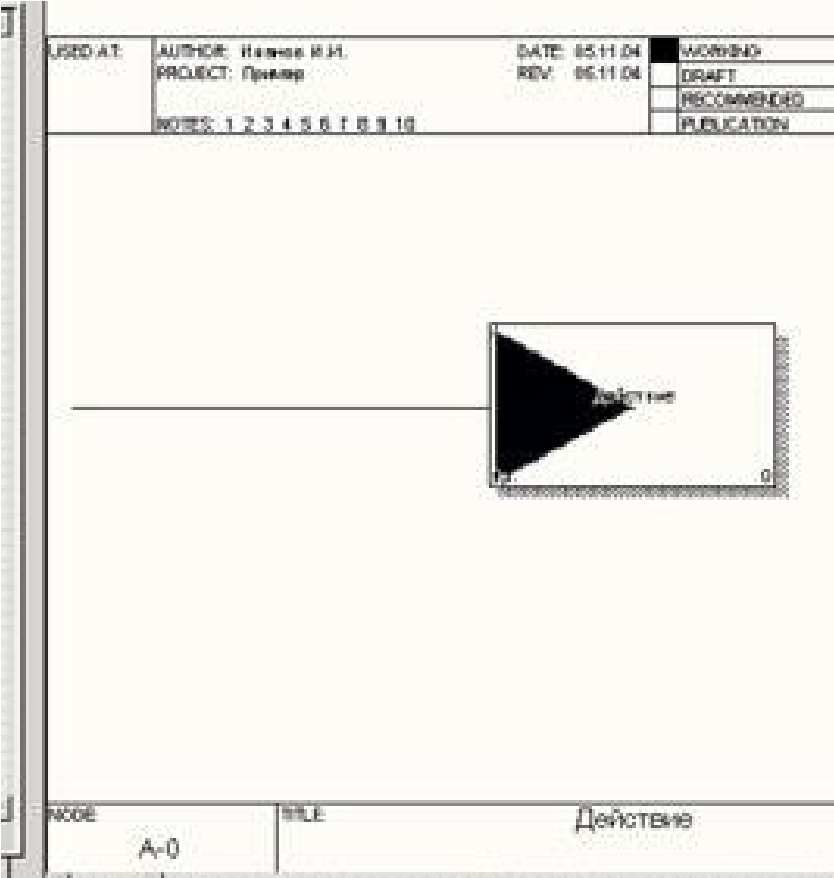
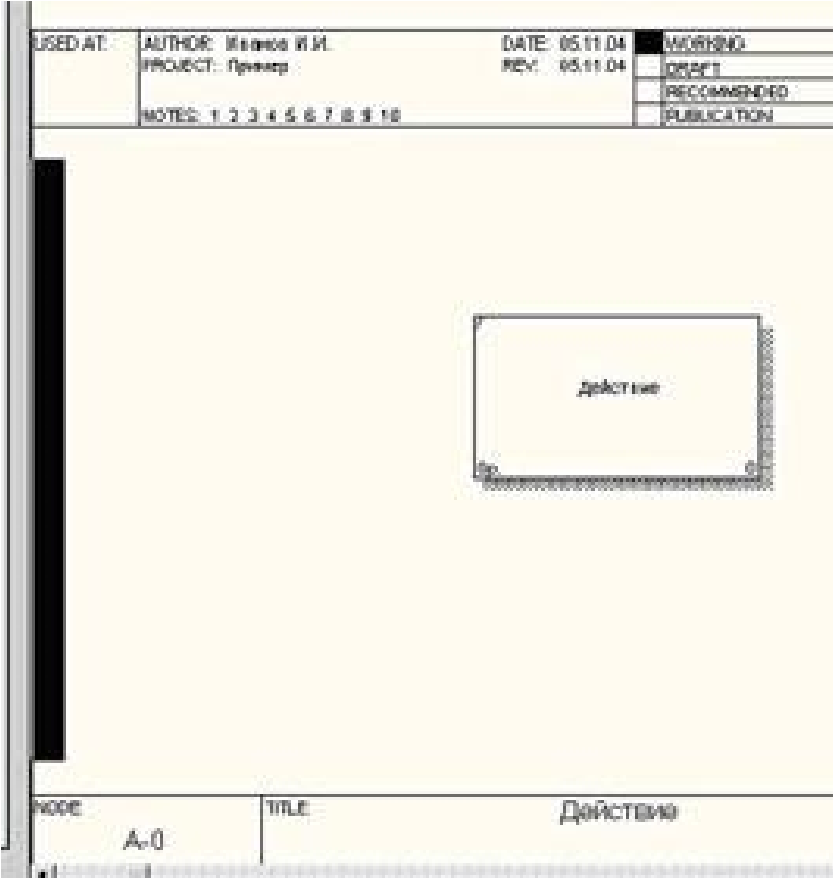
2. Tạo mũi tên Đăng nhập bằng cách làm theo các bước sau:

ồ nhấn nút mũi tên tạo ( Ưu tiên Công cụ mũi tên

– );

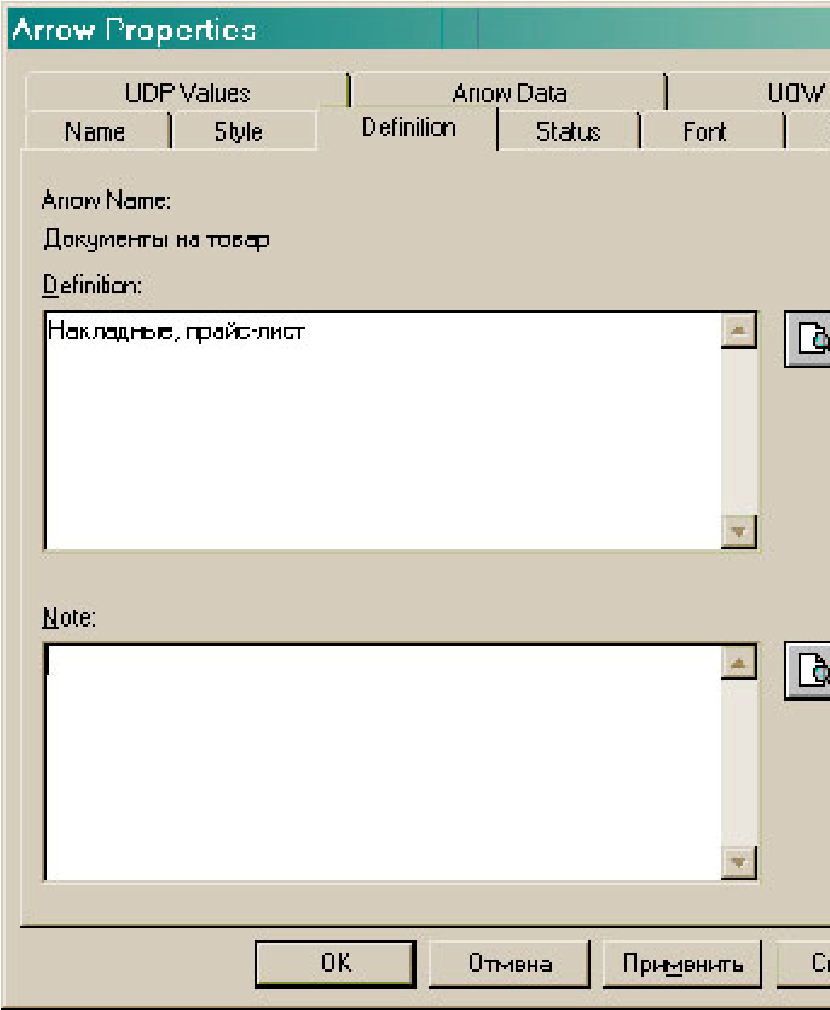
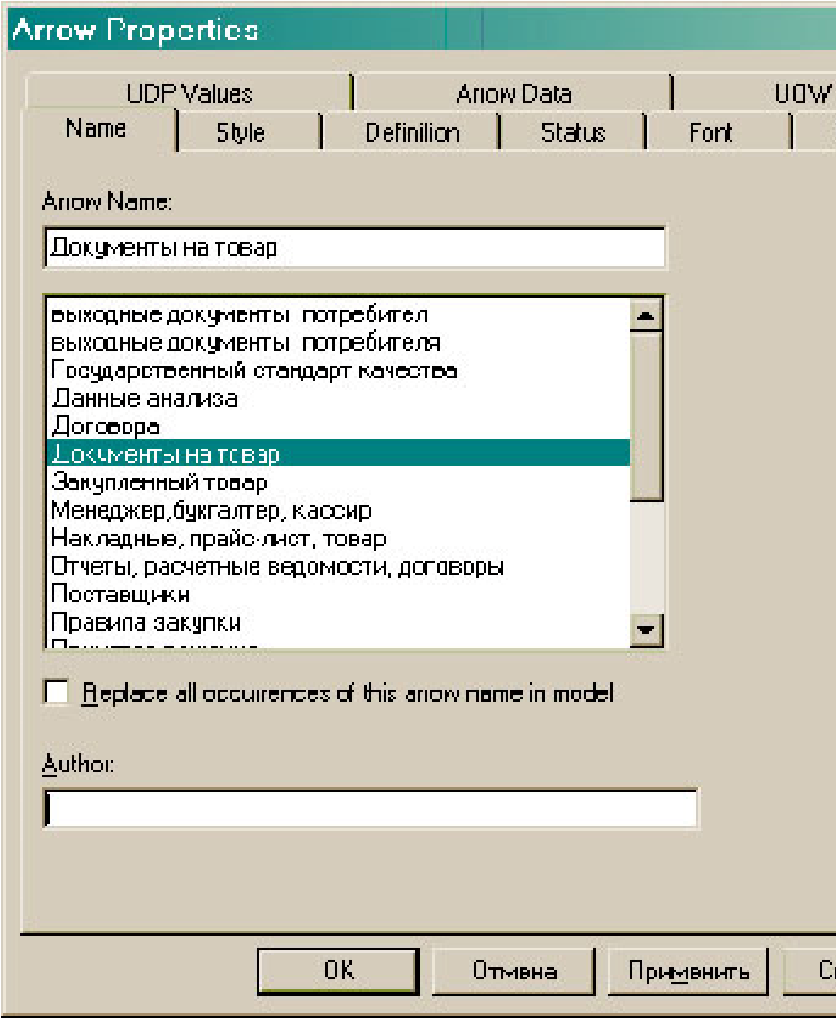
* nhấp đúp vào đường viền bên trái của cửa sổ mô hình, sau đó

bấm vào viền trái của khối chức năng (Hình 2.6); ồ nhấn nút chỉnh sửa mũi tên ( Công cụ con trỏ – ); ồ trong menu ngữ cảnh mũi tên, chọn lệnh Tên...;



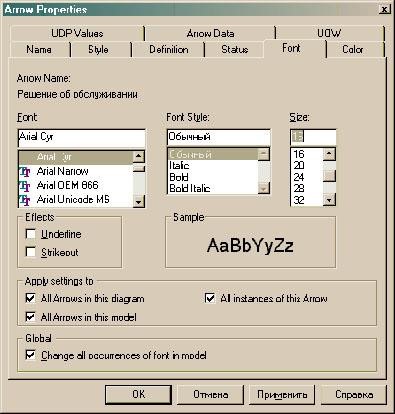
**Cơm. 2.6** Thiết kế mũi tên Đăng nhập

* hộp thoại Thuộc tính Mũi tên (Hình 2.7), trong tab Tên, đặt tên của mũi tên (ngắn) và trong tab Định nghĩa, trong trường Định nghĩa, nhập mô tả chi tiết đầy đủ về mục đích của mũi tên này;



**Cơm. 2.7** Hộp thoại Thuộc tính Mũi tên

* trong menu ngữ cảnh mũi tên, chọn lệnh Phông chữ ...; ồ trong hộp thoại Arrow Properties (Hình 2.8), trong tab Font , đặt phông chữ Arial Cyr và chọn các hộp để sử dụng phông chữ này cho tất cả các mũi tên trong sơ đồ (All Arrows TRONG cái này sơ đồ , Tất cả các mũi tên ***TRONG cái này model , Tất cả các trường hợp này Mũi tên*** và ***Thay đổi tất cả lần xuất hiện của cái này phông chữ TRONG cái người mẫu*** *);*



**Cơm. 2.8** Hộp thoại Thuộc tính Mũi tên

1. Tạo mũi tên ***Thoát*** bằng cách lặp lại bước 2, thay thế đường viền *bên trái bằng đường viền bên phải* ;
2. Thiết kế mũi tên ***Điều khiển*** , lặp lại bước 2, thay thế các đường viền *bên trái bằng các đường viền trên cùng* ;
3. Thiết kế Cơ chế mũi tên ***,*** lặp lại bước 2, thay thế các đường viền *bên trái bằng các đường viền bên dưới* .

### Cấp độ thứ hai

1. Đi đến cấp độ lập mô hình thấp hơn bằng cách sử dụng nút phân tách mô hình .

Trong hộp thoại ***Số lượng hộp hoạt động*** biểu thị:

loại mô hình – ***IDEF0*** ; ồ số khối chức năng cấp thấp hơn (từ ba đến sáu, nếu không sơ đồ sẽ trở nên quá tải và khó đọc).

1. Thiết kế mũi tên kế thừa từ cấp độ đầu tiên bằng cách thực hiện các bước sau: o nhấp vào nút mũi tên tạo;

* bấm vào ***đầu mũi tên Đầu vào*** ( ***Control* , *Mechanism*** );
* Click vào đường viền tương ứng của khối chức năng cần thiết.

3. Thiết kế mũi tên bên trong theo các bước sau: o nhấp vào nút mũi tên tạo;

* bấm vào viền bên phải của khối chức năng; ồ Bấm vào đường viền tương ứng của khối chức năng liên quan.

4. Tạo mũi tên phân nhánh bằng cách làm theo các bước sau:

* nhấn nút mũi tên chỉnh sửa; ồ bấm vào đoạn mũi tên; ồ Bấm vào ranh giới khối chức năng cần thiết.

5. Tạo một mũi tên hợp nhất bằng cách làm theo các bước sau: o nhấn nút mũi tên chỉnh sửa; ồ bấm vào đường viền của khối chức năng; ồ bấm vào đoạn mũi tên; ồ lặp lại trang 2–5 cho tất cả các khối chức năng cấp độ.

# Phòng thí nghiệm số 3

# Tạo sơ đồ ca sử dụng

**Mục đích công việc:** rèn luyện kỹ năng xây dựng sơ đồ tình huống.

**Bài tập:**

1. tạo sơ đồ ca sử dụng chính, xác định các ca sử dụng và các tác nhân trên đó;
2. thêm mối quan hệ giữa các tác nhân và trường hợp sử dụng;
3. tạo sơ đồ ca sử dụng bổ sung;
4. thêm mô tả cho các tác nhân và trường hợp sử dụng;
5. Đối với mỗi trường hợp sử dụng, hãy xác định luồng sự kiện dưới dạng một tệp riêng biệt và đính kèm nó vào trường hợp sử dụng.

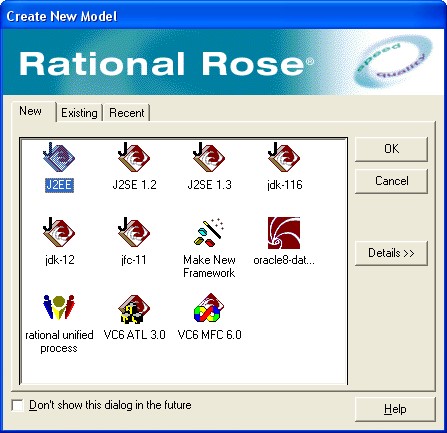
**Nội dung của báo cáo:**

1. tạo sơ đồ ca sử dụng;
2. mô tả ngắn gọn về từng tác nhân và trường hợp sử dụng;
3. Mô tả luồng sự kiện cho từng trường hợp sử dụng.

### Ví dụ về công việc đã thực hiện 1. Hành động sơ bộ - tạo mô hình mới

*Lưu ý:* Trong ví dụ này và các ví dụ tiếp theo, chúng ta sẽ thiết kế hệ thống cho chủ đề “ *Doanh nghiệp lắp ráp và bán máy tính* ”.

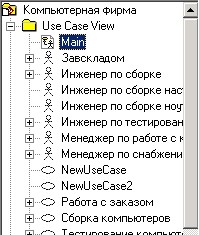
Làm việc với Rational Rose bắt đầu bằng việc tạo một mô hình. Khi bạn khởi động chương trình, một hộp thoại tạo mô hình sẽ tự động xuất hiện (Hình 3.1). Để gọi nó, bạn cũng có thể chọn mục menu Tệp -> Mới. Trong tab *Mới* , chọn khung được yêu cầu (trong trường hợp của chúng tôi là *J2EE* ). Trong tất cả các phòng thí nghiệm, chúng tôi sẽ tạo sơ đồ cho mô hình đã tạo.



**Cơm. 3.1.** Cửa sổ tạo mô hình mới

### 2. Tạo sơ đồ ca sử dụng chính

Theo mặc định, sơ đồ ca sử dụng chính ( *Main* ) đã tồn tại trong khung nhìn Use Case của trình duyệt (Hình 3.2). Để điền vào, bạn cần mở nó bằng cách nhấp đúp vào nó và đặt các tác nhân và trường hợp sử dụng vào đó.



**Cơm. 3.2.** Trình bày các trường hợp sử dụng

Đối với lĩnh vực chủ đề của chúng tôi, chúng tôi đã xác định được các tác nhân sau:

|  |  |
| --- | --- |
| **Diễn viên** | **Mô tả ngắn gọn** |
| Người quản lý tài khoản | Một nhân viên giao tiếp với khách hàng và làm việc theo đơn đặt hàng |
| Giám đốc cung ứng | Nhân viên mua các linh kiện cần thiết |
| Kỹ sư lắp ráp máy tính để bàn | Nhân viên lắp ráp máy tính để bàn |
| Kỹ sư lắp ráp laptop | Nhân viên lắp ráp laptop |
| Kỹ sư kiểm tra | Một nhân viên kiểm tra các máy tính đã lắp ráp |
| Trưởng kho | Nhân viên quản lý kho linh kiện |

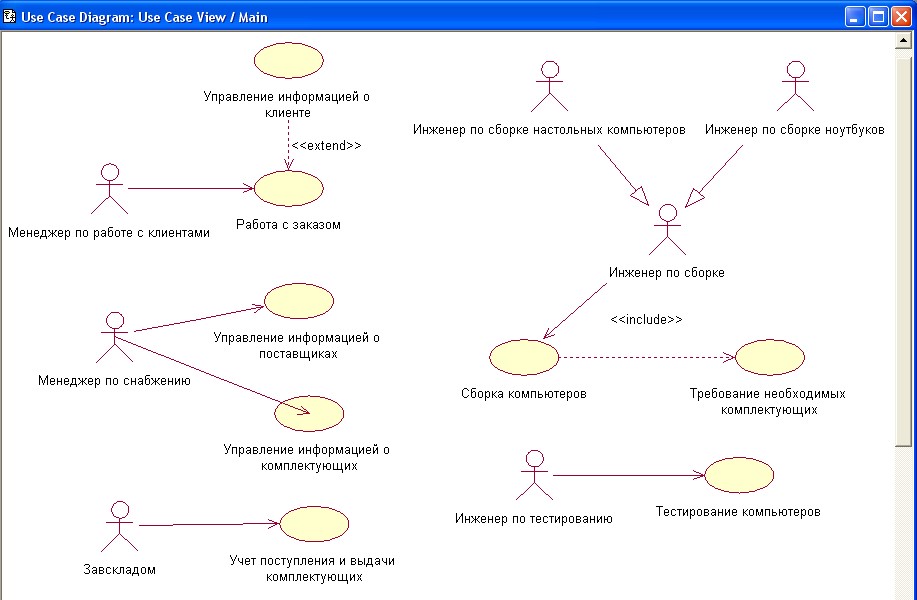
Bây giờ chúng ta hãy xem xét những khả năng mà hệ thống của chúng ta sẽ cung cấp:

* Tác nhân *Người quản lý dịch vụ khách hàng* sử dụng hệ thống để đặt, chỉnh sửa đơn hàng và quản lý thông tin về khách hàng của công ty;
* Tác nhân *Người quản lý cung ứng* sử dụng hệ thống để xem danh sách các thành phần cần thiết cho việc mua hàng và duy trì thông tin về nguồn cung cấp;
* diễn viên *Một kỹ sư lắp ráp máy tính* để bàn sử dụng hệ thống để xem các yêu cầu công việc lắp ráp máy tính cá nhân, đặt hàng các bộ phận từ kho và đánh dấu tiến độ công việc;
* diễn viên *Kỹ sư lắp ráp máy tính xách tay* sử dụng hệ thống để xem các lệnh công việc lắp ráp máy tính xách tay, đặt hàng linh kiện từ kho và đánh dấu tiến độ công việc;
* Tác nhân *Kỹ sư kiểm tra* sử dụng hệ thống để xem các yêu cầu công việc nhằm kiểm tra các sản phẩm đã lắp ráp và đánh dấu tiến độ công việc;
* diễn viên *Zavskladom* sử dụng một hệ thống để ghi lại việc nhận và xuất các linh kiện.

Căn cứ vào những điều trên, có thể xác định được các tiền lệ sau:

|  |  |
| --- | --- |
| **tiền lệ** | **Mô tả ngắn gọn** |
| Làm việc với một đơn đặt hàng | Được khởi chạy bởi người quản lý tài khoản. Cho phép bạn nhập, thay đổi, xóa hoặc xem đơn hàng. |
| Quản lý thông tin khách hàng | Được khởi chạy bởi người quản lý tài khoản. Cho phép bạn thêm, thay đổi hoặc xóa khách hàng cũng như xem thông tin về khách hàng. |
| Quản lý thông tin nhà cung cấp | Ra mắt bởi người quản lý mua hàng. Cho phép bạn thêm, thay đổi hoặc xóa nhà cung cấp cũng như xem thông tin nhà cung cấp. |
| Quản lý thông tin bộ phận | Ra mắt bởi người quản lý mua hàng. Cho phép bạn xem thông tin về các thành phần, phân tích mức tiêu thụ của chúng, dự đoán số lượng cần thiết và đặt hàng. |
| Lắp ráp máy tính | Được điều hành bởi một kỹ sư lắp ráp. Cho phép bạn xem các đơn hàng lắp ráp máy tính và ghi chú tiến độ công việc. |
| Yêu cầu đối với các thành phần cần thiết | Được điều hành bởi một kỹ sư lắp ráp. Được thiết kế để yêu cầu các thành phần cần thiết từ kho. |
| Kiểm tra máy tính | Được điều hành bởi một kỹ sư kiểm tra. Cho phép bạn xem danh sách các máy tính cần kiểm tra và ghi chú tiến độ công việc. |
| Kế toán nhập, xuất linh kiện | Ra mắt bởi người quản lý kho. Cho phép bạn lưu giữ hồ sơ về việc nhận và cấp phụ tùng và linh kiện. |

Sơ đồ ca sử dụng chính đã tạo được hiển thị trong Hình. 3.3:



**Cơm. 3.3.** Sơ đồ ca sử dụng chính

Bây giờ chúng ta hãy xem xét mối quan hệ giữa các chủ thể và tiền lệ. Trong UML, chỉ có một loại mối quan hệ giữa tác nhân và ca sử dụng - mối quan hệ giao tiếp. Do đó, chúng tôi đã kết nối tất cả các tác nhân với các tiền lệ bằng cách sử dụng mối quan hệ *Hiệp hội một chiều* . Vì chúng ta không thể chỉ định một loại mối quan hệ khác ở đây nên khuôn mẫu *giao tiếp* không cần phải được chỉ định (nó được ngụ ý ngầm).

Đối với tiền lệ *Lắp ráp máy tính* , không quan trọng diễn viên nào sẽ tương tác với nó - *Kỹ sư lắp ráp máy tính để bàn* hay *Kỹ sư lắp ráp máy tính xách tay.* Do đó, chúng tôi đã giới thiệu một diễn viên khác - *Kỹ sư lắp ráp* , người mà chúng tôi đã kết nối hai diễn viên đầu tiên bằng một *mối quan hệ khái quát* .

Mối quan hệ giữa các ca sử dụng *Làm việc với đơn hàng* và *Quản lý thông tin khách hàng* là mối quan hệ mở rộng, vì khi tác nhân *Quản lý dịch vụ khách hàng* làm việc với đơn hàng (điền, thay đổi, v.v.), anh ta không phải lúc nào cũng quản lý thông tin khách hàng.

Mối quan hệ giữa tiền lệ *Lắp ráp máy tính* và *Yêu cầu các thành phần cần thiết* là mối quan hệ bao hàm, vì để lắp ráp máy tính, cần phải đặt hàng các thành phần cần thiết từ kho.

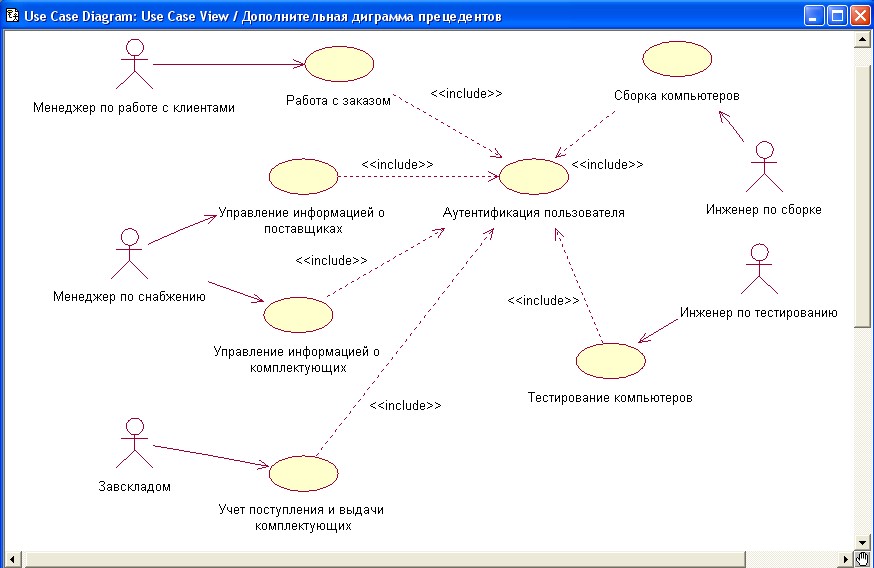
### 3. Luồng sự kiện cho các ca sử dụng trong sơ đồ ca sử dụng chính

Chúng tôi sẽ mô tả luồng sự kiện cho các tiền lệ bằng cách sử dụng mẫu sau:

* Điều kiện tiên quyết X.1;
* Dòng chính X.2;
* Luồng phụ X.3;
* Các luồng thay thế X.4;  Hậu điều kiện X.5. trong đó X là số từ một đến số tiền lệ.

### 4. Tạo sơ đồ ca sử dụng bổ sung

Như có thể thấy từ phần mô tả luồng sự kiện cho tất cả các trường hợp sử dụng, mỗi trường hợp đều bao gồm xác minh người dùng. Việc kiểm tra được thực hiện thống nhất cho mọi tiền lệ. Do đó, nó có thể được biểu diễn dưới dạng *Xác thực người dùng trong trường hợp sử dụng riêng biệt* , được kết nối bằng mối quan hệ bao hàm với tất cả những trường hợp khác. Kết quả của việc tạo sơ đồ được hiển thị trong Hình. 3.4:



**Cơm. 3,4** . Sơ đồ ca sử dụng bổ sung

### Ví dụ thực thi 1. Luồng sự kiện cho ca sử dụng “Làm việc với đơn hàng”

#### 1.1 Điều kiện tiên quyết

Nếu đơn hàng được đặt cho một khách hàng mới thì dòng phụ Thêm khách hàng mới của *tiền* lệ *Quản lý thông tin khách hàng* phải được hoàn thành trước khi bắt đầu.

#### 1.2 Chủ đề chính

Ca sử dụng bắt đầu được thực thi khi người quản lý kết nối với hệ thống và nhập tên và mật khẩu của mình. Hệ thống kiểm tra tính chính xác của mật khẩu (E-1) và hiển thị các tùy chọn hành động có thể thực hiện: *thêm ( Thêm )* , *thay đổi (Thay đổi),* *xóa ( Delete )* , *xem (View)* hoặc *thoát ( Exit ).*

|  |  |
| --- | --- |
| Nếu *thao tác Thêm được* chọn , *quy trình Thêm đơn hàng mới sẽ được* thực thi *.* | S-1: |
| Nếu thao tác *Thay đổi được chọn* , *luồng Thứ tự Thay đổi sẽ được thực thi* . | S-2: |
| Nếu thao tác xóa được chọn *thì luồng lệnh* xóa sẽ được thực thi *( Xóa Đặt hàng ).* | S-3: |
| Nếu thao tác được chọn *xem* *(Xem* ) | S-4: |

*Chuỗi Xem thứ tự* đang chạy *.*

Nếu thao tác Thoát được chọn *, ca sử dụng* sẽ kết thúc.

#### 1.3 Dòng phụ

S-1: *thêm đơn hàng mới ( Add a New Order )*

Hệ thống hiển thị hộp thoại chứa trường trong đó người quản lý phải chọn loại máy tính (máy tính để bàn hoặc máy tính xách tay). Người dùng chọn loại cần thiết. Hệ thống hiển thị trường để chọn máy khách và danh sách các thành phần có thể có cho loại máy tính đã chọn, trong đó người quản lý đánh dấu các thành phần được máy khách chọn. Người quản lý điền vào các trường (E-2). Hệ thống ghi nhớ dữ liệu đã nhập và in hóa đơn để thanh toán. Sau đó, tiền lệ bắt đầu lại.

S-2: *đổi thứ tự (Change Order )*

Hệ thống hiển thị hộp thoại chứa danh sách đơn hàng và ô nhập mã số đơn hàng. Người quản lý chọn đơn hàng cần thiết từ danh sách hoặc nhập số đơn hàng vào trường (E-3). Hệ thống hiển thị thông tin về đơn hàng này. Người quản lý thực hiện những thay đổi cần thiết (E-2). Hệ thống ghi nhớ dữ liệu đã nhập. Sau đó, tiền lệ bắt đầu lại.

S-3: *xóa lệnh ( Xóa Đặt hàng )*

Hệ thống hiển thị hộp thoại chứa danh sách đơn hàng và ô nhập mã số đơn hàng. Người quản lý chọn đơn hàng cần thiết từ danh sách hoặc nhập số đơn hàng vào trường (E-3). Hệ thống xóa lệnh đã chọn (E-4). Sau đó, tiền lệ bắt đầu lại.

*4* : *Xem đơn hàng*

Hệ thống hiển thị hộp thoại chứa danh sách đơn hàng và ô nhập mã số đơn hàng. Người quản lý chọn đơn hàng cần thiết từ danh sách hoặc nhập số đơn hàng vào trường (E-3). Hệ thống hiển thị thông tin về đơn hàng đã chọn. Khi người quản lý xem lại thông tin, ca sử dụng sẽ bắt đầu lại.

#### 1.4 Các luồng thay thế

E-1: Tên người dùng hoặc mật khẩu đã nhập không hợp lệ. Người dùng phải nhập lại dữ liệu đầu vào hoặc hoàn thành ca sử dụng.

E-2: không phải tất cả các linh kiện cần thiết để lắp ráp máy tính đều đã được chọn hoặc không có sẵn. Người quản lý phải thay đổi thành phần của máy tính hoặc hoàn thành ca sử dụng.

E-3: Nhập sai số thứ tự. Người quản lý phải lặp lại đầu vào hoặc hoàn thành ca sử dụng.

E-4: hệ thống không thể xóa đơn hàng. Thông tin được lưu lại, hệ thống sẽ xóa đơn hàng sau. Việc thực hiện tiền lệ vẫn tiếp tục.

### 2. Luồng sự kiện của ca sử dụng “Quản lý thông tin khách hàng”

#### 2.1 Điều kiện tiên quyết 2.2 Luồng chính

Ca sử dụng bắt đầu được thực thi khi người quản lý kết nối với hệ thống và nhập tên và mật khẩu của mình. Hệ thống kiểm tra tính chính xác của mật khẩu (E-1) và hiển thị các tùy chọn hành động có thể thực hiện: *thêm ( Thêm )* , *thay đổi (Thay đổi),* *xóa ( Delete )* , *xem (View)* hoặc *thoát ( Exit ).*

Nếu *thao tác Thêm được chọn* , S-1: *Luồng Thêm khách hàng mới được thực thi* .

|  |  |
| --- | --- |
| Nếu thao tác *Thay đổi được chọn* , luồng *Thay đổi dữ liệu khách hàng sẽ được thực thi* . | S-2: |
| Nếu *thao tác Xóa được chọn* , chuỗi *Xóa Khách hàng sẽ được thực thi .* | S-3: |
| Nếu thao tác được chọn *xem* *(Xem* ) | S-4: |

*Chuỗi Xem dữ liệu khách hàng* được thực thi .\ Nếu thao tác Thoát được chọn *, ca sử dụng* sẽ kết thúc.

#### 2.3 Dòng phụ

S-1: *Thêm khách hàng mới*

Hệ thống hiển thị hộp thoại chứa các trường nhập dữ liệu về máy khách mới. Người dùng điền vào các trường (E-2). Hệ thống ghi nhớ dữ liệu đã nhập. Sau đó, tiền lệ bắt đầu lại.

S-2: *Thay đổi dữ liệu khách hàng*

Hệ thống hiển thị hộp thoại chứa danh sách khách hàng và trường nhập số khách hàng. Người quản lý chọn khách hàng được yêu cầu từ danh sách hoặc nhập số của khách hàng đó vào trường (E-3). Hệ thống hiển thị thông tin về khách hàng này. Người quản lý thực hiện những thay đổi cần thiết (E-2). Hệ thống ghi nhớ dữ liệu đã nhập. Sau đó, tiền lệ bắt đầu lại.

S-3 *: Xóa khách hàng*

Hệ thống hiển thị hộp thoại chứa danh sách khách hàng và trường nhập số khách hàng. Người quản lý chọn khách hàng được yêu cầu từ danh sách hoặc nhập số của khách hàng đó vào trường (E-2). Hệ thống xóa máy khách đã chọn (E-4). Sau đó, tiền lệ bắt đầu lại.

S-4: *Xem dữ liệu khách hàng*

Hệ thống hiển thị hộp thoại chứa danh sách khách hàng và trường nhập số khách hàng. Người quản lý chọn khách hàng được yêu cầu từ danh sách hoặc nhập số của khách hàng đó vào trường (E-3). Hệ thống hiển thị thông tin về khách hàng đã chọn. Khi người quản lý xem lại thông tin, ca sử dụng sẽ bắt đầu lại.

#### 2.4 Các luồng thay thế

E-1: Tên người dùng hoặc mật khẩu đã nhập không hợp lệ. Người dùng phải nhập lại dữ liệu đầu vào hoặc hoàn thành ca sử dụng.

E-2: không phải tất cả các trường đều được điền. Người quản lý phải điền vào chỗ trống hoặc hoàn thành ca sử dụng.

E-3: Nhập sai mã số khách hàng. Người quản lý phải lặp lại đầu vào hoặc hoàn thành ca sử dụng.

E-4: Hệ thống không thể xóa máy khách. Thông tin được lưu lại, hệ thống sẽ xóa client sau đó. Việc thực hiện tiền lệ vẫn tiếp tục.

### 3. Luồng sự kiện của tiền lệ “Kế toán nhập, xuất linh kiện

#### 3.1 Điều kiện tiên quyết 3.2 Luồng chính

Tiền lệ bắt đầu được thực thi khi người quản lý kho kết nối với hệ thống và nhập tên và mật khẩu của mình. Hệ thống kiểm tra tính chính xác của mật khẩu (E-1) và hiển thị các tùy chọn hành động có thể thực hiện: *thêm ( Add )* , *đánh dấu (Mark)* hoặc *thoát ( Exit ).*

Nếu *thao tác Thêm được chọn* , S-1: *quy trình Thêm thành phần mới được thực thi* .

Nếu *thao tác Đánh dấu* được chọn , S-2:

Luồng được thực hiện *để đánh dấu việc phát hành các thành phần (Mark Components )* .

Nếu thao tác Thoát được chọn *, ca sử dụng* sẽ kết thúc.

#### 3.3 Dòng phụ

S-1: *thêm các thành phần đã nhận ( Add a New Components )*

Hệ thống hiển thị hộp thoại chứa các trường để nhập tên linh kiện, số lượng và nhà cung cấp. Người quản lý kho điền vào các trường được chỉ định (E-2). Hệ thống ghi nhớ dữ liệu đã nhập. Sau đó, tiền lệ bắt đầu lại.

S-2: *Ghi chú về việc cấp linh kiện (Change Order )*

Hệ thống hiển thị danh sách linh kiện còn trong kho. Người quản lý kho nhập số lượng xuất ra (E-3) đối diện với thành phần yêu cầu. Hệ thống ghi nhớ dữ liệu đã nhập. Sau đó, tiền lệ bắt đầu lại.

#### 3.4 Các luồng thay thế

E-1: Tên người dùng hoặc mật khẩu đã nhập không hợp lệ. Người dùng phải nhập lại dữ liệu đầu vào hoặc hoàn thành ca sử dụng.

E-2: không phải tất cả các trường đều được điền. Người dùng phải điền vào các trường còn thiếu hoặc hoàn thành ca sử dụng.

E-3: số lượng linh kiện xuất ra vượt quá số lượng trong kho. Người dùng phải nhập lại dữ liệu đầu vào hoặc hoàn thành ca sử dụng.

### 4. Luồng sự kiện cho ca sử dụng “Lắp ráp máy tính”

#### 4.1 Điều kiện tiên quyết 4.2 Luồng chính

Ca sử dụng bắt đầu chạy khi kỹ sư lắp ráp đăng nhập vào hệ thống và nhập tên và mật khẩu của mình. Hệ thống kiểm tra tính chính xác của mật khẩu (E-1) và hiển thị các tùy chọn hành động có thể thực hiện: *xem (Xem)* , *đánh dấu (Đánh dấu)* hoặc *thoát ( Thoát ).*

Nếu *thao tác Xem* được chọn , S-1:

chủ đề đang chạy *Xem lệnh công việc lắp ráp máy tính ( Xem Đặt hàng máy tính )* .

Nếu *thao tác Đánh dấu* được chọn , S-2:

Thread được thực thi *nhằm đánh dấu về trạng thái của máy tính được lắp ráp dọc theo dây chuyền (Mark Computer)* .

Nếu thao tác Thoát được chọn *, ca sử dụng* sẽ kết thúc.

#### 4.3 Dòng phụ

S-1 : *Xem đơn hàng lắp ráp máy tính (Xem đơn hàng Đặt hàng máy tính )*

Hệ thống hiển thị hộp thoại chứa danh sách đơn hàng và ô nhập mã số đơn hàng. Kỹ sư chọn thứ tự công việc được yêu cầu từ danh sách hoặc nhập số của nó vào trường (E-2). Hệ thống hiển thị thông tin về trang phục đã chọn. Khi kỹ sư xem xét thông tin, ca sử dụng sẽ bắt đầu lại.

S-2: *đánh dấu về tình trạng máy tính đang được lắp ráp (Mark Computer)*

Hệ thống hiển thị hộp thoại chứa danh sách đơn hàng. Gần lệnh công việc được yêu cầu, kỹ sư ghi chú về trạng thái của máy tính cho lệnh công việc này. Kỹ sư lưu các thay đổi. Sau đó, tiền lệ bắt đầu lại.

#### 4.4 Các luồng thay thế

E-1: Tên người dùng hoặc mật khẩu đã nhập không hợp lệ. Người dùng phải nhập lại dữ liệu đầu vào hoặc hoàn thành ca sử dụng.

E-2: không phải tất cả các trường đều được điền. Người dùng phải điền vào các trường còn thiếu hoặc hoàn thành ca sử dụng.

E-3: Nhập sai số thứ tự công việc. Kỹ sư phải lặp lại đầu vào hoặc hoàn thành ca sử dụng.

### 5. Luồng sự kiện cho ca sử dụng “Yêu cầu các thành phần cần thiết

#### 5.1 Điều kiện tiên quyết 5.2 Luồng chính

Ca sử dụng bắt đầu chạy khi kỹ sư lắp ráp đăng nhập vào hệ thống và nhập tên và mật khẩu của mình. Hệ thống kiểm tra tính chính xác của mật khẩu (E-1) và hiển thị các tùy chọn hành động có thể thực hiện: *xem (Xem)* , *yêu cầu ( Đặt hàng )* hoặc *thoát ( Thoát ).*

Nếu *thao tác Xem* được chọn , S-1:

luồng đang chạy *để xem các thành phần được yêu cầu trong kho (Xem Đã đặt hàng Linh kiện TRÊN Kho )* .

Nếu thao tác *Yêu cầu ( Đặt hàng ) được chọn* , S-2:

một luồng đang chạy *để yêu cầu các thành phần cần thiết từ kho ( Đặt hàng Yêu cầu Linh kiện TRÊN Kho )* .

Nếu thao tác Thoát được chọn *, ca sử dụng* sẽ kết thúc.

#### 5.3 Dòng phụ

S-1 : *Xem các linh kiện được yêu cầu trong kho (View Ordered Linh kiện TRÊN Kho )*

Hệ thống hiển thị các thông tin sau về tất cả các đơn đặt hàng do kỹ sư lắp ráp này đặt: ngày yêu cầu, tên các bộ phận, số lượng, đơn hàng đã hoàn thành hay chưa. Khi kỹ sư lắp ráp xem xét danh sách xong sẽ thông báo cho hệ thống. Tiền lệ lại bắt đầu.

S-2: *Yêu cầu các linh kiện cần thiết từ kho ( Order Yêu cầu Linh kiện TRÊN Kho )*

Hệ thống hiển thị hộp thoại chứa các trường để nhập danh sách các thành phần cần thiết và số lượng của chúng. Kỹ sư lắp ráp điền vào nó. Hệ thống ghi nhớ dữ liệu đã nhập. Sau đó, tiền lệ bắt đầu lại.

#### 5.4 Các luồng thay thế

E-1: Tên người dùng hoặc mật khẩu đã nhập không hợp lệ. Người dùng phải nhập lại dữ liệu đầu vào hoặc hoàn thành ca sử dụng.

Mô tả luồng sự kiện cho các trường hợp sử dụng *Quản lý thông tin nhà cung cấp* và *Quản lý thông tin bộ phận* là tương tự nhaumô tả use case *Quản lý thông tin khách hàng;* dành cho việc *Kiểm tra máy tính trước đây* - dành cho việc *Lắp ráp máy tính trước đây.*

# Phòng thí nghiệm số 4

# Tạo sơ đồ tương tác

**Mục đích công việc:** đạt được kỹ năng xây dựng sơ đồ trình tự và hợp tác.

**Bài tập:** tạo sơ đồ trình tự và hợp tác cho một trong các kịch bản của bất kỳ ca sử dụng nào được tạo trong bài thí nghiệm số 1.

2.

**Một ví dụ về công việc đang được thực hiện.**

Chúng tôi sẽ tạo sơ đồ tương tác cho kịch bản “Thêm đơn hàng mới” của tiền lệ “Làm việc với đơn hàng”. Trong trường hợp này, ngoài luồng chính, còn có các luồng thay thế. Mặc dù tiêu chuẩn ngôn ngữ UML cho phép phân nhánh theo sơ đồ tuần tự và hợp tác, nhưng để không làm lộn xộn sơ đồ của chúng tôi, chúng tôi sẽ hạn chế chỉ xem xét trường hợp người dùng nhập đúng mật khẩu của mình, điền chính xác vào các trường bắt buộc và dữ liệu đã nhập là được lưu trữ trong cơ sở dữ liệu mà không có lỗi. Nếu cần thiết, các luồng thay thế có thể được thể hiện bằng sơ đồ trình tự và hợp tác bổ sung.

Chúng tôi sẽ tạo sơ đồ tương tác trong Chế độ xem logic của trình duyệt. Để tách các sơ đồ này khỏi các sơ đồ khác (mà chúng tôi đã tạo hoặc sẽ tạo trong tương lai), trước tiên chúng tôi sẽ tạo một gói mới trong Chế độ xem logic - *Sơ đồ tương tác* , trong đó các sơ đồ được tạo bên dưới sẽ được đặt.

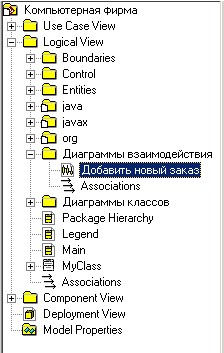
Việc xây dựng bất kỳ sơ đồ tương tác nào đều bắt đầu bằng việc xác định danh sách các đối tượng sẽ tham gia vào tương tác. Một sơ đồ lớp đã được phát triển cho kịch bản đã chọn trong Phòng thí nghiệm 3. Các thể hiện của các lớp trong sơ đồ này sẽ là những người tham gia vào sơ đồ tương tác.

*Lưu ý:* Nếu bạn có một trong hai loại sơ đồ tương tác, Rational Rose cho phép bạn tạo loại sơ đồ thứ hai. Để thực hiện việc này, bạn cần mở sơ đồ tương tác hiện có và chọn mục menu Duyệt > Tạo nên Sơ đồ trình tự ( cộng tác ) . Loại biểu đồ thứ hai sẽ được tạo tự động với cùng tên, trong cùng một gói và có cùng nội dung với loại biểu đồ đầu tiên. Hạn chế duy nhất của kỹ thuật này là các phần tử trong sơ đồ được tạo sẽ không được căn chỉnh tự động. Vì vậy, nếu sơ đồ ban đầu đủ lớn thì sẽ khó hiểu sơ đồ được tạo ra, bởi vì các phần tử có thể chồng lên nhau.

Trong công việc này, chúng tôi sẽ xây dựng cả hai loại sơ đồ tương tác từ đầu.

### Tạo sơ đồ trình tự cho kịch bản "Thêm đơn hàng mới" của trường hợp sử dụng "Làm việc với đơn hàng"

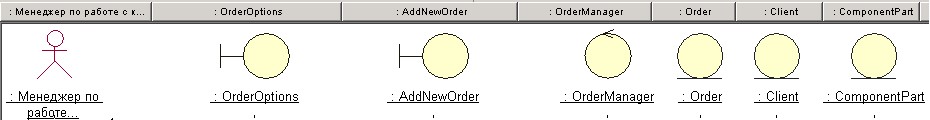
Để tạo sơ đồ trình tự, nhấp chuột phải vào gói Sơ đồ tương tác và chọn Mới > từ menu xuất hiện. Sự liên tiếp Sơ đồ , nhập tên của nó, sau đó nhấp đúp vào nó trong trình duyệt để mở (Hình 4.1).



**Cơm. 4.1.** Tạo sơ đồ trình tự

Việc xây dựng sơ đồ trình tự bắt đầu bằng việc đặt lên đó các đối tượng sẽ trao đổi thông điệp. Trước tiên, bạn cần đặt các đối tượng gửi tin nhắn và sau đó là các đối tượng nhận chúng. Sự tương tác được thực hiện bởi *người quản lý dịch vụ khách hàng.* Do đó, trong sơ đồ nó sẽ ở góc bên trái. Tiếp theo chúng ta đặt (Hình 4.2):

* đối tượng của lớp *OrderOptions* ( *Các tham số để làm việc với đơn hàng* ) *,* chịu trách nhiệm lựa chọn hành động có thể thực hiện được với đơn hàng trong trường hợp đang xem xét;
* một đối tượng của lớp *AddNewOrder* ( *Thêm một đơn hàng mới* ), chịu trách nhiệm thêm một đơn hàng;
* một đối tượng của lớp *OrderManager* ( *Order Manager* ), chịu trách nhiệm xử lý luồng sự kiện của trường hợp sử dụng đang xem xét;
* đối tượng của lớp *Order* ( *Order* );
* đối tượng của lớp *Client* ( *Client* );
* đối tượng của lớp *ComponentPart* ( *Component item* ).



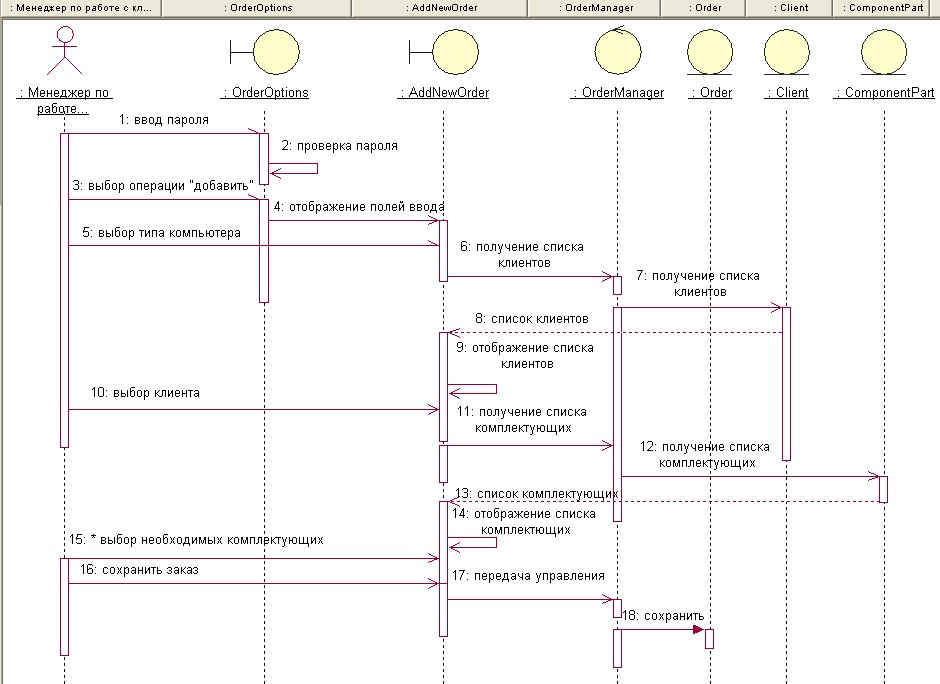
**Cơm. 4.2** . Sắp xếp các đối tượng trong sơ đồ trình tự

Bây giờ trên sơ đồ, bạn nên đặt các thông điệp sẽ được trao đổi giữa các đối tượng (Bảng 4.1, Hình 4.3):

Bảng 4.1

|  |  |  |  |
| --- | --- | --- | --- |
| **Số tin nhắn** | **Sự vật -**  **người gửi tin nhắn** | **Sự vật -**  **người nhận tin nhắn** | **Tên** |
| 1 | Người quản lý  làm việc với khách hàng | Tùy chọn đặt hàng | nhập mật khẩu |
| 2 | Tùy chọn đặt hàng | Tùy chọn đặt hàng | kiểm tra mật khẩu |
| 3 | Người quản lý  làm việc với khách hàng | Tùy chọn đặt hàng | sự lựa chọn  thao tác "thêm" |
| 4 | Tùy chọn đặt hàng | Thêm đơn hàng mới | hiển thị các trường đầu vào |
| 5 | Người quản lý  làm việc với khách hàng | Thêm đơn hàng mới | chọn loại máy tính |
| 6 | Thêm đơn hàng mới | Trình quản lý đơn hàng | lấy danh sách khách hàng |
| 7 | Trình quản lý đơn hàng | Khách hàng | lấy danh sách khách hàng |
| 8 | Khách hàng | Thêm đơn hàng mới | danh sách khách hàng |
| 9 | Thêm đơn hàng mới | Thêm đơn hàng mới | hiển thị danh sách khách hàng |
| 10 | Người quản lý  làm việc với khách hàng | Thêm đơn hàng mới | sự lựa chọn của khách hàng |

|  |  |  |  |
| --- | --- | --- | --- |
| 11 | Thêm đơn hàng mới | Trình quản lý đơn hàng | nhận được  danh sách các thành phần |
| 12 | Trình quản lý đơn hàng | Thành phần | nhận được  danh sách các thành phần |
| 13 | Thành phần | Thêm đơn hàng mới | danh sách các thành phần |
| 14 | Thêm đơn hàng mới | Thêm đơn hàng mới | trưng bày  danh sách các thành phần |
| 15 | Người quản lý  làm việc với khách hàng | Thêm đơn hàng mới | \* sự lựa chọn  thành phần cần thiết |
| 16 | Người quản lý  làm việc với khách hàng | Thêm đơn hàng mới | lưu đơn hàng |
| 17 | Thêm đơn hàng mới | Trình quản lý đơn hàng | chuyển giao quyền kiểm soát |
| 18 | Trình quản lý đơn hàng | Đặt hàng | cứu |



**Cơm. 4.3.** Sơ đồ trình tự cuối cùng

Để hiển thị số tin nhắn trong Rational Rose, bạn cần vào menu Tools

> Tùy chọn > tab Sơ đồ , chọn hộp bên cạnh Trình tự đánh số .

# Phòng thí nghiệm số 5

# Tạo sơ đồ lớp

**Mục đích công việc:** đạt được kỹ năng xây dựng sơ đồ lớp, tạo các gói và nhóm các lớp thành các gói.

**Bài tập:**

1. tạo sơ đồ lớp *\** cho một trong các tình huống sơ đồ ca sử dụng được tạo trong phòng thí nghiệm trước. Đối với mỗi lớp, các thuộc tính và hoạt động phải được chỉ định. Mỗi lớp phải được ghi lại chi tiết - cần cung cấp mô tả văn bản về chính lớp đó, mô tả về các thuộc tính và hoạt động của nó;
2. tạo các gói để nhóm các lớp được tạo trong đoạn văn

1;

1. nhóm các lớp từ điểm 1 thành các gói; 4. Tạo sơ đồ lớp của riêng bạn cho mỗi gói.

5. xây dựng sơ đồ lớp chủ.

*\* Lưu ý: Nên xem sơ đồ lớp từ quan điểm khái niệm, được sử dụng trong các giai đoạn đầu của mô hình hóa và phát triển.*

**Nội dung của báo cáo:**

1. đã tạo sơ đồ lớp (đối với sơ đồ lớp ở mục 2 của nhiệm vụ, phải nêu rõ kịch bản xây dựng sơ đồ lớp này);
2. Mô tả ngắn gọn về từng lớp được tạo và mối quan hệ giữa các lớp.

Sơ đồ lớp ( lớp diagram ) được sử dụng để mô hình hóa chế độ xem tĩnh của hệ thống theo quan điểm thiết kế. *Sơ đồ lớp* là sơ đồ thể hiện nhiều lớp, giao diện, sự cộng tác và mối quan hệ giữa chúng. Được sử dụng cho các mục đích sau:

* để mô hình hóa từ vựng của một hệ thống: liên quan đến việc quyết định những phần trừu tượng nào là một phần của hệ thống và cái nào không. Sơ đồ lớp có thể được sử dụng để xác định những phần trừu tượng này và trách nhiệm của chúng;
* để mô hình hóa các hợp tác đơn giản. Hợp tác là một cộng đồng gồm các lớp, giao diện và các thành phần khác làm việc cùng nhau để cung cấp một số hành vi hợp tác;  để mô hình hóa thiết kế logic của cơ sở dữ liệu. Theo Martin Fowler, có ba trường phái tư tưởng khác nhau khi xây dựng sơ đồ lớp hoặc bất kỳ mô hình nào khác:
* quan điểm khái niệm - sơ đồ lớp dùng để thể hiện các khái niệm về lĩnh vực chủ đề đang được nghiên cứu. Các khái niệm này sẽ tương ứng với các lớp triển khai chúng, nhưng có thể không có sự tương ứng trực tiếp. Một mô hình khái niệm có thể có ít hoặc không có mối quan hệ nào với phần mềm triển khai nó, vì vậy nó có thể được xem mà không cần tham chiếu đến bất kỳ ngôn ngữ lập trình nào;
* quan điểm đặc tả - một hệ thống phần mềm được xem xét và chỉ các giao diện của nó được xem xét chứ không phải việc triển khai nó;
* Quan điểm thực hiện - các lớp của sơ đồ tương ứng với các lớp thực của hệ thống phần mềm. **Ví dụ về công việc đã thực hiện**

### 1. Tạo sơ đồ lớp cho kịch bản "Thêm đơn hàng mới" của trường hợp sử dụng "Làm việc với đơn hàng"

Chúng ta sẽ xem xét sơ đồ lớp từ quan điểm khái niệm. Để đơn giản hóa nhiệm vụ và không làm lộn xộn sơ đồ với các chi tiết không quan trọng , chúng tôi sẽ không chỉ định các phương thức setX và getX cho từng thuộc tính của các lớp X.

Hãy tạo một sơ đồ lớp mới trong Logical View của trình duyệt và gọi nó là " Add New Order ". Trong trường tài liệu, hãy viết nội dung sau cho nó: “Sơ đồ lớp cho kịch bản “Thêm đơn hàng mới” của tiền lệ “Làm việc với đơn hàng”

Hãy bắt đầu điền vào sơ đồ bằng cách xác định các lớp thực thể.

Kịch bản được đề cập bao gồm:

* bản thân mệnh lệnh;
* khách hàng đặt hàng;
* thành phần có trong đơn hàng.

Hãy tạo các lớp thực thể Đơn hàng (Đặt hàng), Khách hàng

(Khách hàng) và ComponentPart . Vì một đơn hàng có thể bao gồm nhiều thành phần khác nhau và một thành phần có thể được bao gồm trong nhiều đơn hàng nên chúng tôi giới thiệu một thực thể lớp khác OrderItem (Nội dung đơn hàng). Hãy mô tả từng lớp.

**Lớp khách hàng *:***

|  |  |
| --- | --- |
| **tham số** | **Nghĩa** |
| Bình luận | Một lớp đại diện cho khách hàng của một công ty |
| Thuộc tính | tên : Sợi dây - địa chỉ tên khách hàng : Sợi dây - địa chỉ điện thoại của khách hàng : Sợi dây - số điện thoại của khách hàng  Tất cả các thuộc tính đều có công cụ sửa đổi truy cập - riêng tư |
| Hoạt động | AddClient ( ) - thêm ứng dụng khách mới RemoveClient ( ) - xóa ứng dụng khách hiện có GetInfo ( ) - lấy thông tin về ứng dụng khách  Tất cả các hoạt động đều có công cụ sửa đổi truy cập - công khai |

**Lớp học *Đặt hàng :***

|  |  |
| --- | --- |
| **tham số** | **Nghĩa** |
| Bình luận | Một lớp đại diện cho đơn hàng mà khách hàng đặt |
| Thuộc tính | số thứ tự : Số nguyên - số thứ tự orderDate : Ngày - ngày đơn hàng được đặt orderComplete : Ngày - ngày đơn hàng được hoàn thành  Tất cả các thuộc tính đều có công cụ sửa đổi truy cập - riêng tư |
| Hoạt động | Create ( ) - tạo đơn hàng mới SetInfo ( ) - nhập thông tin về đơn hàng GetInfo ( ) - lấy thông tin về đơn hàng  Tất cả các hoạt động đều có công cụ sửa đổi truy cập - công khai |

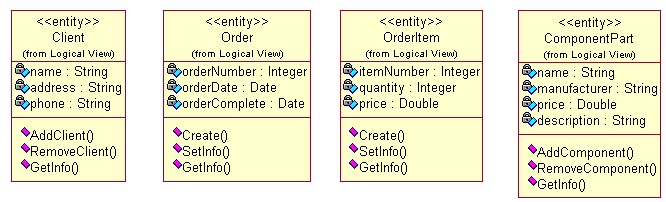
**Lớp học *Đặt hàng :***

|  |  |
| --- | --- |
| **tham số** | **Nghĩa** |
| Bình luận | Một lớp đại diện cho một mặt hàng đặt hàng mà khách hàng thực hiện |
| Thuộc tính | mụcSố : Số nguyên - mã số đơn hàng số lượng : Số nguyên - số lượng linh kiện giá : Đôi - đơn giá  Tất cả các thuộc tính đều có công cụ sửa đổi truy cập - riêng tư |
| Hoạt động | Create ( ) - tạo dòng lệnh mới SetInfo ( ) - nhập thông tin về dòng lệnh GetInfo ( ) - lấy thông tin về dòng lệnh  Tất cả các hoạt động đều có công cụ sửa đổi truy cập - công khai |

**Lớp học *Thành phần :***

|  |  |
| --- | --- |
| **tham số** | **Nghĩa** |
| Bình luận | Một lớp đại diện cho các thành phần |
| Thuộc tính | tên : Chuỗi - tên nhà sản xuất : Chuỗi - giá nhà sản xuất : Gấp đôi - giá mỗi đơn vị mô tả - mô tả  Tất cả các thuộc tính đều có công cụ sửa đổi truy cập - riêng tư |
| Hoạt động | AddComponent ( ) - thêm một cái mới  sản phẩm thành phần  RemoveComponent ( ) - loại bỏ một thành phần  GetInfo ( ) - lấy thông tin về một sản phẩm thành phần  Tất cả các hoạt động đều có công cụ sửa đổi truy cập - công khai |

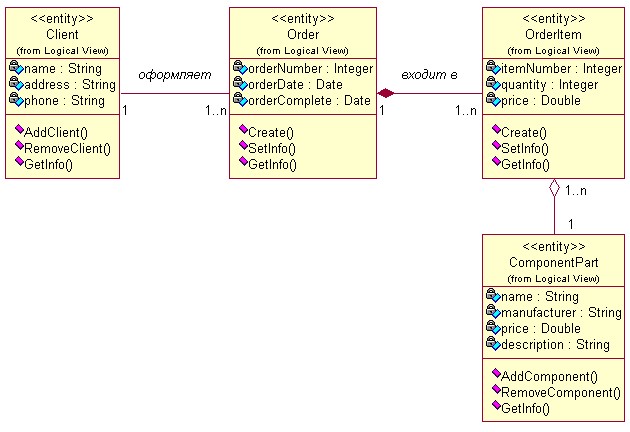
Kết quả của việc tạo các lớp thực thể được thể hiện trong Hình 2. 5.1:



**Cơm. 5.1** . Các lớp thực thể đã tạo Hãy thêm mối quan hệ giữa các lớp (Hình 5.2):

* Khách hàng và Đơn đặt hàng là một mối quan hệ liên kết, vì hai lớp này chỉ liên quan với nhau và không có loại mối quan hệ nào khác có thể được áp dụng ở đây. Một khách hàng có thể thực hiện một số đơn hàng, mỗi đơn hàng chỉ đến từ một khách hàng, do đó bội số giao tiếp ở phía lớp Khách hàng là 1, ở phía Đơn hàng - 1.. n;
* lớp Order và OrderItem là một mối quan hệ tổng hợp, vì dòng đơn hàng là một phần của đơn hàng và không thể tồn tại nếu không có nó. Một đơn hàng có thể chứa nhiều dòng đơn hàng, dòng đơn hàng chỉ đề cập đến một đơn hàng, do đó bội số của kết nối ở phía Đơn hàng là 1, ở phía OrderItem - 1.. n;
* lớp OrderItem và ComponentPart - một mối quan hệ tổng hợp, vì các sản phẩm thành phần là một phần của dòng đơn hàng nhưng cả hai đều là các lớp độc lập. Một mục thành phần có thể được bao gồm trong nhiều dòng đơn hàng; một dòng đơn hàng chỉ bao gồm một mục thành phần, do đó bội số của kết nối ở phía ComponentPart là 1, ở phía OrderItem -

1..n.



**Cơm. 5.2.** Các lớp thực thể và mối quan hệ giữa chúng

Bây giờ chúng ta hãy thêm các lớp ranh giới và lớp kiểm soát vào sơ đồ (Hình 5.3). Kịch bản được đề cập chỉ là một trong những hành động mà trường hợp sử dụng “Làm việc theo đơn đặt hàng” cung cấp.

Ca sử dụng này cũng cho phép bạn xem, chỉnh sửa hoặc xóa đơn hàng. Điều này có nghĩa là cần phải cung cấp một cơ chế cho phép lựa chọn hành động mong muốn.

Với mục đích này, hãy tạo một lớp ranh giới OrderOptions với nhận xét "Lớp cung cấp cơ chế làm việc với các đơn đặt hàng".

Chúng tôi cũng sẽ tạo một lớp ranh giới AddNewOrder (Thêm đơn hàng mới), lớp này sẽ được sử dụng để thêm đơn hàng mới (bình luận - “Lớp được sử dụng để thêm đơn hàng mới”).

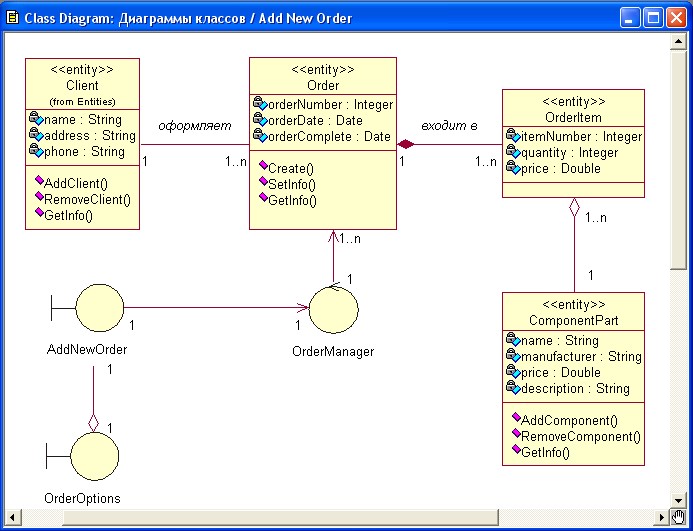
Mối quan hệ giữa các lớp này là một tập hợp, vì trong trường hợp này lớp AddNewOrder được coi là một phần của lớp OrderOptions , trong đó các lớp để xem, chỉnh sửa và xóa đơn hàng cũng sẽ là một phần. Mối quan hệ là 1 đến 1, vì lớp OrderOptions chỉ bao gồm một lớp AddNewOrder .

Bây giờ chúng ta hãy chuyển sang kiểm soát các lớp. Hãy thêm một lớp điều khiển OrderManager (Trình quản lý để làm việc với các đơn hàng) với nhận xét "Lớp điều khiển để xử lý luồng sự kiện của" Làm việc với các đơn hàng " trước đó", sẽ đảm bảo xử lý luồng sự kiện cho trường hợp sử dụng được đề cập .

Lớp này sẽ được liên kết với các lớp AddNewOrder và Order . Mối quan hệ giữa lớp AddNewOrder và OrderManager là mối liên kết một chiều

với tỷ lệ mối quan hệ là 1 trên 1, vì một phiên bản của lớp AddNewOrder chỉ tương tác với một phiên bản của lớp OrderManager . Mối quan hệ giữa các lớp OrderManager và Order là một liên kết một chiều với tỷ lệ mối quan hệ từ 1 đến 1..n , vì một lớp OrderManager có thể tương tác với nhiều lớp Order .

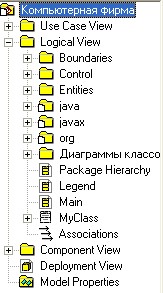
Phiên bản cuối cùng của sơ đồ lớp được hiển thị trong Hình. 5.3:



**Cơm. 5.3.** Sơ đồ lớp học cuối cùng

### 2. Tạo gói

Các gói được thiết kế để nhóm các mục thành các nhóm dựa trên các tiêu chí nhất định. Trong trường hợp đơn giản nhất, các lớp có thể được nhóm lại theo khuôn mẫu của chúng. Hãy tạo ba gói: Entities ( các lớp thực thể ), Boundaries (các lớp ranh giới) và Control (các lớp điều khiển). Để thực hiện việc này, nhấp chuột phải vào Logical View của trình duyệt và chọn New > trong menu ngữ cảnh xuất hiện . Gói (Mới > Gói) và nhập tên cho gói. Kết quả tạo gói được thể hiện trong Hình 5.4:



**Cơm. 5.4.** Gói đã tạo

Trong trường Tài liệu, đối với mỗi gói chúng ta sẽ đặt nhận xét:

* đối với nhận xét gói Thực thể : gói chứa các lớp thực thể ;
* cho gói Ranh giới nhận xét: gói chứa các lớp ranh giới;
* đối với nhận xét về gói Điều khiển: gói chứa các lớp điều khiển.

### 3. Nhóm các lớp thành các gói

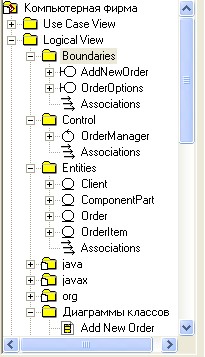
Việc nhóm các lớp thành các gói được thực hiện bằng cách kéo lớp thích hợp vào gói thích hợp trong Chế độ xem logic của trình duyệt. Chúng ta sẽ nhóm các lớp đã tạo như sau:

* lớp học *Khách hàng, Đơn hàng, Mục đặt hàng* Và *Thành phần* chúng tôi sẽ di chuyển nó V. túi nhựa *Thực thể;*
* Các lớp *Tùy chọn đặt hàng* và *Thêm NewOrder* di chuyển đến

Gói *ranh giới ;*

* Lớp học *Trình quản lý đơn hàng* Hãy chuyển nó sang *gói Control.*

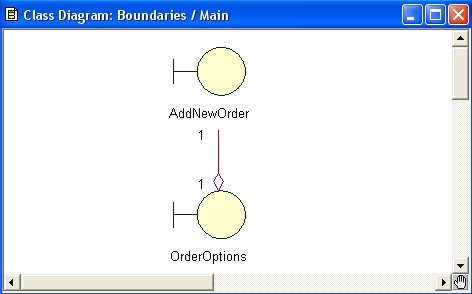
Kết quả cuối cùng được thể hiện trong hình. 5.5.



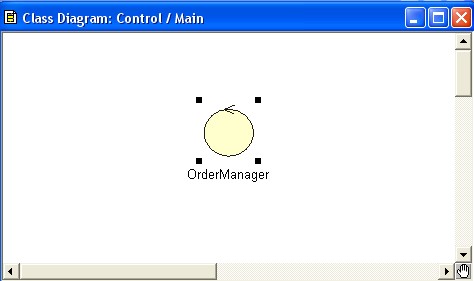
**Cơm. 5.5.** Các lớp và gói cho kịch bản "Thêm đơn hàng mới"

### 4. Thêm sơ đồ lớp cho từng gói

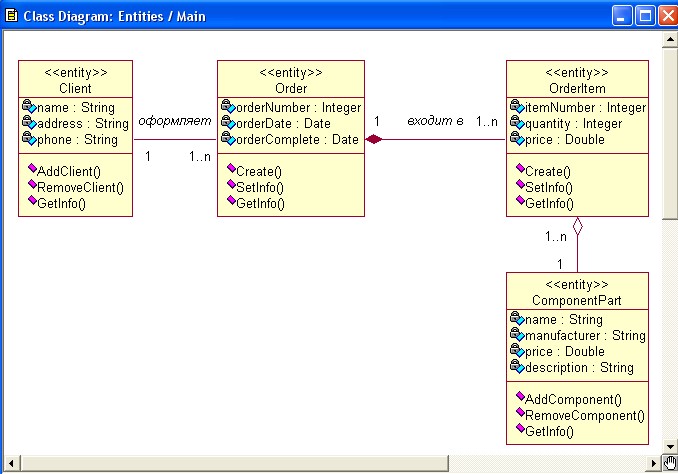
Để thêm sơ đồ vào gói, nhấp chuột phải vào gói, chọn Mới > Sơ đồ lớp trong menu ngữ cảnh xuất hiện , nhập tên lớp *Main* (Chính), sau đó mở sơ đồ bằng cách bấm đúp vào sơ đồ và chuyển các lớp cần thiết vào đó. Mối quan hệ giữa các lớp thuộc cùng một gói sẽ được chuyển tự động. Kết quả tạo sơ đồ lớp cho gói *Boundaries* được hiển thị trong Hình 5.6, đối với gói *Điều khiển* - trong Hình 5.7, đối với gói *Thực thể -* trong Hình. 5.8.



**Cơm. 5.6.** Sơ đồ lớp gói ranh giới



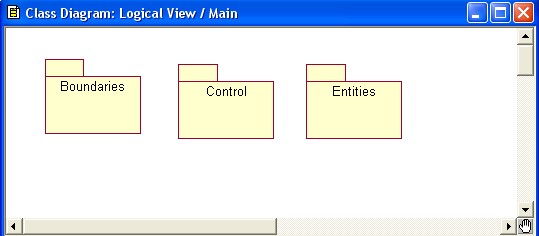
**Cơm. 5,7** . Sơ đồ lớp gói điều khiển



**Cơm. 5.8.** Sơ đồ lớp gói thực thể

### 5. Tạo sơ đồ lớp chính

Sơ đồ chính trong khung nhìn mô hình logic thường hiển thị các gói của hệ thống. Theo mặc định, sơ đồ lớp chính ( *Main* ) đã tồn tại trong Chế độ xem logic của trình duyệt. Để điền vào, bạn cần mở nó bằng cách nhấp đúp vào nó trong Chế độ xem logic của trình duyệt và kéo ba gói mà chúng tôi đã tạo vào đó (Hình 5.9):



**Cơm. 5.9.** Sơ đồ lớp chính

# Phòng thí nghiệm số 6

# Tạo sơ đồ hoạt động

**Mục đích công việc:** rèn luyện kỹ năng xây dựng sơ đồ hoạt động.

**Bài tập:**

1. tạo sơ đồ hoạt động mô tả một trong các quy trình kinh doanh của lĩnh vực chủ đề đã chọn;
2. Tạo sơ đồ hoạt động mô tả luồng sự kiện cho một trong các trường hợp sử dụng được tạo trong Lab #2.

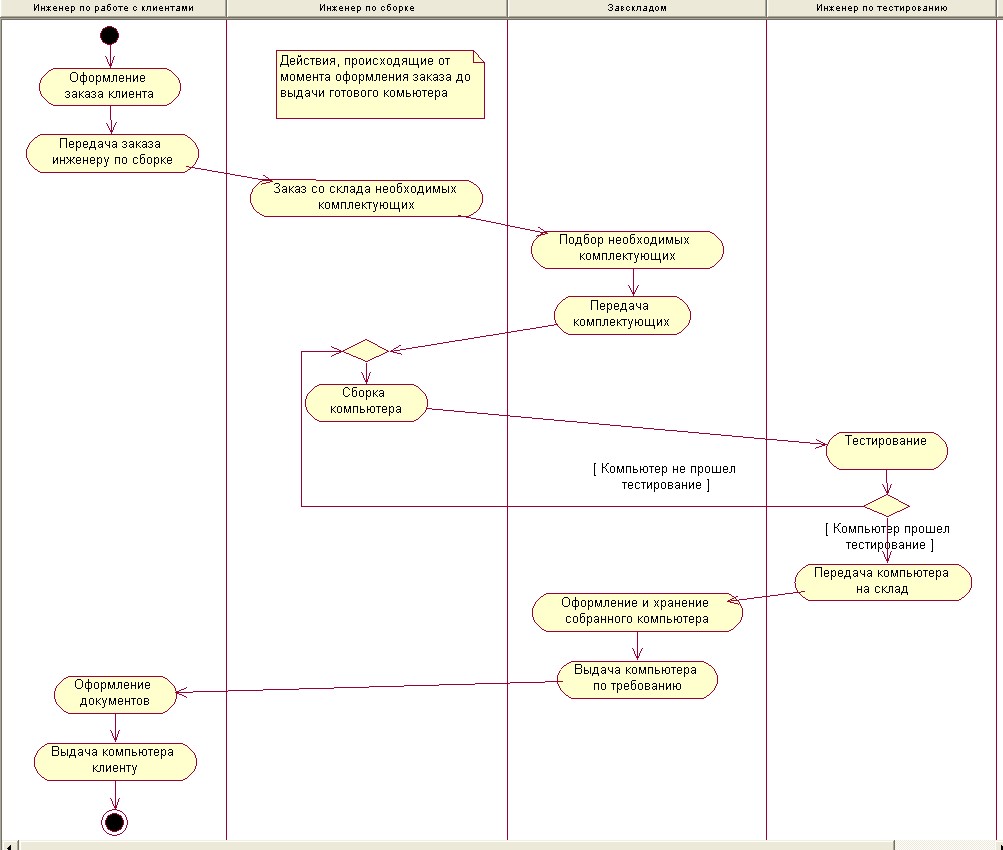
**Nội dung của báo cáo:** Các sơ đồ hoạt động được tạo cho biết quy trình kinh doanh và luồng sự kiện nào trong đó chúng mô tả trường hợp sử dụng.

### Ví dụ công việc đã thực hiện 1. Xây dựng sơ đồ hoạt động quy trình nghiệp vụ của doanh nghiệp lắp ráp máy tính

Nói chung, chúng ta hãy xem xét điều gì xảy ra tại doanh nghiệp từ thời điểm đặt hàng lắp ráp máy tính cho đến khi giao máy tính thành phẩm. Sau khi đặt hàng, người quản lý dịch vụ khách hàng sẽ chuyển nó cho người quản lý lắp ráp, người này đặt hàng các bộ phận cần thiết từ kho trước khi bắt đầu lắp ráp. Tại kho, người quản lý lựa chọn các linh kiện cần thiết (nếu không có thì đặt hàng từ người quản lý cung ứng) và chuyển cho kỹ sư lắp ráp. Sau khi nhận được linh kiện, người quản lý lắp ráp sẽ lắp ráp máy tính và bàn giao cho kỹ sư kiểm tra. Nếu máy tính kiểm tra không thành công, nó sẽ được trả lại để lắp ráp lại. Sau khi kiểm tra thành công, máy tính được chuyển về kho để lưu trữ. Từ kho, theo yêu cầu, máy tính được chuyển đến kỹ sư dịch vụ khách hàng, người này chuẩn bị tài liệu cho nó và đưa cho khách hàng.

Để tạo sơ đồ hoạt động, nhấp chuột phải vào Use Case View và chọn New > từ menu xuất hiện. Hoạt động Sơ đồ , nhập tên của nó, sau đó nhấp đúp vào biểu đồ trong trình duyệt của bạn để mở.

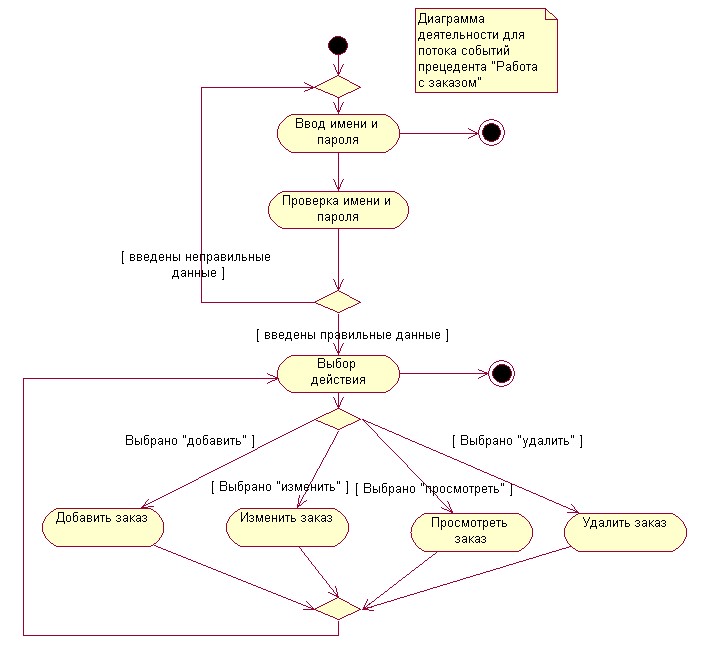
Kết quả xây dựng sơ đồ được thể hiện ở Hình 2. 6.1:



**Cơm. 6.1.** Sơ đồ hoạt động quy trình kinh doanh

### 2. Tạo sơ đồ luồng hoạt động cho luồng sự kiện ca sử dụng "Làm việc theo đơn đặt hàng"

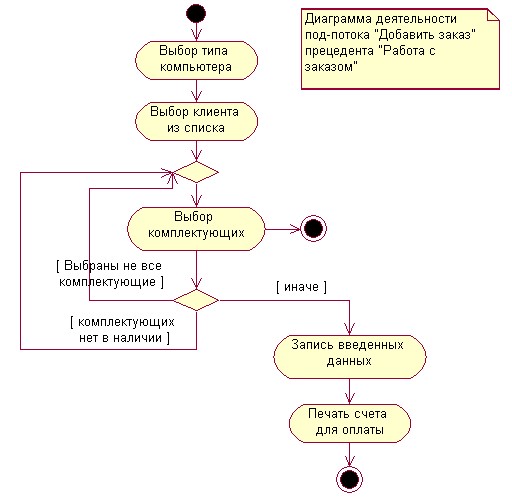
Luồng sự kiện trong trường hợp sử dụng "Làm việc với đơn hàng" bao gồm luồng chính, luồng phụ và luồng thay thế. Để không làm sơ đồ lộn xộn, chúng tôi sẽ hiển thị luồng sự kiện trong một số sơ đồ hoạt động. Trên đầu tiên (hãy gọi nó là chính), chúng tôi sẽ hiển thị các hành động cho luồng chính và luồng thay thế được liên kết với nó (Hình 6.2). Các luồng con có thể được hiển thị bằng cách phân tách hoạt động tương ứng của sơ đồ chính.



**Cơm. 6.2** . Sơ đồ hoạt động cho luồng sự kiện của trường hợp sử dụng "Làm việc theo đơn đặt hàng"

Để phân tách hoạt động của sơ đồ hoạt động, nhấp chuột phải vào sơ đồ đó và chọn Sơ đồ con > Hoạt động mới từ menu xuất hiện. Sơ đồ .

phân rã hành động Trong hình. Hình 6.3 hiển thị sơ đồ hoạt động của luồng phụ Thêm đơn hàng, đây là sự phân tách hoạt động Thêm đơn hàng của sơ đồ hoạt động chính.



**Cơm. 6.3.** Sơ đồ hoạt động của hành động Thêm đơn hàng

* 1. *Vendrov L.M.* Thiết kế phần mềm hệ thống thông tin kinh tế: Giáo trình. - tái bản lần thứ 2, đã sửa đổi . và bổ sung - M.: Tài chính và Thống kê, 2005. - 544 tr.: ill.
* 2. *Mikhailov L.* Công nghệ hướng đối tượng để phát triển hệ thống phần mềm. - M.: Tài chính và Thống kê, 2005. - 298 tr.: ill.
* 3. *Gaisaryan S.S.* Công nghệ hướng đối tượng để thiết kế hệ thống phần mềm ứng dụng. - M.: Tài chính và Thống kê, 2005.-349 tr.: ill.
* 4. http://www.citforum.ru/programming/oop\_rsis/glaval.shtml.
* 5. *Vendrov L.M.* Công nghệ hiện đại để tạo ra phần mềm. Ôn tập. - M.: Tài chính và Thống kê, 2005. - 260 tr.: ill.
* 6. http:// zeus . sai . msu . ru:7000/seminars/cis99/sap. shtml
* 7. http://www.interface.ru/rational/rose/rp8.htm
* 8. http://www.pressa.spb.ru/newspapers/it/arts/it-17-art-8.html
* 9. http://baks.gaz.ru/oradoc/Designer/des2000x.htm
* 10. http://www.interface.ru/news/rel\_visu.htm
* 11. http://www.interface.ru/fset.asp?Url=/ca/pplus.htm
* 12. http://www.intuit.rU/department/pl/umlbasics/l/3.html
* 13. http://www.info-system.ru/../new\_standart\_uml\_article.html
* 14. *Romanovsky K.Yu., Kuznetsov S.V., Koznov D.V.* Cách tiếp cận hướng đối tượng và sơ đồ lớp trong UML // Mô hình hóa trực quan hướng đối tượng / Ed. giáo sư A.N. ​- St. Petersburg: Nhà xuất bản Đại học St. Petersburg, 1999. - 215 tr.
* 15. Thiết kế hệ thống thông tin: Sách giáo khoa, cẩm nang/N.N. Zabotina. - Bratsk: Tổ chức giáo dục nhà nước "BrSTU". - Lúc 2 giờ - 2004. - Phần 1-2.-Phần. 1,- 146 tr.
* 16. *Maklakov S.V.* Tạo hệ thống thông tin với AllFusion Bộ mô hình hóa . - M.: Williams, 2004. - 223 tr.: ill.
* 17. *Matsyashek L.* Phân tích yêu cầu và thiết kế hệ thống. Phát triển hệ thống thông tin sử dụng UML: Per. từ tiếng Anh - M.: Williams, 2002. - 432 tr.: ill.
* 18. http://ooad.asf.ru/standarts/uml/
* 19. *Vendrov A.M.* Công nghệ CASE Các phương pháp và công cụ hiện đại để thiết kế hệ thống thông tin. - M.: Tài chính và Thống kê, 2004. - 238 tr.: ill.
* 20. *Vendrov L.M.* Xem xét các công cụ thiết kế hệ thống thông tin. - M.: Tài chính và Thống kê, 2003. - 341 tr.: ill.
* 21. *Vendrov L.M.* Hội thảo thiết kế phần mềm hệ thống thông tin kinh tế: Giáo trình, cẩm nang. —